

**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE  
MINAS GERAIS  
CAMPUS V  
EQUIPE ERA2-D2**

**RIOS DE DADOS: EXPLORANDO UMA  
PLATAFORMA DE MONITORAMENTO HÍDRICO**

**Divinópolis/MG  
2024**

Ana Clara Monteiro Caetano, Ana Clara Siqueira, Davi Faria de Sousa Guimarães,  
Gustavo Campos da Silva Silveira, Isabella Olímpio Rocha, João Carlos Santos, Lucas  
Gabriel Soares Borges, Nina Lage Motta, Pedro Henrique Carlos de Souza, Suellen de  
Faria Silvério

## **RIOS DE DADOS: EXPLORANDO UMA PLATAFORMA DE MONITORAMENTO HÍDRICO**

Trabalho submetido ao Torneio Brasil de  
Robótica como requisito parcial à avaliação  
no quesito Mérito Científico.

Técnico: João Luiz de Sousa Vieira

Mentor: Prof. Me. Diêgo Fernandes da Cruz

**Divinópolis/MG  
2024**

## **Agradecimentos**

Tendo em vista a longa jornada para a elaboração do trabalho não podemos dizer que chegamos aqui sem ajuda. Nosso sincero agradecimento a todas as pessoas que divulgaram o projeto, compraram a rifa do sorteio que realizamos, e torceram positivamente. Agradeço também aos pais por colaborarem e se manterem presentes sem medir esforços para realização deste trabalho. Por fim, agradecemos também aos apoiadores CEFET-MG, MakeSoft Solutions, ACID Divinópolis e SEBRAE também aos patrocinadores, Auto Elétrica do Vaguinho, Auto Elétrica JB, Distribuidora Gontijo, Fátima Designs, Gonçalves e Niza Advogados, Gráfica GL, Grupo Planejar, Harpia Harpyja, MAC Supermercados, Sicoob Divicred, Supermercado KIT 10, Supermercado Olímpio Rocha e USELIGAS, pois sem eles não haveria a possibilidade de realizarmos esse projeto.

## **Resumo**

O 6º Objetivo de Desenvolvimento Sustentável (ODS) da ONU, água potável e saneamento, destaca-se como uma das metas cruciais para a sustentabilidade global. Dentro dessa meta, o objetivo específico 6.6, determinado pela modalidade High, ressalta a necessidade de proteger e restaurar os ecossistemas relacionados à água. Além disso, destaca a importância de promover a conscientização e a competência para agir, não apenas entre a sociedade civil, mas também no setor privado e governamental.

Com o intuito de atingir esse objetivo, a equipe ERA2-D2 desenvolveu uma plataforma de monitoramento aquático que é capaz de registrar parâmetros como pH, turbidez, níveis de oxigênio dissolvido, temperatura e condutividade. Esses dados são disponibilizados por meio de um site que permite a visualização e análise da qualidade da água, facilitando a tomada de decisões informadas, a implementação de medidas corretivas e o engajamento comunitário.

A utilização da plataforma contribui para a sustentabilidade dos recursos hídricos e a segurança da população, permitindo a detecção precoce de problemas e ações corretivas oportunas. Esta ferramenta representa uma solução prática e viável para o monitoramento contínuo da qualidade da água. Ao fornecer um panorama preciso e atualizado das condições aquáticas, contribui significativamente para o cumprimento de metas ambientais, incluindo aquelas relacionadas à ODS 6, que visa a ação global da água e de todos os sistemas aquáticos.

**Palavras-chave:** monitoramento, plataforma, água, conscientização, qualidade.

## **Abstract**

The 6th Objective of UN Sustainable Development (SDG), clean water and sanitation, stands out as one of the crucial goals for global sustainability. Within this goal, specific objective 6.6, determined by the High modality, highlights the need to protect and restore water-related ecosystems. Furthermore, it highlights the importance of promoting awareness and capacity for action not only among civil society, but also in the private and government sectors.

In order to achieve this goal, the ERA2-D2 team developed an aquatic monitoring platform that is capable of recording parameters such as pH, turbidity, dissolved oxygen levels, temperature and conductivity. This data is made available through a website that allows visualization and analysis of water quality, facilitating informed decision-making, implementation of corrective measures and community engagement.

Using the platform contributes to the sustainability of water resources and the safety of the population, allowing for the early detection of problems and timely corrective actions. This platform represents a practical and viable solution for the continuous monitoring of water quality, by providing an accurate and updated overview of aquatic conditions it significantly contributes to the achievement of environmental goals, including those related to SDG 6, which aims for global action on water and all aquatic systems.

Keywords: monitoring, platform, water, awareness, quality.

# Sumário

<b>1. INTRODUÇÃO .....</b>	<b>7</b>
1.1 PROBLEMA ABORDADO.....	7
1.2 HIPÓTESES.....	10
1.3 OBJETIVOS.....	11
<b>1.3.1 Objetivo geral.....</b>	<b>11</b>
<b>1.3.2 Objetivos específicos.....</b>	<b>11</b>
1.4 JUSTIFICATIVA .....	12
<b>2. REFERENCIAL TEÓRICO .....</b>	<b>14</b>
2.1 USO HÍDRICOS.....	14
2.2 QUALIDADE DA ÁGUA.....	16
2.3 SANEAMENTO INADEQUADO .....	19
2.4 ALTERAÇÕES NO CICLO HIDROLÓGICO .....	24
<b>3. METODOLOGIA.....</b>	<b>26</b>
3.1 DESENVOLVIMENTO DA PLATAFORMA .....	34
3.2 FERRAMENTAS.....	35
3.3 SIMULAÇÃO .....	29
3.4 DESENVOLVIMENTO DA PCI.....	30
3.5 CONFECÇÃO DO CIRCUITO ELETRÔNICO.....	37
3.6 ARQUITETURA DA PLATAFORMA.....	40
3.7 PROGRAMAÇÃO .....	45
3.8 SITE .....	55
<b>4. RESULTADOS .....</b>	<b>66</b>
4.1 DISCUSSÃO.....	69
4.2 IMPLICAÇÕES AMBIENTAIS.....	70
4.3 EMBASAMENTO CIENTÍFICO.....	71
<b>5. RECURSOS UTILIZADOS .....</b>	<b>73</b>
<b>6. CRONOGRAMA.....</b>	<b>74</b>
<b>7. CONSIDERAÇÕES FINAIS .....</b>	<b>76</b>
<b>8. REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>77</b>
<b>9. APÊNDICE .....</b>	<b>80</b>

# **1. INTRODUÇÃO**

## **1.1 Problema abordado**

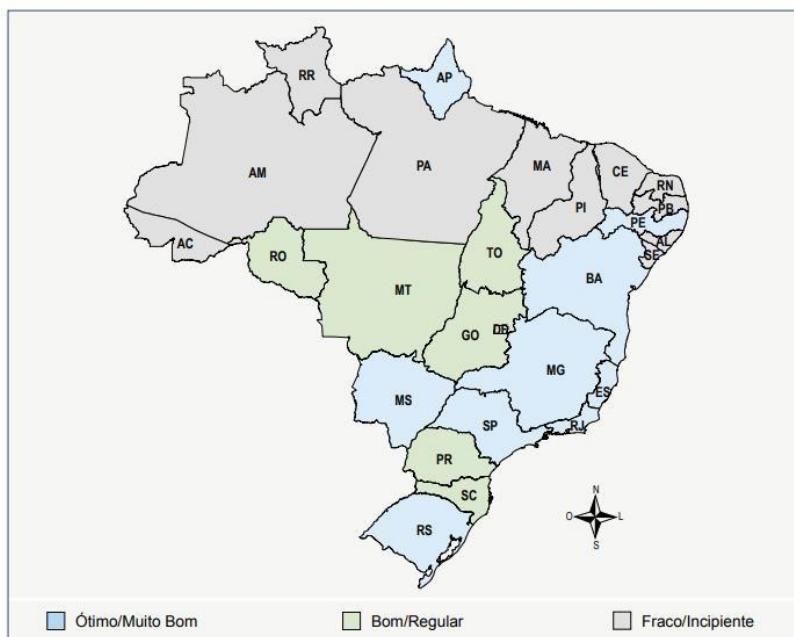
A água está intrinsecamente ligada à sobrevivência de todas as formas de vida que conhecemos e está intimamente relacionada ao desenvolvimento das sociedades e culturas, tendo uma importância ecológica, econômica e social. No Brasil, a água é utilizada principalmente para a irrigação de lavouras, abastecimento público, atividades industriais, geração de energia, extração mineral, aquicultura, navegação, turismo e lazer. Cada uso depende de condições específicas de quantidade e qualidade das águas, e também pode afetá-las. Esses parâmetros são influenciados pela presença de microrganismos causadores de doenças, além de detritos, poluentes como ferro, areia, sedimentos, e outros resíduos, que vêm prejudicando seriamente toda a ecologia terrestre e comprometendo sua vitalidade.

A poluição dos ambientes aquáticos tem se tornado um problema frequente, especialmente devido às suas consequências mais evidentes, como a escassez de fontes limpas para abastecimento e a mortandade de organismos (Queiroz et al. 2008). Dessa forma, a crescente degradação da qualidade da água tem levado à busca por estratégias para proteger e manter a qualidade dos ecossistemas aquáticos. Para avaliar o estado desses ecossistemas, um dos procedimentos mais comuns é o monitoramento, definido como a observação e estudo sistemático do meio ambiente. Em termos científicos, os dados coletados são utilizados para gerar conhecimento.

A importância da qualidade da água está bem conceituada na Política Nacional de Recursos Hídricos (PNRH), que define, dentre seus objetivos, "assegurar à atual e às futuras gerações a necessária disponibilidade de água, em padrões de qualidade adequados aos respectivos usos" (Art. 2º, Cap. II, Tit. I, Lei nº 9.433). Entretanto, apesar da relevância desse recurso, a gestão da qualidade da água não tem recebido o mesmo destaque que a gestão da quantidade, seja no aspecto legal, nos arranjos institucionais em funcionamento no setor ou na operacionalização dos sistemas de gestão (PORTO, 2002).

Apesar da importância da qualidade da água ressaltada pela PNRH, o monitoramento efetivo dessa qualidade ainda apresenta muitas falhas no Brasil. A informação sobre a qualidade da água no país ainda é escassa ou inexistente em várias bacias. Poucos estados possuem redes de monitoramento adequadas. Segundo o Ministério do Meio Ambiente (MMA), apenas nove unidades da Federação possuem sistemas de monitoramento da qualidade da água considerados ótimos ou muito bons; cinco possuem sistemas bons ou regulares; e treze apresentam sistemas fracos ou incipientes. Esse levantamento, realizado entre outubro de 2000 e julho de 2001, agrupou os estados de acordo com quatro aspectos: porcentagem das bacias hidrográficas monitoradas, tipos de parâmetros analisados, frequência de amostragem e forma de disponibilização da informação pelos estados (MMA, 2002).

Figura 1.1 - Mapa que indica a qualidade de monitoramento das bacias hidrográficas.



Fonte: MMA, 2002.

Em termos gerais, considerando as redes estaduais e a Rede Hidrometeorológica Nacional, é perceptível que apenas a região Sudeste possui uma condição adequada de monitoramento da qualidade da água. As demais regiões apresentam-se bastante inferiores nesse quesito, com destaque para as regiões Norte e Nordeste. Essas limitações no monitoramento dificultam o diagnóstico detalhado da qualidade dos corpos d'água no país.

Dessa forma, destaca-se o programa de monitoramento da qualidade das águas superficiais e subterrâneas de Minas Gerais, denominado Programa Águas de Minas, o qual é executado desde 1997 e está sob a responsabilidade do IGAM, Instituto Mineiro de Gestão das Águas. Os resultados desse monitoramento possibilitam, por exemplo, identificar a variação da qualidade das águas em períodos específicos, detectar e medir tendências, elaborar diagnósticos e propor ações visando à proteção e à recuperação dos recursos hídricos.

O monitoramento da qualidade das águas é uma ferramenta imprescindível para a PNRH, tanto para garantir o cumprimento de seus objetivos quanto para servir como base na execução de seus instrumentos. Segundo a Agência Nacional de Águas e Saneamento Básico (ANA), as parcelas de água devem ser constantemente monitoradas para que se determine a quantidade de recursos hídricos e se verifiquem suas condições.

A ANA (2019) também afirma que, para avaliar as condições das águas em rios e cursos d'água, são determinados os níveis de parâmetros biológicos, físicos e químicos. Essas variáveis são utilizadas como indicadores de qualidade e auxiliam na classificação dos corpos hídricos. Diante disso, é correto afirmar que o acompanhamento contínuo desses parâmetros é indispensável para a proteção dos recursos hídricos.

Nesse contexto, a ODS 6 se alinha diretamente com nossa realidade atual, buscando enfrentar esses desafios. Ela destaca a necessidade de proteger os ecossistemas aquáticos, essenciais para a vida. Isso requer promover a conscientização da sociedade sobre o ambiente ao seu redor, gerando mudanças a longo prazo. Assim, o Torneio Brasil de Robótica, na categoria High, nos desafiou a pensar em soluções para proteger e restaurar sistemas relacionados à água. Portanto, nossa equipe optou por desenvolver uma plataforma que, por meio do monitoramento contínuo, permitirá melhorar a gestão de recursos hídricos, assegurar a saúde pública e a preservação ambiental, garantindo esse recurso para as gerações futuras.

Figura 1.2 - Tema referente a ODS 6.



Fonte: Organização das Nações Unidas, 2010.

## 1.2 Hipóteses

De acordo com a organização não governamental SOS Mata Atlântica, uma análise da qualidade de 111 rios brasileiros revelou que 23,3% das águas são classificadas como de qualidade ruim ou péssima. Segundo a legislação brasileira, águas nessas condições não podem ser tratadas para consumo humano nem utilizadas para irrigação de lavouras. Isso é preocupante, visto que a poluição dos rios não apenas compromete a biodiversidade aquática, mas também ameaça a segurança hídrica de milhões de indivíduos. A qualidade da água é um indicador crucial do estado dos ecossistemas e do bem-estar das populações humanas que dependem desses recursos. Quando um quarto dos rios analisados apresenta níveis de poluição tão altos, isso reflete uma falha significativa na gestão ambiental e na implementação de políticas públicas eficazes.

As fontes de contaminação dos cursos d'água são diversas devido às diferentes formas de uso dos recursos hídricos, como os usos urbanos e industriais, que geram efluentes e têm colaborado consideravelmente para a poluição dos rios. Apesar de estudos e monitoramentos em algumas áreas, é importante notar que a maioria dos cursos d'água não possui monitoramento algum. Portanto, a adoção de redes amplas de monitoramento é essencial para a implementação de medidas específicas que levem em consideração as peculiaridades de cada área monitorada, permitindo uma análise mais precisa. Assim, o monitoramento eficaz é a principal medida para identificar fontes de poluição e possibilitar ações corretivas.

Diante desse cenário, nossa equipe desenvolveu uma plataforma de

monitoramento equipada com sensores capazes de medir e coletar parâmetros qualitativos da água, fornecendo dados em tempo real sobre pH, temperatura, turbidez, condutividade e níveis de oxigênio dissolvidos. Os dados coletados são apresentados em um site, permitindo a identificação rápida de alterações na qualidade da água. A implementação da plataforma visa, sobretudo, ecossistemas aquáticos de água doce, como zonas úmidas, rios, aquíferos e lagos. Dessa forma, é possível acessar dados específicos sobre o ambiente monitorado, visto que cada um desses ecossistemas possui problemas e características próprias, demandando medidas específicas. Ao realizar esse monitoramento contínuo, a plataforma se torna um recurso eficiente e viável para informar políticas públicas, planejar e gerir recursos hídricos, apoiar a agricultura e a indústria, proteger a saúde pública, engajar a comunidade e atender a diversos outros setores.

## **1.3 Objetivos**

### **1.3.1 Objetivo geral**

Desenvolver uma plataforma para monitoramento da qualidade da água, visando proteger e restaurar ecossistemas aquáticos.

### **1.3.2 Objetivos específicos**

- Identificar e classificar os parâmetros prioritários para garantir a qualidade da água (pH, temperatura, condutividade elétrica, turbidez e níveis de oxigênio dissolvido);
- Avaliar a qualidade da água, com base nos dados coletados, seguindo os parâmetros do Índice de Qualidade das Águas (IQA);
- Permitir a análise de tendências ao longo do tempo, facilitando a avaliação de melhorias ou agravamentos na qualidade da água e nas concentrações de poluentes, ajudando a identificar fontes recorrentes de poluição;
- Monitorar continuamente para detectar rapidamente a presença de poluentes e tomar medidas corretivas de forma ágil;

- Auxiliar no planejamento e desenvolvimento de infraestruturas, como estações de tratamento de água e esgoto, além de apoiar pesquisas científicas e a gestão de crises hídricas;
- Conscientizar a população sobre a importância da preservação dos recursos hídricos e a adoção de práticas sustentáveis;
- Implementar a plataforma em diversos ambientes para garantir um estudo amplo e significativo da qualidade da água;
- Estabelecer o uso da plataforma em diferentes instituições e comunidades, visando monitorar a água de maneira colaborativa e interinstitucional.

A plataforma contribuirá significativamente para a proteção e restauração dos ecossistemas aquáticos. A capacidade de detectar e responder rapidamente a eventos de poluição, aliada à conscientização pública e à colaboração entre instituições, promoverá um ambiente mais saudável para as gerações presentes e futuras.

#### **1.4 Justificativa**

O monitoramento da qualidade da água permite avaliar as condições de um corpo hídrico e classificá-lo de acordo com as definições estabelecidas na legislação pertinente. Assim, as análises de monitoramento são de extrema importância, não apenas para garantir o cumprimento da legislação, mas, principalmente, para prevenir implicações prejudiciais ao meio ambiente e à saúde humana.

Segundo Maurren Vieira, especialista em recursos hídricos da ANA, a qualidade da água é insuficientemente monitorada, o que impede uma avaliação precisa da situação no país. Dessa forma, torna-se essencial implementar ações de monitoramento em ecossistemas aquáticos para garantir sua proteção e restauração.

A equipe ERA2-D2 desenvolveu uma plataforma de monitoramento da água que visa melhorar diversos aspectos, como a mitigação de riscos à saúde pública, a redução da degradação ambiental, a preservação dos recursos hídricos,

a minimização de problemas econômicos e a melhoria da qualidade de vida. A plataforma é uma solução altamente viável e acessível, podendo ser adotada por indivíduos, comunidades, empresas e instituições para o monitoramento eficiente de sistemas de água.

Além disso, a plataforma desempenha um papel essencial na conscientização da população sobre a qualidade da água. Considerando que a poluição dos ecossistemas aquáticos é um desafio global, essa conscientização é fundamental para fomentar a cooperação entre países e setores da sociedade. Informar as pessoas cria uma base de apoio para a implementação de ações coordenadas em níveis local, nacional e internacional. Isso capacita as pessoas a tomarem decisões informadas em suas vidas cotidianas, como o descarte adequado de lixo, melhorias no saneamento, apoio a políticas sustentáveis e outras ações. Além disso, a conscientização pública pode exercer pressão sobre governos e empresas, incentivando políticas e práticas mais sustentáveis.

## **2. REFERENCIAL TEÓRICO**

### **2.1 Uso Hídricos**

A água consiste em um recurso natural finito e essencial à vida, como componente bioquímico de todos os seres vivos, como meio de vida de várias espécies, como elemento representativo de valores sociais e culturais, além de importante fator de produção no desenvolvimento de diversas atividades econômicas. (Santin, 2013,p.201)

A água é crucial para diversas atividades humanas e setores, como o abastecimento doméstico, a indústria, a agricultura e a pecuária. Ela também é essencial para a aquicultura e a geração de energia em usinas hidrelétricas. Além disso, é utilizada na diluição e transporte de despejos, ajudando a minimizar o impacto ambiental. Portanto, a gestão eficaz e sustentável dos recursos hídricos é vital para atender às necessidades humanas e proteger esses recursos para o futuro.

Com o avanço da urbanização e da industrialização, a disparidade entre o aumento da demanda e a escassez de água tem se agravado, tornando o desenvolvimento sustentável dos recursos hídricos uma questão global urgente. Estima-se que a demanda global por água aumentará em 30% até 2050, chegando a cerca de 6.000 km<sup>3</sup> por ano, o que agrava ainda mais o equilíbrio entre oferta e demanda de água (Burek et al., 2016; Ali et al., 2017).

No Brasil, a situação da disponibilidade hídrica já é crítica. O número de pessoas afetadas pela seca aumentou drasticamente nos últimos anos, com o racionamento de água se tornando comum em várias regiões do país (ANA, 2018). A Associação Brasileira de Engenharia Sanitária e Ambiental (ABES) destaca que uma parte significativa do desperdício de água está relacionada às perdas na distribuição, que chegam a cerca de 40% do total de água tratada no Brasil.

Nesse contexto de crescente demanda, a Tabela 2.1 apresenta o consumo médio anual de água nas regiões do Brasil, e as Figuras 2.1 e 2.2 mostram a distribuição por tipo de uso consuntivo e por região. É possível observar que o consumo de água tem aumentado a cada ano.

Em 2019, cada habitante consumiu, em média, 153,9 litros de água por dia,

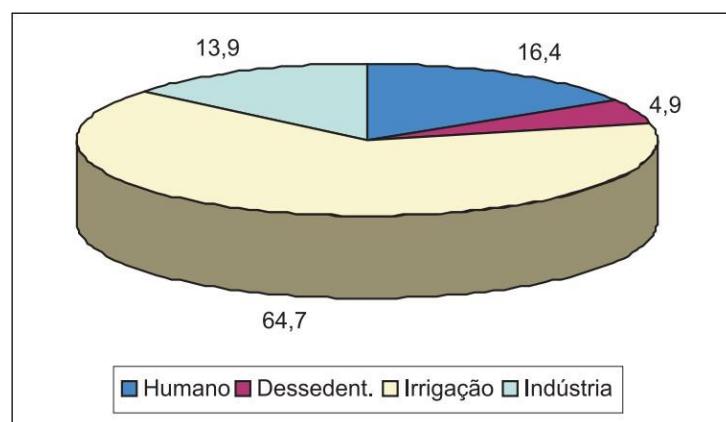
sendo que o ano de 2012 registrou a maior média, com 167,5 litros diários. A região Sudeste apresentou o maior consumo médio per capita, com destaque também para as regiões Norte e Nordeste. Mesmo com a crise hídrica de 2015, o consumo de 176 litros por habitante ultrapassou os maiores indicadores registrados em todas as regiões, como mostrado na Tabela 2.1.

Tabela 2.1 - Consumo médio per capita.

	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
Região Norte <sup>1</sup>	143.5	151.2	155.8	155.8	154.2	154.3	154.5	132.3	131.8	129.1
Região Nordeste <sup>1</sup>	117.3	120.6	131.2	125.8	118.9	116.1	112.5	113.6	115.4	120.6
Região Sudeste <sup>1</sup>	185.9	189.7	194.8	194.0	187.9	176.0	179.7	180.3	182.6	177.4
Região Sul <sup>1</sup>	145.4	147.9	149.3	149.9	153.6	148.7	144.2	145.2	146.1	146.4
Região Centro-Oeste <sup>1</sup>	154.9	157.5	156.5	160.7	158.8	148.8	148.5	146.1	144.9	147.8
Brasil <sup>1</sup>	159.0	162.6	167.5	166.3	162.0	154.0	154.1	153.6	154.9	153.9
População Amostrada <sup>2</sup>	156.9	157.9	160.3	165.7	167.9	168.9	170.9	172.1	173.2	174.7

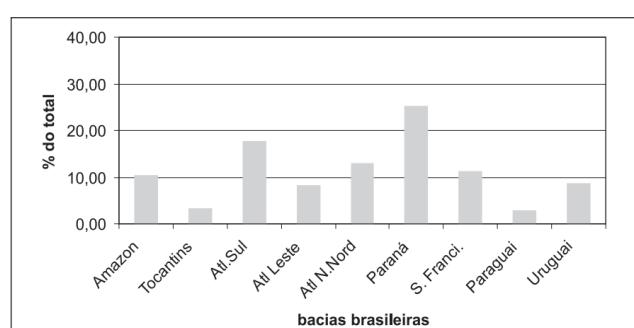
Fonte: SNIS, 2021.

Figura 2.1 - Distribuição de água por tipo de consumo.



Fonte: Organização das Nações Unidas, para Agricultura e Alimentação.

Figura 2.2 - Distribuição da demanda por água por grandeza.



Fonte: FGV, 1998.

Ademais, devido à importância da água para diversos aspectos, como sua

função como insumo produtivo, sua contribuição para a paisagem, seu papel em processos biogeoquímicos e sua essencialidade para a vida humana, os recursos hídricos são empregados para uma variedade de finalidades. Isso inclui a extração de água dos corpos hídricos, o que pode resultar na diminuição de sua quantidade e na degradação da qualidade da água devido aos usos subsequentes da água. Em contraste, os usos não consuntivos envolvem a utilização da água sem a necessidade de removê-la de seus corpos hídricos, ou seja, a água é utilizada *in situ*, sem causar alterações significativas em seu volume ou qualidade.

À medida que os centros urbanos e as atividades agrícolas se expandem, o consumo de água aumenta, o que também leva a uma maior geração de efluentes, como esgotos domésticos e industriais, graxas, águas contaminadas com defensivos químicos, rações e antibióticos para aquicultura, e partículas em suspensão, entre outros poluentes que afetam a qualidade da água. Portanto, há uma relação direta entre o uso da água e a necessidade de manter a qualidade da água para atender a essas demandas.

Diante desse cenário, é fundamental reforçar o monitoramento ambiental da água e investir em infraestrutura de saneamento básico. Uma abordagem holística e políticas eficazes são essenciais para enfrentar os desafios do uso da água e garantir a sustentabilidade dos recursos hídricos para as gerações futuras.

## 2.2 Qualidade da água

A disponibilidade de recursos hídricos é fundamental tanto para o desenvolvimento socioeconômico quanto ambiental de qualquer nação. Em países em desenvolvimento, especialmente, tem havido crescentes desafios na oferta de serviços de água, agravados pelo crescimento populacional, melhoria nos padrões de vida, urbanização e condições climáticas variáveis. Dessa forma, a água é essencial para a vida; no entanto, tão crucial quanto sua disponibilidade é a sua qualidade, que define sua adequação para consumo e proteção ambiental.

De modo geral, a qualidade da água é definida por sua composição física, química, biológica e radioativa, e pelos efeitos que seus constituintes podem causar ao ambiente. Certos usos da água, considerados críticos, exigem rigoroso controle de qualidade. Além disso, os padrões de qualidade variam significativamente conforme o uso, seja para consumo humano, indústria ou irrigação. No Brasil, as

normas de qualidade da água para consumo humano são regidas pelo Ministério da Saúde, conforme a Portaria nº 518, de 25 de março de 2004 (Brasil, 2004).

Os usos não consuntivos, aqueles que não implicam em consumo ou transformação da água de forma que ela se torne imprópria para outros usos ou precise ser substituída, não exigem padrões rígidos de qualidade, embora seja necessário manter um mínimo aceitável de qualidade e características estéticas, especialmente no que diz respeito à função paisagística. Esses usos estão associados a atividades como recreação e lazer, transporte, navegação e descarte de resíduos. Por outro lado, os usos consuntivos requerem padrões mais elevados de qualidade da água, pois incluem abastecimento público e industrial, práticas agroindustriais, dessedentação de animais e, o mais nobre de todos os usos, o consumo humano.

A qualidade da água em cursos hídricos é fundamental para a saúde ambiental e humana. A poluição, causada por efluentes industriais, esgoto doméstico, produtos químicos e resíduos sólidos, afeta diretamente essa qualidade. Contaminantes introduzidos nos corpos d'água podem alterar a composição química, reduzir a quantidade de oxigênio disponível e aumentar a carga de nutrientes, levando a problemas como a eutrofização e a morte de organismos aquáticos.

A poluição dos ambientes aquáticos tem se tornado um problema frequente, especialmente pelas suas consequências mais evidentes, tais como a escassez de fontes limpas para abastecimento e a mortandade dos organismos (Queiroz et al. 2008).

A contaminação desses recursos hídricos resulta de atividades industriais, urbanas, agrícolas e domésticas, que introduzem substâncias nocivas como produtos químicos, metais pesados, nutrientes em excesso e resíduos orgânicos. Esses poluentes têm impactos diretos na qualidade da água, afetando não apenas a saúde dos ecossistemas aquáticos, mas também a saúde humana e a economia. Atualmente, resíduos de pesticidas são frequentemente encontrados em amostras de água potável, como observa o oceanólogo Mathias Alberto Schramm, doutor em Ciência dos Alimentos e professor do Câmpus Itajaí.

A escassez de fontes limpas para abastecimento é uma das consequências mais visíveis da poluição hídrica. A água doce potável é um recurso finito e

essencial para a vida humana, sendo a poluição uma das principais ameaças à sua disponibilidade. A contaminação torna necessários investimentos significativos em tratamento de água, especialmente em países e regiões onde as infraestruturas são inadequadas para lidar com níveis crescentes de poluição. Além disso, a mortalidade dos organismos aquáticos evidencia os impactos ecológicos da poluição. Ecossistemas aquáticos complexos são habitados por uma vasta diversidade de seres vivos, todos interligados em uma teia de relações. A introdução de poluentes pode levar à diminuição das populações de peixes, anfíbios, aves aquáticas e outros organismos, desestabilizando os ecossistemas e comprometendo serviços ambientais essenciais, como a regulação do clima e a purificação da água.

Apesar da influência dos fatores naturais na qualidade da água, como o escoamento e a infiltração de partículas e impurezas do solo, a principal fonte de contaminação é a ação humana. A forma como o homem utiliza e ocupa o solo tem um impacto direto e significativo na qualidade dos recursos hídricos. A ação antrópica é a principal responsável pela deterioração da qualidade da água de fontes alternativas (nascentes, córregos e poços) de captação de água para o abastecimento público. As práticas agrícolas com a utilização indiscriminada de agrotóxicos (VEIGA et al., 2006), o lançamento de efluentes urbanos (PEREIRA, 2004), a supressão da mata ciliar (PINTO; ROMA; BALIEIRO, 2012) e a falta de conservação do solo (MERTEN; MINELLA, 2002) influenciam diretamente na qualidade dessas fontes de água.

Além das fontes de poluição que podem ser facilmente identificadas, onde é possível estabelecer medidas de controle, há também as fontes difusas de poluição, cuja origem não é explicitamente identificável. Essas ocorrem em extensas áreas e, associadas à chuva, chegam aos corpos de água de forma intermitente. A poluição difusa tem sido considerada uma das principais fontes de deterioração da qualidade de águas naturais superficiais e subterrâneas. Atualmente, países com elevados índices de tratamento de esgotos enfrentam desafios relacionados à poluição difusa de águas. Nos Estados Unidos, por exemplo, a ocorrência de fontes difusas é, há tempos, o principal motivo pelo qual muitos rios, lagos e estuários não estão limpos o suficiente para pesca e recreação (USEPA, 1996). Assim, se observa a gravidade do problema ao ver que certas

origens de contaminação da água sequer são reconhecidas, tornando a solução do problema inviável.

Nesse sentido, além da gestão eficaz dos recursos hídricos, o monitoramento contínuo da qualidade da água tem permitido um foco renovado em cursos d'água severamente poluídos, representando um avanço crucial no controle de perdas nas redes de distribuição. A utilização crescente de técnicas estatísticas tem se destacado nesse cenário devido à sua aplicação simplificada e capacidade de interpretação dos dados em tempo real (JUNG e LANSEY, 2015; BORGES et al., 2017; AHN e JUNG, 2019).

## **2.3 Saneamento inadequado**

Assegurar a gestão sustentável da água e saneamento para todos, conforme objetivado pelo ODS 6, é um desafio caracterizado por grande complexidade; envolve a necessidade de articular vários setores governamentais e não governamentais atuando em múltiplos níveis, de conciliar diversos interesses e motivações, colocando a questão da efetividade da governança no topo da agenda política. OECD (2011, p. 17)

O saneamento inadequado tem consequências graves e multifacetadas. Ele não só expõe as comunidades a riscos de saúde, como também contribui significativamente para a poluição das águas superficiais e dos aquíferos. De acordo com o Atlas do Esgoto (2017) da Agência Nacional de Águas (ANA), mais de 110 mil quilômetros de rios no Brasil estão contaminados devido ao contato direto com esgoto não tratado.

O lançamento indiscriminado de esgotos sem tratamento nos corpos d'água pode causar vários problemas, tais como a depleção do oxigênio dissolvido no meio, a formação de bancos de lama, o aumento do nível de nutrientes (especialmente nitrogênio e fósforo, que promovem o perigo de eutrofização), alterações na biota aquática, e o aporte de microorganismos patogênicos, incluindo bactérias, vírus e protozoários, responsáveis pela transmissão de diversas doenças (MARTINIANO, 2011, p.26).

Tundisi e Matsumura-Tundisi (2008) enfatizam o aumento da eutrofização dos ecossistemas aquáticos continentais como consequência de atividades como

a descarga de esgotos domésticos não tratados e descargas industriais e agrícolas. A constituição e a concentração dos esgotos sanitários dependem, entre outros fatores, dos costumes socioeconômicos da população, do clima e também da quantidade e tipo de indústria existente na região. Em locais com baixo consumo per capita de água, normalmente se tem esgotos mais concentrados (Von Sperling, 2005).

Cerca de 99,9% da composição dos esgotos é água, e apenas 0,1% é matéria sólida, composta por duas frações: a inorgânica e a orgânica. A fração inorgânica é composta principalmente de areia, sais e metais, enquanto a fração orgânica é composta por carboidratos, lipídios e proteínas (Callado, 2001).

Tabela 2.2 - Composição Tipica de esgotos

*Capítulo 3 - Características dos esgotos*

PARÂMETRO	Unidade	CONCENTRAÇÃO		
		FORTE	MÉDIO	FRACO
Sólidos Totais	mg/l	1200	720	350
Sól. Dissolvidos Totais	mg/l	850	500	250
Sól. Dissolvidos Fixos	mg/l	525	300	145
Sól. Dissolvidos Voláteis	mg/l	323	200	105
Sól. Suspensos	mg/l	350	220	100
Sól. Suspensos Fixos	mg/l	75	55	20
Sól. Suspensos Voláteis	mg/l	275	165	80
Sól. Sedimentáveis	ml/l	20	10	5
Demandra Bioquímica de Oxigênio - DBO <sub>5dias/20°C</sub>	mg/l	400	220	110
Carbono Orgânico Total	mg/l	290	160	80
Demandra Química de Oxigênio - DQO	mg/l	1000	500	250
Nitrogênio Total	mg/l	85	40	20
Nitrogênio Orgânico	mg/l	35	15	8
Nitrogênio Amoniacal	mg/l	50	25	12
Nitritos	mg/l	0	0	0
Nitratos	mg/l	0	0	0
Fósforo Total	mg/l	15	8	4
Fósforo Orgânico	mg/l	5	3	1
Fósforo Inorgânico	mg/l	10	5	3
Cloreto	mg/l	100	50	30
Sulfato	mg/l	50	30	20
Alcalinidade em CaCO <sub>3</sub>	mg/l	200	100	50
Graxa	mg/l	150	100	50
Coliformes Totais	NMP/100ml	10 <sup>7</sup> a 10 <sup>9</sup>	10 <sup>7</sup> a 10 <sup>8</sup>	10 <sup>6</sup> a 10 <sup>7</sup>
Compostos Orgânicos Voláteis	µg/l	> 400	100 a 400	< 100

FONTE : METCALF & EDDY (1991), p.109.

De acordo com a Organização das Nações Unidas para Educação, Ciência e Cultura (Unesco), hoje no mundo mais de 80% das águas residuais retornam ao meio ambiente sem tratamento (WWAP, 2019).

Observa-se um descompasso entre o crescimento urbano acelerado, com

a expansão das favelas e dos bairros de periferia, e a morosidade na implantação de infraestrutura de tratamento de água e saneamento, além da quase omissão de estratégias para reduzir o uso da água. O processo de urbanização no Brasil foi desigual, com grandes disparidades entre classes sociais e regiões. Consequentemente, os estados brasileiros apresentam desigualdades significativas em relação ao saneamento básico. Isso tem gerado problemas nas cidades, como informalidade na habitação e na infraestrutura de saneamento. Cerca de 1/4 da população brasileira (24,3%), ou aproximadamente 49 milhões de pessoas, continua sem acesso a uma estrutura adequada de saneamento básico (IBGE, 2022).

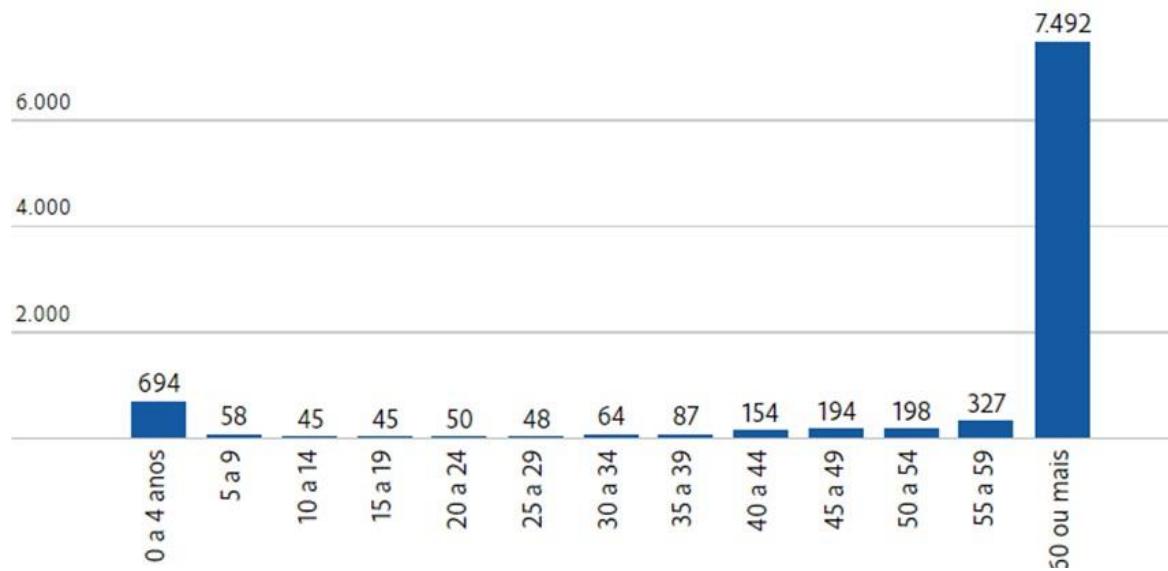
Em 2007, foi sancionada a Lei Nacional de Saneamento Básico (Lei nº 11.445), que estabelece que os serviços públicos de saneamento básico devem ser prestados com base em princípios fundamentais, com destaque para a universalização do acesso. A lei aborda as especificidades de cada serviço de saneamento, conforme definido na legislação: abastecimento de água, esgotamento sanitário, limpeza urbana e manejo de resíduos sólidos, drenagem e manejo das águas pluviais urbanas. No entanto, a lei trata dessas questões sob a perspectiva da integralidade, aplicando-se ao saneamento básico e à integração desses serviços com outras políticas públicas diretamente relacionadas (BRASIL, 2009).

De acordo com a Organização Mundial da Saúde (OMS, 2007), aproximadamente 233 mil pessoas morriam anualmente no Brasil devido à exposição a fatores de risco ambiental, como poluição do ar, água não tratada e falta de infraestrutura urbana, sendo que 19% dessas mortes poderiam ser evitadas com a implementação de políticas públicas eficazes. A pesquisa indicou que a poluição do ar causava a morte de 12,9 mil pessoas por ano. Ademais, com 22% da população vivendo abaixo da linha da pobreza, a ausência de água tratada e esgotamento sanitário resultava na morte de 15 mil brasileiros anualmente. Assim, a qualidade da água é um fator crucial para um saneamento adequado. No entanto, o monitoramento das águas consumidas por muitos brasileiros continua tão inadequado quanto o tratamento desses recursos hídricos. Isso é evidente na Figura 2.3, que mostra um total de 9.463 óbitos atribuídos a fontes de água inseguras, saneamento inseguro e falta de higiene nos últimos anos (IBGE, 2022).

Figura 2.3 - Mortes atribuídas a saneamento inseguro e falta de higiene.

### Mortes atribuídas a saneamento inseguro e falta de higiene

Número de óbitos por faixa etária no Brasil, em 2022



Fonte: IBGE, 2022.

A ausência de saneamento básico não só expõe as comunidades a sérios riscos de saúde, mas também provoca a poluição das águas superficiais e dos aquíferos, comprometendo a qualidade da água disponível em meio à crescente demanda. Para mitigar esses impactos, é essencial envolver a sociedade em todos os níveis de decisão sobre recursos hídricos e saneamento, garantindo a participação de comunidades tradicionais e vulneráveis, bem como de diversos setores sociais comprometidos com a preservação ambiental.

Para enfrentar os desafios do saneamento inadequado e melhorar a qualidade dos recursos hídricos, é fundamental implementar políticas eficazes e promover uma governança robusta. Melhorar o monitoramento das águas e expandir a supervisão com tecnologias de baixo custo pode ajudar a abordar as lacunas existentes e garantir um futuro sustentável e saudável.

Nesse contexto, nossa plataforma se destaca como uma solução promissora para regiões carentes de tais recursos. Com seu baixo custo e simplicidade de análise, é possível expandir a supervisão de recursos hídricos para inúmeros cursos d'água.

## **2.4 Alterações no ciclo hidrológico**

De acordo com Tundisi (2003), outro fator que agrava a situação dos recursos hídricos no mundo é a interferência antrópica no ciclo hidrológico, o que pode diminuir a vazão de água em alguns países em até 70%. As principais atividades humanas que interferem no ciclo hidrológico incluem a construção de reservatórios, que altera o fluxo natural dos rios; o uso excessivo de águas subterrâneas, que reduz os níveis aquíferos; e a importação e transposição de água entre bacias hidrográficas, que desequilibram os ecossistemas locais. O consumo de água nas atividades humanas varia significativamente entre diversas regiões e países.

Os assentamentos humanos têm desempenhado um papel significativo na alteração do ambiente desde o início da ocupação geográfica. Atualmente, as cidades são uma das formas mais agressivas de modificação do ambiente natural, causando uma série de impactos, como a impermeabilização do solo, o aumento do escoamento superficial e a alteração dos padrões de drenagem. Esses impactos resultam em inundações, assoreamento e erosão, com destaque para mudanças no ciclo hidrológico e nas características naturais de drenagem (Tucci, 2002; Guerra e Mendonça, 2010).

Um dos temas mais debatidos atualmente em diversos níveis é a gestão dos recursos hídricos, especialmente em áreas urbanas, onde são propensas ao desencadeamento de processos de degradação ambiental. Nesse sentido, Oliveira e Herrmann (2006, p. 190) enfatizam que "as cidades hoje são o maior exemplo de degradação ambiental, colocando em risco a segurança e qualidade de vida de sua população, sendo palco de conflitos ecológicos." Portanto, o estudo das bacias hidrográficas como unidades fundamentais de análise ambiental, juntamente com os componentes do sistema hidrológico e seus processos, pode fornecer conclusões sobre a qualidade ambiental do espaço em questão e os efeitos da intervenção humana (Botelho e Silva, 2010).

Nesse cenário, a hidrologia urbana apresenta características distintas em comparação com áreas rurais e florestais, devido à intensidade da ocupação humana e aos níveis de alteração ambiental. Botelho e Silva (2010) destacam que o processo de urbanização, frequentemente sem planejamento adequado, trouxe

complexidades significativas para a gestão das águas urbanas. Segundo Tucci (2003a, p. 12), o ciclo hidrológico natural envolve uma complexa interação de processos físicos, químicos e biológicos. Quando o ser humano interfere nesse sistema ao ocupar o espaço, provoca alterações profundas que podem modificar drasticamente o ciclo, resultando em impactos significativos, muitas vezes irreversíveis, tanto para o homem quanto para o meio ambiente.

Desse modo, torna-se essencial conscientizar a sociedade sobre o ciclo hidrológico. Nossa plataforma, por meio de gráficos e estatísticas em tempo real, mostra a qualidade decadente da água em rios ou cursos d'água específicos. Isso possibilita uma atenção especial ao ecossistema observado e permite que a sociedade tome ações fundamentadas na consciência ambiental gerada pelas informações divulgadas pela plataforma, em parceria com o governo, para melhorar o estado das águas em uma determinada região e assim combater futuras alterações no ciclo hidrológico.

### 3. METODOLOGIA

A partir de um estudo realizado em 1970 pela "National Sanitation Foundation" dos Estados Unidos, a CETESB (Companhia Ambiental do Estado de São Paulo) adaptou e desenvolveu o Índice de Qualidade das Águas (IQA), que incorpora nove variáveis consideradas relevantes para a avaliação da qualidade das águas, tendo como principal finalidade sua utilização para abastecimento público. Assim, utilizaremos esses parâmetros e princípios do IQA para embasar nossa tecnologia.

A qualidade das águas brutas, indicada pelo IQA, pode ser determinada por meio de um cálculo específico, resultando em uma escala que varia de 0 a 100, conforme representado na Tabela 3.1.

Tabela 3.1 – Classificação do IQA

Classificação do IQA - CETESB	
Categoria	Ponderação
ÓTIMA	$79 < \text{IQA} \leq 100$
BOA	$51 < \text{IQA} \leq 79$
REGULAR	$36 < \text{IQA} \leq 51$
RUIM	$19 < \text{IQA} \leq 36$
PÉSSIMA	$\text{IQA} \leq 19$

Fonte: CETESB

O Índice de Qualidade das Águas (IQA) é o principal indicador qualitativo utilizado no Brasil. Ele foi desenvolvido para avaliar a qualidade da água destinada ao abastecimento público, após tratamento convencional. A interpretação dos resultados deve considerar esse uso da água. Por exemplo, um valor baixo de IQA indica má qualidade para abastecimento, mas essa água pode ainda ser adequada para usos menos exigentes, como navegação ou geração de energia.

O IQA é composto por nove parâmetros (ver descrição do parâmetros do IQA), com seus respectivos pesos (w), que foram fixados em função da sua importância para a conformação global da qualidade da água (Tabela abaixo).

Tabela 3.2 - Parâmetros de Qualidade da Água do IQA e respectivo peso.

PARÂMETRO DE QUALIDADE DA ÁGUA	PESO (w)
<u>Oxigênio dissolvido</u>	0,17
<u>Coliformes termotolerantes</u>	0,15
<u>Potencial hidrogeniônico - pH</u>	0,12
<u>Demanda Bioquímica de Oxigênio - DBO5,20</u>	0,10
<u>Temperatura da água</u>	0,10
<u>Nitrogênio total</u>	0,10
<u>Fósforo total</u>	0,10
<u>Turbidez</u>	0,08
<u>Resíduo total</u>	0,08

Fonte: CETESB

### **3.1 Descrição dos Parâmetros do IQA**

O IQA é calculado com base nos seguintes parâmetros: temperatura da água, pH, oxigênio dissolvido, resíduo total, demanda bioquímica de oxigênio, coliformes termotolerantes, nitrogênio total, fósforo total e turbidez. Nesse contexto, a equipe ERA2-D2, ao definir os parâmetros a serem monitorados pela plataforma desenvolvida, considerou os fatores mais relevantes para uma avaliação concreta da poluição em cursos d'água, bem como a facilidade de realizar medições simples e não complexas. Com base na acessibilidade e relevância, foram selecionadas cinco variáveis essenciais: pH, oxigênio dissolvido, condutividade, temperatura e turbidez.

#### **a) Oxigênio Dissolvido**

O oxigênio dissolvido é vital para a preservação da vida aquática, já que vários organismos, como peixes, necessitam de oxigênio para respirar. Águas poluídas por esgotos apresentam baixa concentração de oxigênio dissolvido, pois ele é consumido no processo de decomposição da matéria orgânica. Por outro lado, águas limpas apresentam concentrações mais elevadas, geralmente superiores a 5 mg/L, exceto em condições naturais que possam reduzir esses valores. Águas eutrofizadas (ricas em nutrientes) podem apresentar concentrações superiores a 10 mg/L em uma situação conhecida como supersaturação, comum em lagos e represas com crescimento excessivo de algas. Durante o dia, a fotossíntese aumenta os níveis de oxigênio, mas à noite, a respiração dos organismos pode diminuir consideravelmente esses níveis, causando mortandade de peixes.

Para avaliar esse fator, foi selecionado o sensor de oxigênio dissolvido DO9100, um medidor digital portátil com escala de 0,0-30 mg/L. No entanto, devido ao seu alto custo, ele ainda não foi adquirido, estando apenas na proposta da plataforma para possível aquisição em fases futuras.

Figura 3.1 - sensor de OD Do9100.

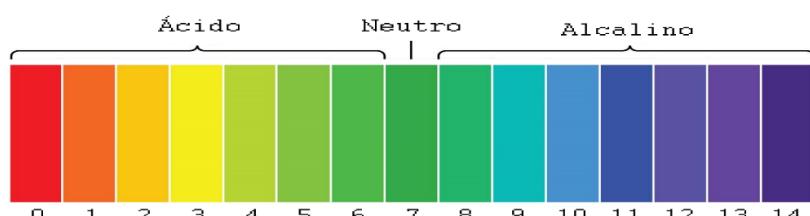


Fonte: Elaborada pelos autores.

### b) Potencial Hidrogeniônico (pH)

O pH é uma unidade de medida que quantifica a alcalinidade ou acidez de uma solução, indicando a concentração de íons de hidrogênio nela contidos. A escala de pH varia de 0 a 14. Um pH de valor igual a 7 é considerado um líquido neutro, uma vez que a água tende a um valor próximo a isso, já valores entre 1 e 6 são considerados líquidos ácidos e valores de pH entre 8 e 14 são considerados líquidos alcalinos, assim como podemos observar na Figura abaixo:

Figura 3.2 - Índice de pH conforme valor de referência.



Fonte: Elaborada pelos autores.

O pH afeta o metabolismo de várias espécies aquáticas. A Resolução CONAMA 357 estabelece que para a proteção da vida aquática o pH deve estar entre 6 e 9. Alterações nos valores de pH também podem aumentar o efeito de substâncias químicas que são tóxicas para os organismos aquáticos, tais como os metais pesados.

O Sensor pH - Módulo De Leitura pH-4502c + Eletrodo Sonda Bnc foi adquirido para a detecção de valor para arduino, medidor e monitoramento de pH líquido 0-14, além do seu módulo de controle de monitoramento.

Figura 3.3 - Sensor pH - Módulo De Leitura pH-4502c + Eletrodo Sonda BNC.



Fonte: Elaborada pelos autores.

### c) Temperatura da água

A temperatura influencia vários parâmetros físico-químicos da água, como a tensão superficial e a viscosidade. Organismos aquáticos são afetados por temperaturas fora de seus limites de tolerância, o que pode impactar seu crescimento e reprodução. Todos os corpos d'água apresentam variações de temperatura ao longo do dia e das estações do ano. No entanto, o lançamento de efluentes com altas temperaturas pode causar impactos

significativos. Para avaliar a temperatura, foi adquirido o sensor DS18B20.

Figura 3.4 - Sensor DS18B20.



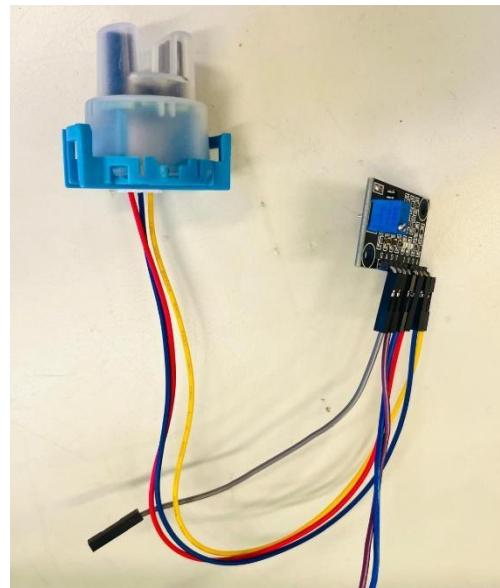
Fonte: Elaborada pelos autores.

#### d) Turbidez

A turbidez mede o grau de atenuação que um feixe de luz sofre ao atravessar a água, refletindo a quantidade de partículas em suspensão, como silte, areia, argila, algas e detritos. Esse fenômeno afeta negativamente a preservação de organismos aquáticos, o uso industrial e as atividades recreativas, além de aumentar os custos de tratamento de água devido à necessidade de mais produtos químicos, como coagulantes.

No Brasil, a Portaria GM/MS Nº 888 de 2021 estabelece limites específicos de turbidez para água potável: 0,5 UT para filtração rápida, 1,0 UT para filtração lenta e 5,0 UT para água na rede de distribuição. Um sensor de turbidez compatível com Arduino foi adquirido para monitorar essas condições em projetos específicos.

Figura 3.5 - Sensor de turbidez.



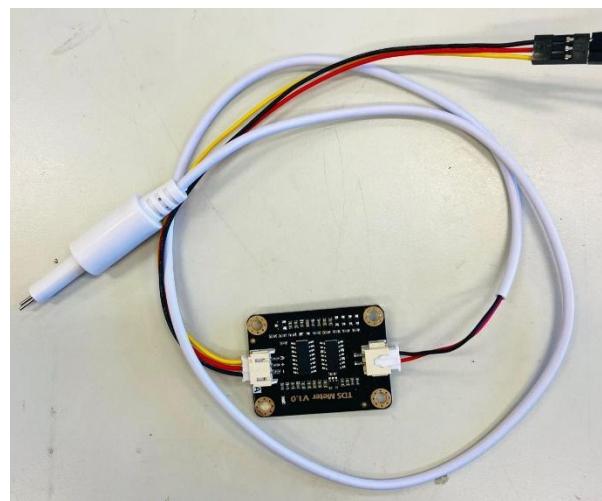
Fonte: Elaborada pelos autores.

#### e) Condutividade

A condutividade da água é uma medida do fluxo de elétrons o qual é facilitado pela presença de íons. Como existe uma relação proporcional entre o teor de sais dissolvidos e a condutividade elétrica, esse parâmetro possibilita a estimativa da quantidade de sais na água. Sendo uma variável de extrema importância, pois influencia diretamente na pureza da água e pode comprometer a sua utilização.

O sensor de condutividade de água para Arduino avalia a capacidade que uma solução tem de conduzir uma corrente elétrica.

Figura 3.6 - Sensor de condutividade.



Fonte: Elaborada pelos autores.

A partir da plataforma podemos medir certos parâmetros significativos ao avaliarmos os desdobramentos da poluição de cursos d'água. Alguns deles:

**f) Coliformes termotolerantes**

As bactérias coliformes termotolerantes ocorrem no trato intestinal de animais de sangue quente e são indicadoras de poluição por esgotos domésticos. Elas não são patogênicas (não causam doenças) mas sua presença em grandes números indicam a possibilidade da existência de microorganismos patogênicos que são responsáveis pela transmissão de doenças de veiculação hídrica (ex: disenteria bacilar, febre tifoide, cólera).

**g) Demanda Bioquímica de Oxigênio (DBO<sub>5,20</sub>)**

A Demanda Bioquímica de Oxigênio representa a quantidade de oxigênio necessário para oxidar a matéria orgânica presente na água através da decomposição microbiana aeróbia. A DBO<sub>5,20</sub> é a quantidade de oxigênio consumido durante 5 dias em uma temperatura de 20°C.

Valores altos de DBO<sub>5,20</sub>, num corpo d'água são provocados geralmente causados pelo lançamento de cargas orgânicas, principalmente esgotos domésticos. A ocorrência de altos

valores deste parâmetro causa uma diminuição dos valores de oxigênio dissolvido na água, o que pode provocar mortandades de peixes e eliminação de outros organismos aquáticos.

#### **h) Nitrogênio Total**

O nitrogênio em corpos d'água pode existir como nitrogênio orgânico, amoniacal, nitrito e nitrato, sendo os nitratos tóxicos para humanos, especialmente crianças, podendo causar a doença metahemoglobinemia. O excesso de compostos de nitrogênio, junto com fósforo, leva à eutrofização, um crescimento excessivo de algas que impacta o abastecimento de água, a recreação e a vida aquática. As principais fontes de nitrogênio incluem esgotos, efluentes industriais, fertilizantes agrícolas, drenagem urbana, fixação biológica por algas e bactérias, e deposição atmosférica via chuva.

#### **i) Fósforo Total**

Do mesmo modo que o nitrogênio, o fósforo é um importante nutriente para os processos biológicos e seu excesso pode causar a eutrofização das águas. Entre as fontes de fósforo destacam-se os esgotos domésticos, pela presença dos detergentes superfosfatados e da própria matéria fecal. A drenagem pluvial de áreas agrícolas e urbanas também é uma fonte significativa de fósforo para os corpos d'água. Entre os efluentes industriais destacam-se os das indústrias de fertilizantes, alimentícias, laticínios, frigoríficos e abatedouros, que resultam em um impacto significativo nos corpos d'água.

#### **j) Resíduo Total**

O resíduo total é a matéria que permanece após a evaporação, secagem ou calcinação da amostra de água durante um determinado tempo e temperatura.

Quando os resíduos sólidos se depositam nos leitos dos corpos d'água podem causar seu assoreamento, que gera problemas para a navegação e pode aumentar o risco de enchentes. Além disso, podem causar danos à vida aquática pois ao se depositarem no leito eles destroem os organismos que vivem nos sedimentos e servem de alimento para outros organismos, além de danificar os locais de

desova de peixes.

### 3.1 Desenvolvimento da plataforma

Nesta fase, foi desenvolvido o código para a programação dos sensores a partir do uso do software de programação do Arduino IDE e da linguagem C + +.

#### a) Microcontrolador

No presente trabalho, foram realizados testes com quatro diferentes microcontroladores. Inicialmente, na etapa regional do torneio, foi utilizado, para a aquisição e o armazenamento dos dados dos sensores, o Arduino Mega 2560 (Figura 3.7). Ele foi escolhido devido à facilidade e familiaridade de manuseio pela equipe, além de ser um componente já disponível nos laboratórios da instituição. Para a comunicação e transmissão das informações coletadas em tempo real, utilizou-se o ESP32 (Figura 3.8), empregado exclusivamente para comunicação via Wi-Fi ou *Bluetooth*, garantindo o envio das informações analisadas para o site desenvolvido pela equipe.

Figura 3.7 – Microcontrolador Arduino MEGA 2560.



Fonte: Elaborada pelos autores.

Figura 3.8 – Microcontrolador ESP32.

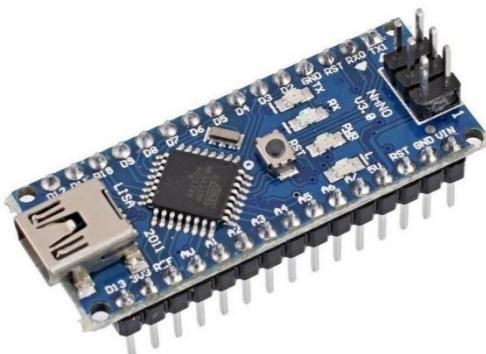


Fonte: Elaborada pelos autores.

Entretanto, após a primeira etapa do TBR, a equipe realizou uma reavaliação dos componentes utilizados no projeto, buscando opções alternativas que desempenhassem as mesmas funções com um custo-benefício mais atrativo. O objetivo foi alinhar o desenvolvimento da plataforma à proposta de um produto acessível e viável economicamente, sem comprometer sua funcionalidade.

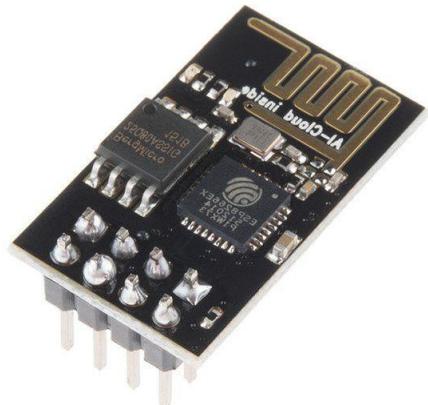
Nesse sentido, com base nesse pressuposto, a equipe decidiu substituir o Arduino Mega 2560 pelo Arduino Nano (Figura 3.9) o ESP32 pelo ESP01 (Figura 3.10). Essa escolha foi motivada por diversos fatores, incluindo o custo reduzido e a compatibilidade desses microcontroladores com as necessidades do projeto. O Arduino Nano oferece a mesma capacidade de aquisição e processamento de dados em um formato mais compacto e econômico. Já o ESP01, baseado no chip ESP8266, possui um conjunto de recursos que o tornam ideal para projetos que requerem conectividade eficiente e simplicidade. Ele conta com algumas entradas e saídas digitais (GPIOs) e suporta comunicação serial (UART), bem como protocolos de comunicação como SPI e I2C. Sua principal vantagem está no módulo Wi-Fi integrado, que permite a conexão direta com redes sem fio sem a necessidade de componentes adicionais, mantendo a transmissão de dados em tempo real com a mesma qualidade, porém com menor custo em relação ao ESP32.

Figura 3.9 – Microcontrolador ESP32.



Fonte: Elaborada pelos autores.

Figura 3.10 – Microcontrolador ESP01.



Fonte: Elaborada pelos autores.

A adoção dessas novas placas reforça o compromisso da equipe em desenvolver uma solução inovadora, acessível e tecnicamente robusta, alinhada à proposta de um protótipo de baixo custo, mas com alto impacto para o monitoramento ambiental.

## 3.2 Ferramentas

### a) Arduino IDE

A Arduino IDE (*Integrated Development Environment*, em português, Ambiente Integrado de Desenvolvimento) é a ferramenta oficial de desenvolvimento para programar placas Arduino e microcontroladores compatíveis, como o ESP01. Ela é fácil de usar e adequada tanto para iniciantes quanto para experientes. A IDE permite escrever e compilar código em C/C++, oferece bibliotecas prontas e exemplos, e possibilita carregar o código para a placa via USB, além de monitorar dados em tempo real. Devido a essas funcionalidades, a equipe decidiu usar a Arduino IDE para desenvolver o código dos sensores e controladores.

### b) Proteus

O Proteus é um software de simulação e design eletrônico que permite a criação, simulação e verificação de circuitos eletrônicos em um ambiente virtual. Ele é amplamente utilizado para testar circuitos e programas antes de implementá-los

fisicamente. Nesse sentido, no seguinte trabalho, a ferramenta Proteus foi encarregada de realizar a simulação do protótipo, já que as bibliotecas do *software* possuem todos os componentes presentes na plataforma.

### c) Fusion

A ferramenta Fusion é um *software* desenvolvido pela Autodesk, que se destaca por suas capacidades avançadas em modelagem, design e simulação. É amplamente utilizado em áreas como engenharia, design de produto e manufatura. Diante disso, no trabalho proposto, a equipe fez uso de tal ferramenta para desenvolver o *design* da estrutura qual o circuito da plataforma ficou acoplado, já que, para obter tal estrutura pela impressora 3D, foi preciso desenvolver antes um modelo que orientasse a máquina.

### d) KiCad

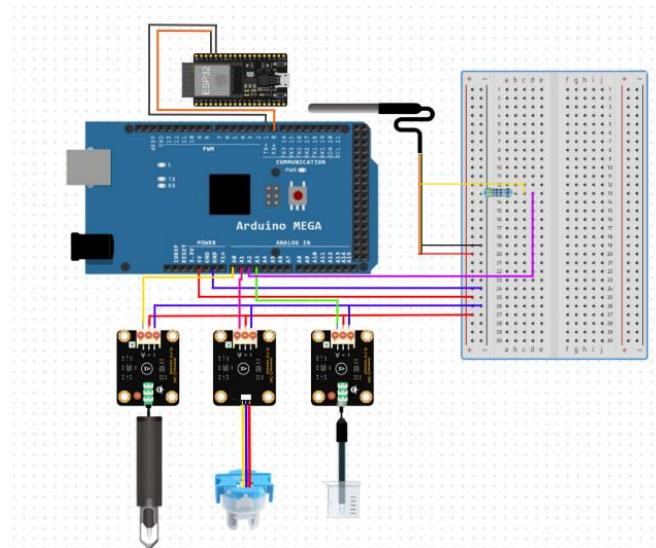
O KiCAD é um *software* de código aberto amplamente utilizada para *design* de circuitos eletrônicos. Ele é voltado para projetistas que precisam criar esquemáticos elétricos e projetar PCBs (*Printed Circuit Boards*, em português, Placa de Circuito Impresso), em portugês, placas de circuito impresso. Reconhecido por sua interface intuitiva e funcionalidades robustas, o KiCAD é amplamente empregado por amadores, estudantes e profissionais da área de eletrônica. A partir disso, a equipe utilizou desta ferramenta com objetivo de desenvolver a PCI (Placa de Circuito Impresso) da plataforma, visando maior organização e melhor funcionalidade do circuito.

## 3.3 Simulação do circuito eletrônico

A simulação do medidor proposto, realizada utilizando a ferramenta Proteus, foi feita há três meses durante a etapa regional do projeto e foi essencial para o entendimento inicial do funcionamento do sistema. Naquele momento, a simulação foi construída com o microcontrolador Arduino MEGA 2560, utilizando sensores para medir pH, turbidez, temperatura e condutividade. O processo de coleta de dados pelos sensores foi representado na simulação, onde os dados eram enviados via comunicação serial para o ESP 32, que transmitia as informações para a plataforma web, onde eram exibidas. Essa simulação foi fundamental para adquirir um conhecimento aprofundado sobre os sensores e os materiais utilizados,

o que facilitou a construção do protótipo físico na etapa regional e tornou o desenvolvimento mais preciso e eficiente.

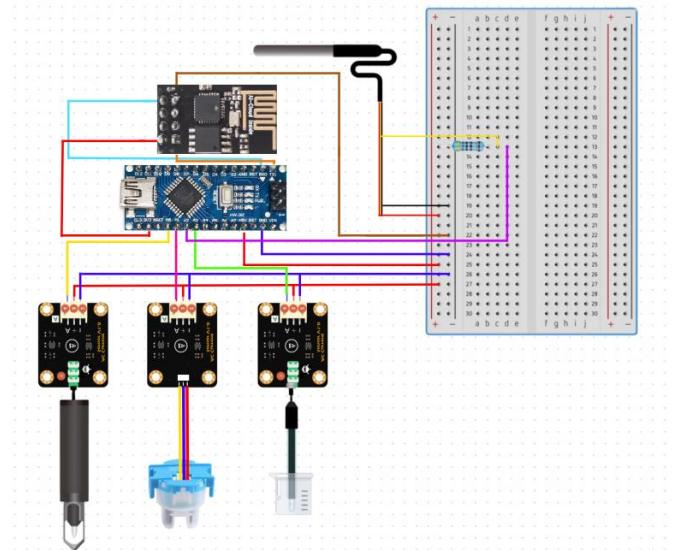
Figura 3.11 - Simulação da plataforma feita no software Proteus para o TBR regional.



Fonte: Elaborada pelos autores.

Agora, com o avanço para a etapa nacional, a equipe optou por utilizar novos microcontroladores, o Arduino Nano e o ESP01, visando uma solução mais econômica e de melhor custo-benefício. Assim, a equipe realizou uma nova simulação para verificar se o sistema funcionaria adequadamente com os novos microcontroladores. A simulação atual foi ajustada para refletir as alterações feitas no hardware e garantir que as modificações não comprometeriam o desempenho do projeto.

Figura 3.12 - Simulação da plataforma feita para o TBR Nacional.



Fonte: Elaborada pelos autores.

Antes de construir o circuito do projeto, foi essencial visualizar o que seria materializado. Por isso, a equipe desenvolveu essa simulação preliminar. Essa etapa permitiu adquirir um conhecimento aprofundado sobre os sensores e materiais utilizados, tornando o desenvolvimento da montagem do projeto real mais preciso e eficiente.

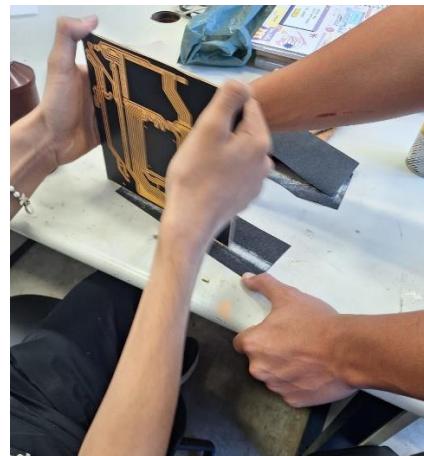
### 3.4 Desenvolvimento da PCI

A criação da PCI envolveu um processo detalhado, passando por várias etapas fundamentais que garantiram um design funcional e organizado. Inicialmente, definiu-se que a placa conteria apenas os seguintes componentes: Arduino Nano, os módulos dos sensores de pH, turbidez, temperatura, um resistor necessário para a conexão do sensor DS18b20 e o ESP01. Essa decisão foi tomada para compactar o projeto, tendo em vista que o módulo do sensor de condutividade poderia ser acoplado com parafusos na caixa que acopla a placa do circuito. Além disso, é importante destacar que, a escolha da equipe de desenvolver uma PCI surgiu a partir da disponibilidade de máquinas que realizam tal trabalho nos laboratórios da instituição qual residem os estudantes.

A partir disso, foi desenvolvida pelo grupo uma nova forma de criar PCBs, de maneira diferente da usual, que é feita com prensa térmica e papel com o design

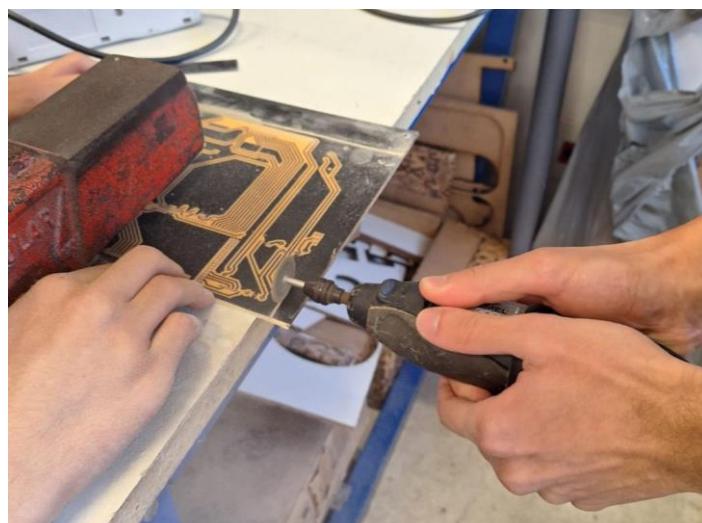
impresso. Placas de fenolite cobreadas foram pintadas com tinta *spray*, e logo após isso, utilizando-se uma máquina CNC (Comando Numérico Computadorizado) a laser, foi feita a retirada da tinta na placa com maior precisão (Figuras 3.13 e 3.14).

Figura 3.13 – PCI cortada e lixada.



Fonte: Elaborada pelos autores.

Figura 3.8 – PCI cortada e lixada.

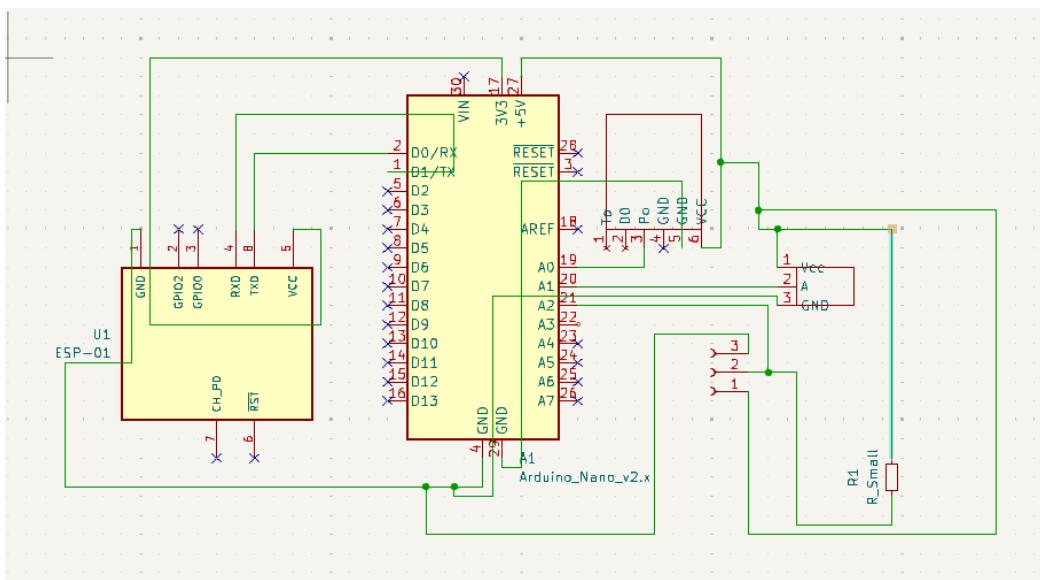


Fonte: Elaborada pelos autores.

O Percloreto de Ferro (III), fórmula química  $\text{FeCl}_3$ , foi usado para a corrosão do cobre, sobrando apenas as trilhas nos lugares desejados. Logo depois a placa foi cortada e lixada, com uma micro retifica e lixas variando entre 80 a 1600. Foram feitos os furos com uma broca de 1, 3 e 4 milímetros de diâmetro com auxílio de uma furadeira de bancada.

Nesse sentido, o primeiro passo foi a elaboração do esquemático no software KiCAD (Figura 3.15). Durante esse estágio, os componentes foram definidos e interligados de acordo com o funcionamento esperado do circuito, que foi desenvolvido previamente a partir da simulação no software proteus e em testes com a protoboard. A partir disso, é válido ressaltar que apenas o arduino Nano possui seu símbolo e *footprint* (na biblioteca do KiCAD), enquanto os desenhos esquemáticos e os *footprints* para os módulos dos sensores de pH, turbidez, temperatura e do ESP01 foram projetados manualmente pela equipe. Esse trabalho incluiu o dimensionamento adequado dos componentes e a atribuição de terminais condizentes com as conexões reais.

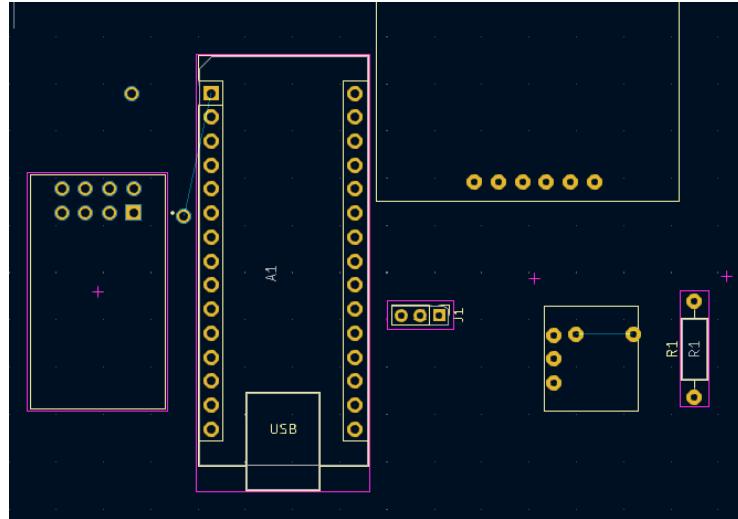
Figura 3.15 - Circuito esquemático da PCI no software KiCAD.



Fonte: Elaborada pelos autores.

Após finalizar o esquemático, iniciou-se a etapa de associação dos componentes aos seus respectivos footprints (Figura 3.16).

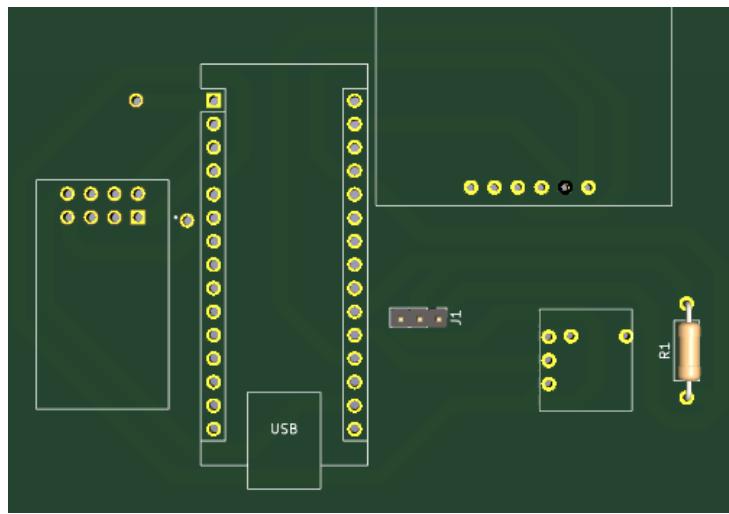
Figura 3.16 – *Footprints* associados à PCI no software KiCAD.



Fonte: Elaborada pelos autores.

Os *footprints* personalizados projetados foram integrados ao *layout*, garantindo que suas dimensões físicas e os pontos de conexão estivessem de acordo com os componentes reais. Essa fase foi crucial para que o circuito pudesse ser transferido corretamente para a visualização 3D da placa (Figura 3.17).

Figura 3.17 – Visualização 3D da PCI no software KiCAD.

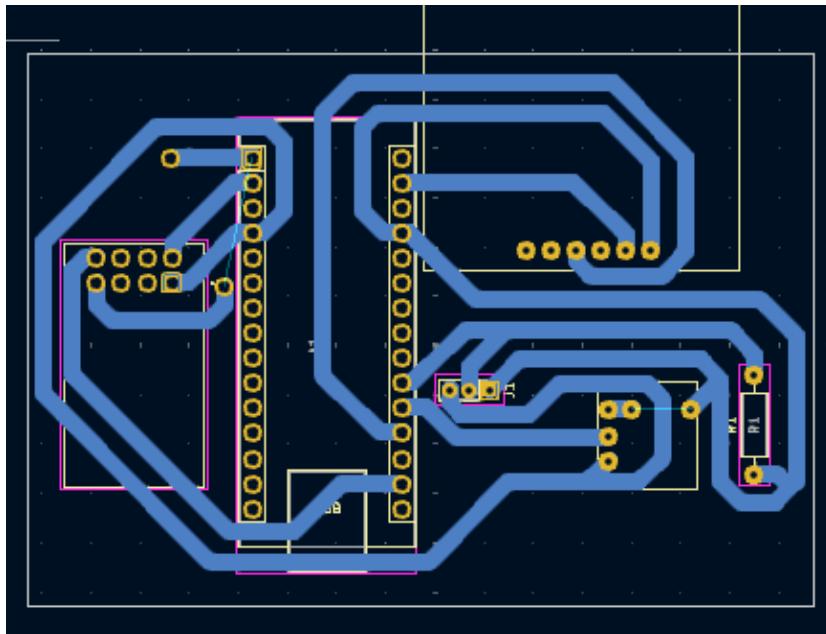


Fonte: Elaborada pelos autores.

Com os *footprints* definidos, passou-se à etapa de roteamento das trilhas (Figura 3.18). As conexões elétricas foram convertidas em trilhas condutoras na PCI, levando em consideração boas práticas de *design*, como minimizar cruzamentos, evitar ruídos e trilhas em 90º graus e organizar as ligações de forma eficiente. Os pinos do ESP01 foram posicionados estratégicamente para facilitar a

comunicação UART com o Arduino Nano, enquanto os sensores foram organizados em áreas específicas para evitar interferências elétricas e garantir a precisão das medições.

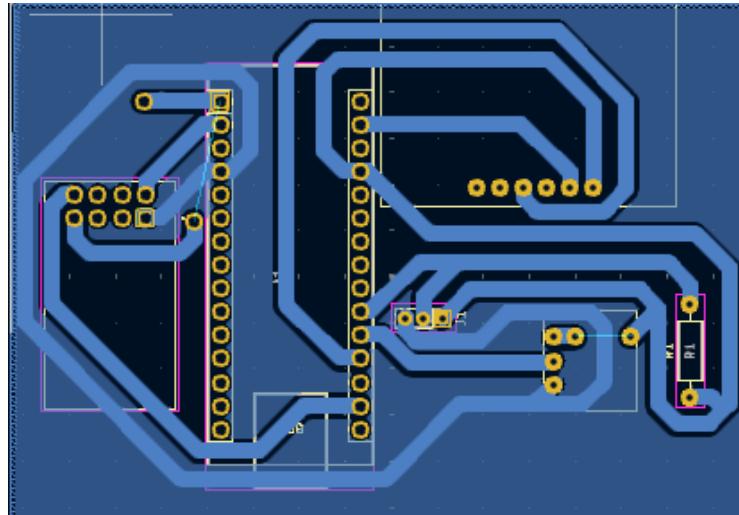
Figura 3.18 – Roteamento das trilhas na PCI no software KiCAD.



Fonte: Elaborada pelos autores.

Além disso, para evitar ruídos e facilitar o fluxo de dados, foram usadas regras de roteamento específicas, como o dimensionamento adequado das trilhas e a inclusão de vias apenas onde necessário. O *layout* da placa também foi otimizado para separar as áreas de sensores e o módulo ESP01, garantindo maior eficiência e clareza na montagem final. Ademais, é importante ressaltar a confecção da malha GND (terra) no projeto (Figura 3.19) (terra). Isso porque, no desenvolvimento de uma PCI no KiCAD, o software disponibiliza a função crucial de fornecer uma referência comum de tensão para todos os componentes eletrônicos da placa. Ela conecta todos os pinos de aterramento dos circuitos e módulos, garantindo que as tensões nos componentes sejam relativas a esse ponto comum. Nesta etapa também decidiu-se o tamanho da placa, sendo ele 56mm de altura e 80 mm de largura. Optou-se por tal tamanho principalmente para gastar menos material, em virtude de que algo maior que isso seria desnecessário, e também, para fabricar algo de cunho profissional, que fosse compacto e condizesse com a proposta do projeto.

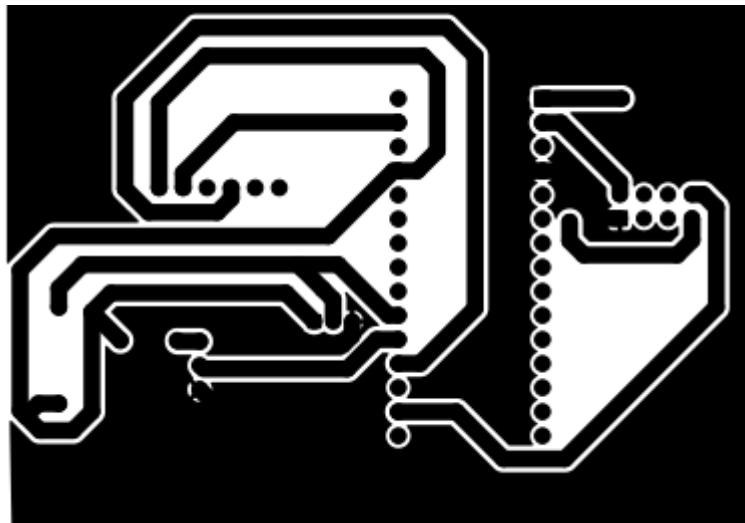
Figura 3.19 – Visualização da malha GND na PCI no software KiCAD.



Fonte: Elaborada pelos autores.

Por fim, a última etapa consistiu na plotagem dos arquivos gerados a partir do *layout* (Figura 3.20).

Figura 3.20 – Arquivo da PCI plotado na camada *Edge Cuts*.

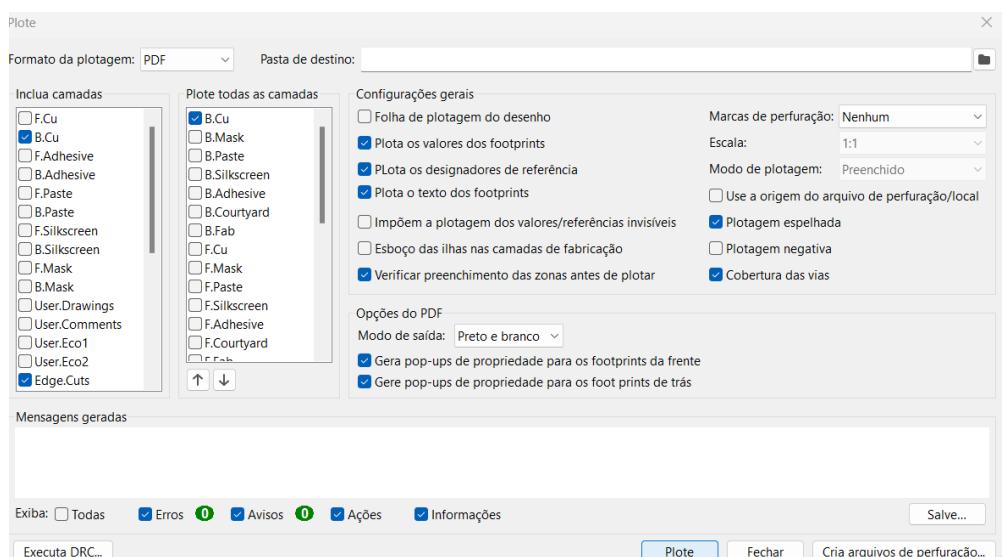


Fonte: Elaborada pelos autores.

Os arquivos da placa devem ser exportados em formato PDF, incluindo camadas essenciais como B.Cu (cobre inferior) e *Edge Cuts* (borda da placa), que são fundamentais para a fabricação precisa da placa (Figura 3.21). A camada B.Cu representa a distribuição de cobre na face inferior da placa, onde as trilhas de cobre conectam os componentes eletrônicos e formam as vias de comunicação entre os pinos dos componentes. Essa camada é crucial para garantir que o circuito funcione corretamente, fornecendo as conexões elétricas necessárias entre os diferentes

pontos do sistema. Já a camada *Edge Cuts* define o contorno físico da placa, determinando seu formato e tamanho para o corte adequado durante a fabricação. Ela garante que a placa tenha as dimensões corretas e seja cortada com precisão. Além disso, é necessário gerar a versão espelhada da placa para garantir a correta disposição dos componentes durante o processo de fabricação, especialmente nas camadas de cobre, assegurando a precisão das trilhas e o posicionamento adequado dos componentes durante a soldagem.

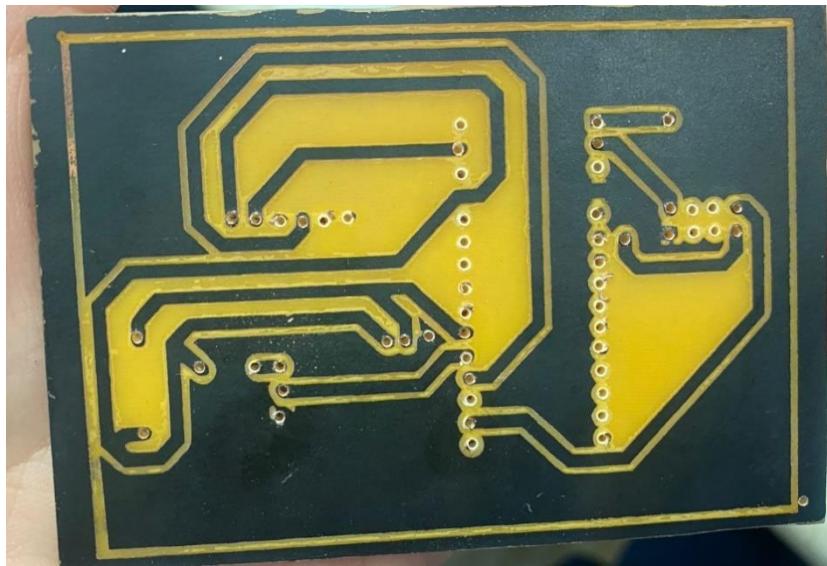
Figura 3.21 – Ferramentas selecionadas para plotagem da PCI.



Fonte: Elaborada pelos autores.

Esse processo detalhado resultou em uma PCI funcional e eficiente, projetada para acomodar os componentes essenciais de forma compacta e organizada (Figura 3.22). A personalização dos desenhos e footprints dos sensores e do ESP01 foi fundamental para atender às necessidades específicas do projeto, garantindo um design único e adaptado aos requisitos do protótipo.

Figura 3.22 – PCI corroída e furada.



Fonte: Elaborada pelos autores.

### 3.5 Confecção do circuito eletrônico

Há três meses, durante a etapa regional do TBR, a equipe enfrentou diversos desafios relacionados à montagem do circuito e ao uso dos componentes. Inicialmente, foi utilizado o microcontrolador ESP32 e sensores diversos para monitorar os parâmetros do sistema. A ideia era realizar a comunicação das portas GPIO e ADC do ESP32 para ler os dados dos sensores e, utilizando o protocolo UART, transmitir as informações para um site. Após avaliar a complexidade dessa abordagem, optou-se por uma solução mais simples: utilizar o ESP32 exclusivamente para coletar os dados processados pelo Arduino e realizar sua transmissão para a plataforma web.

No entanto, outro obstáculo significativo surgiu quando o sensor de turbidez queimou durante os testes. Como o prazo até o torneio era curto e os recursos financeiros eram limitados, a equipe decidiu relatar no documento oficial os resultados obtidos pelo sensor enquanto ele funcionava, planejando adquirir um novo componente caso o projeto avançasse para a etapa nacional.

Atualmente, com o avanço para a etapa nacional do torneio, esse problema foi resolvido com sucesso. Graças ao apoio de patrocinadores, a equipe adquiriu um novo sensor de turbidez, que foi integrado ao sistema. Além disso, para atender ainda mais à proposta de baixo custo do projeto, foram substituídos o ESP32 pelo

ESP01 e o Arduino Mega 2560 pelo Arduino Nano, garantindo uma solução funcional e econômica.

Para a confecção do circuito, a montagem do esquema se iniciou na protoboard, onde foram posicionados os componentes principais: o Arduino Nano, o ESP01 e os sensores. O Arduino Nano foi alimentado via pinos 5V e GND, enquanto o ESP01 foi alimentado com 3.3V e GND, conectando-os às trilhas de alimentação da protoboard para garantir o funcionamento adequado. Em seguida, os sensores de pH, turbidez, temperatura e condutividade foram conectados aos pinos analógicos do Arduino, como A0, A1, A2 e A3, respectivamente, para permitir a leitura dos parâmetros da água. Todos os sensores também foram conectados aos pinos de VCC e GND, garantindo a alimentação necessária. O ESP01 foi utilizado para comunicação sem fio via Wi-Fi, com o pino TX do ESP01 conectado ao pino RX do Arduino Nano e o pino RX do ESP01 ao pino TX do Arduino, estabelecendo a troca de dados entre os dois. A comunicação serial permitiu ao Arduino enviar as informações dos sensores para o ESP01, que então as transmitia para a plataforma web. Após realizar todas as conexões, o circuito foi testado para garantir que as leituras dos sensores fossem válidas e que o ESP01 estivesse corretamente conectado à rede Wi-Fi e enviando os dados de forma eficaz. A montagem na protoboard permitiu realizar ajustes rápidos e testes de funcionamento, antes de finalizar a construção do circuito. O processo envolveu a verificação contínua das conexões, a realização de testes de software e a validação dos resultados obtidos pelos sensores. Assim, foi possível garantir que o sistema estivesse funcionando de acordo com o esperado, com todas as conexões bem estabelecidas e o envio de dados bem-sucedidos.

Os testes realizados com os sensores, incluindo o novo sensor de turbidez, foram bem-sucedidos, e o sistema apresentou resultados consistentes e dentro do esperado (Figura 3.24). A integração do ESP01 e do Arduino Nano foi realizada com sucesso, comprovando a viabilidade da abordagem revisada e permitindo o envio eficiente dos dados para a plataforma web ([Figura 3.25](#)).

Figura 3.24 - Dados avaliados de maneira eficiente.

```
ph=8.92&turbidez=0.50&temperatura=24.94&condutividade=1651.03  
ph=8.95&turbidez=0.97&temperatura=24.94&condutividade=1657.27  
ph=8.95&turbidez=1.08&temperatura=24.94&condutividade=1651.03  
ph=8.93&turbidez=0.85&temperatura=24.94&condutividade=1651.03  
ph=8.94&turbidez=1.31&temperatura=24.94&condutividade=1644.81  
ph=8.91&turbidez=1.00&temperatura=24.88&condutividade=1655.49  
ph=8.90&turbidez=1.09&temperatura=24.94&condutividade=1651.03  
ph=8.92&turbidez=0.88&temperatura=24.94&condutividade=1657.27  
ph=8.92&turbidez=1.29&temperatura=24.94&condutividade=1657.27  
ph=8.89&turbidez=1.34&temperatura=24.88&condutividade=1680.66  
ph=8.84&turbidez=1.99&temperatura=24.94&condutividade=1657.27  
ph=8.86&turbidez=1.99&temperatura=24.88&condutividade=1649.25  
ph=8.84&turbidez=1.99&temperatura=24.94&condutividade=1644.81  
ph=8.83&turbidez=1.16&temperatura=24.88&condutividade=1649.25  
ph=8.84&turbidez=1.05&temperatura=24.94&condutividade=1595.80  
ph=8.80&turbidez=1.28&temperatura=24.94&condutividade=1583.75
```

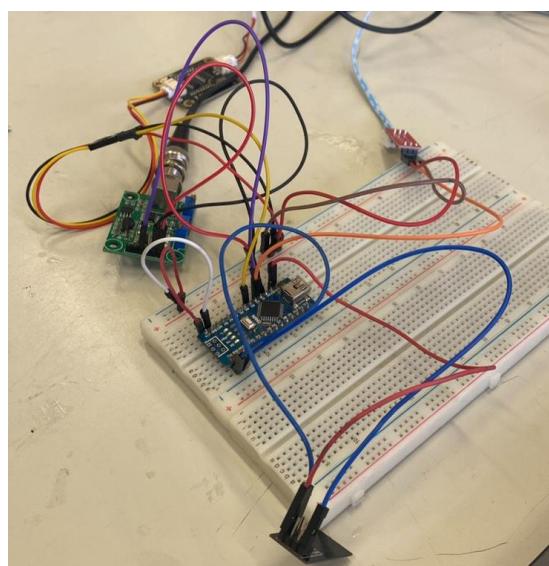
Fonte: Elaborada pelos autores.

Figura 3.25 - Transmissão dos dados para site.

Fonte: Elaborada pelos autores.

A trajetória até o momento reflete o aprendizado contínuo da equipe, que, com criatividade e resiliência, solucionou os problemas enfrentados e aprimorou o projeto para esta nova etapa.

Figura 3.26 - Circuito eletrônico completo.

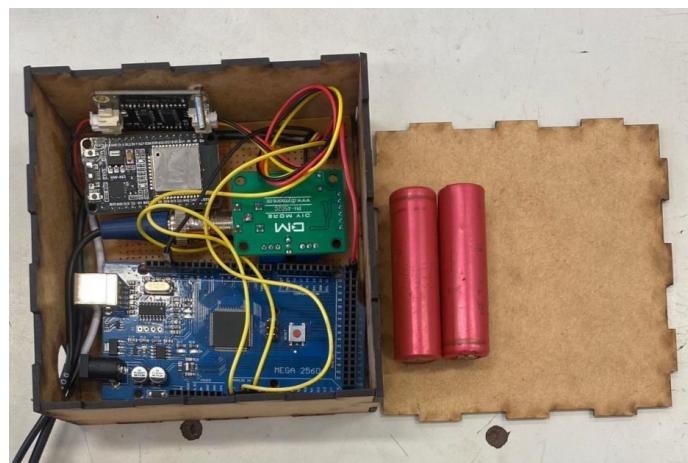


Fonte: Elaborada pelos autores.

### 3.6 Arquitetura da plataforma

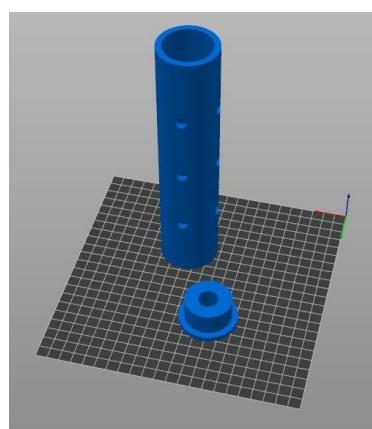
Durante a etapa regional do torneio brasileiro de robótica, a equipe definiu a estrutura inicial da plataforma. A carcaça foi projetada em duas partes: para o circuito eletrônico, que inclui microcontroladores e módulos, foi utilizada uma caixa feita em MDF, com o objetivo de proteger os componentes que não poderiam ter contato com a água (Figura 3.27) Equanto os sensores foram acoplados em um modelo de impressão 3D, projetado na ferramenta Fusion (Figura 3.28), para garantir sua imersão na água. A escolha pela utilização de impressão 3D foi motivada pela disponibilidade da máquina no laboratório e pelos resultados satisfatórios obtidos, como resistência à água, bom alcance de monitoramento e flexibilidade estrutural.

Figura 3.27 – Esquema eletrônico da plataforma acoplado na caixa de MDF.



Fonte: Elaborada pelos autores.

Figura 3.28 – Primeiro modelo 3D da plataforma desenvolvido pela equipe.



Fonte: Elaborada pelos autores.

No entanto, após a participação na etapa regional e com o avanço para a fase nacional, a equipe revisitou o design do protótipo devido a problemas identificados durante os testes. Foi observado que, no modelo anterior, as paredes do projeto estavam muito grossas (5mm), e que para o futuro modelo seria interessante e inteligente reduzir para 3mm, visando maior qualidade e redução de materiais. Ademais, os sensores estavam dispostos muito próximos uns dos outros, o que comprometia a precisão das medições devido a interferências entre os módulos dos sensores, como o de pH e condutividade. Além disso, as aberturas para entrada de água eram limitadas, restringindo o fluxo adequado e prejudicando a análise dos parâmetros aquáticos (Figura 3.29).

Figura 3.29 – Protótipo utilizado na etapa regional.



Fonte: Elaborada pelos autores.

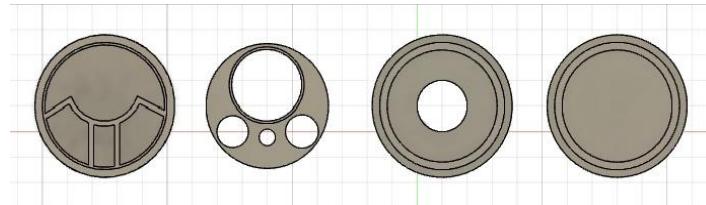
Com base nesses aprendizados, a equipe desenvolveu um novo modelo para a fase nacional (Figura 3.30). O projeto atualizado incluiu divisórias na estrutura (Figura 3.31), que separa cada sensor de acordo com seu tamanho e garante o distanciamento adequado entre eles, e evita interferências nas medições.

Figura 3.30 – Modelo 3D da plataforma desenvolvida Para o TBR nacional.



Fonte: Elaborada pelos autores.

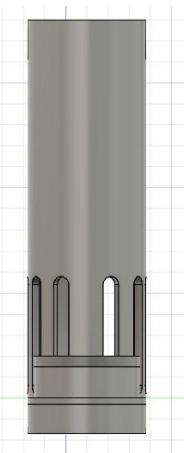
Figura 3.31 – Vista superior do protótipo contendo as divisórias.



Fonte: Elaborada pelos autores.

Além disso, também foram realizadas alterações nas aberturas do protótipo, permitindo um fluxo de água mais eficiente e em conjunto foi desenvolvido um compartimento na parte inferior do protótipo que permite a acoplação de peso para que a plataforma possa afundar na água sem auxílio externo e permita maior imersão dos sensores (Figura 3.32).

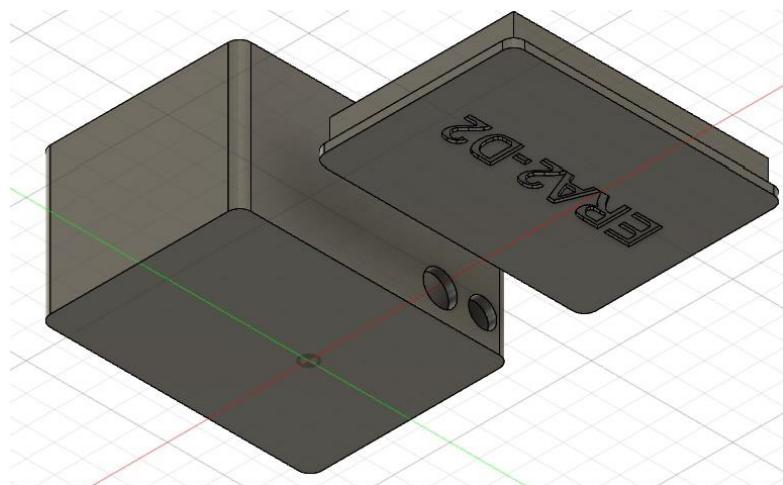
Figura 3.32 – Vista lateral direita do protótipo que contem a visualização do compartimento que acoplará peso.



Fonte: Elaborada pelos autores.

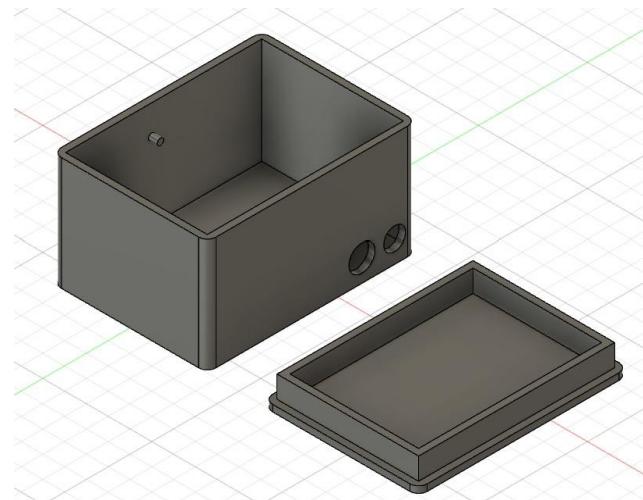
Ademais, a caixa de MDF utilizada na etapa regional foi substituída por uma estrutura completamente impressa em 3D (Figuras 3.33 e 3.34), nas medidas de 60mm por 85mm, que foram determinadas a partir do tamanho da PCI que ficaria acoplada na caixa. conferindo maior resistência, precisão no encaixe dos componentes e melhor acabamento.Também é válido destacar qual a motivação pela cor do projeto em que os sensores residem, sendo ela laranja. Optou-se por tal escolha com a intenção de manter o cilindro que mantém os sensores imersos na água de maneira bem visível, evitando possíveis interferências no processo e facilitando a localização do protótipo em meio a diferentes ambientes aquáticos.

Figura 3.33 – Vista isométrica 3D da caixa que acoplará a placa.



Fonte: Elaborada pelos autores.

Figura 3.34 – Vista isométrica 3D da caixa que acoplará a placa.



Fonte: Elaborada pelos autores.

Figura 3.35 – Projeto da caixa que acopla a placa pronto.



Fonte: Elaborada pelos autores.

Em relação ao circuito eletrônico, como já citado anteriormente, a montagem permaneceu utilizando uma placa PCB para organizar os microcontroladores e módulos. Essa escolha continuou a oferecer vantagens, como praticidade na soldagem dos componentes e maior organização do sistema eletrônico. Para o sensor de condutividade, foi mantida a fixação com dois parafusos de três milímetros, enquanto os demais componentes foram soldados na placa de cobre, visando compactar o protótipo e otimizar seu desempenho. (Figura 3.28). Além disso, para a alimentação do circuito foram utilizadas 2 baterias de lítio em série, que juntas somam a tensão, totalizando 8.4 volts ( $4.2 + 4.2$  V). O suporte para

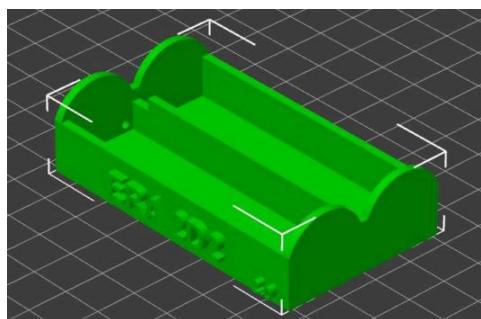
as baterias foi modelado em 3D também a partir do programa de modelação fusion, e molas para a fixação foram presas no suporte (Figura 3.29).

Figura 3.36 – PCB com os componentes soldados.



Fonte: Elaborada pelos autores.

Figura 3.37 – Suporte para baterias.



Fonte: Elaborada pelos autores.

Nesse sentido, a reestruturação do protótipo representou um avanço significativo no desenvolvimento do projeto, garantindo maior confiabilidade e eficiência nas medições, atendendo aos requisitos da etapa nacional do torneio (Figura 3.38).

Figura 3.38 – Protótipo 3D modelado para o TBR nacional.



Fonte: Elaborada pelos autores.

### 3.7 Programação

Nesta etapa, descreve-se o processo de programação dos dados coletados e sua validação por meio dos sensores físico-químicos específicos, além do código referente a transmissão dessas informações pelo microcontrolador ESP 32, utilizando o protocolo UART.

#### a) Programação dos sensores

Ao iniciar o desenvolvimento das funcionalidades para todos os componentes, utilizando a interface do Arduino IDE, foi desenvolvida, primeiramente, a programação dos sensores utilizados na plataforma. As atividades a seguir incluem a leitura dos parâmetros avaliados e a configuração da conexão entre o ESP01 e o Arduino Nano. Vale destacar que todos os códigos descritos neste documento podem ser encontrados no apêndice.

No início de qualquer programação em Arduino, é comum iniciar o código com a inclusão das bibliotecas e a declaração das portas que farão a comunicação entre os sensores o arduino (Figuras 3.39 e 3.40), além das variáveis e algorítimos necessários para o funcionamento do projeto, (Figuras 3.41 e 3.42). Essas

bibliotecas fornecem as ferramentas essenciais para interagir com os diversos componentes, como sensores e atuadores, enquanto as variáveis armazenam os dados que serão utilizados ao longo do programa.

Figura 3.39 – Declaração das bibliotecas.

```
// Bibliotecas.  
#include <Wire.h>           // Biblioteca auxiliar para o protocolo de comunicação UART.  
#include <OneWire.h>          // Biblioteca auxiliar para o sensor de temperatura.  
#include <DallasTemperature.h> // Biblioteca auxiliar para o sensor de temperatura.
```

Fonte: Elaborada pelos autores.

As bibliotecas apresentadas acima são conjuntos de códigos pré-desenvolvidos, cujo objetivo é simplificar e abstrair o processo de criação de aplicações. Elas podem ser nativas da linguagem de programação do Arduino ou fornecidas pelos fabricantes dos dispositivos, facilitando a implementação de diversas funções sem a necessidade de programar tudo do zero. Todas as bibliotecas utilizadas pela equipe foram encontradas nos datasheets dos sensores, documentos técnicos fornecidos pelos fabricantes de componentes eletrônicos, que contêm todas as informações necessárias para entender o funcionamento de cada componente.

Figura 3.40 – Declaração das portas.

```
// Declaração dos pinos.  
#define pHPin A0              // pH.  
#define turbidezPin A1          // Turbidez.  
#define temperaturaPin A2        // Sinal do sensor DS18B20 (Temperatura).  
#define condutividadePin A3       // Condutividade.
```

Fonte: Elaborada pelos autores.

Aqui são definidos os pinos do Arduino aos quais os sensores estão conectados. Os pinos A0, A1, A2 e A3 são usados para os sensores de pH, turbidez, temperatura (DS18B20) e condutividade, respectivamente. Optou-se por utilizar portas analógicas (A0, A1, A2 e A3) porque todos os sensores usados, exceto o de temperatura, emitem uma tensão variável como saída, a qual precisa ser convertida em um valor digital pelo conversor ADC (*Analog-to-Digital Converter*) do Arduino. O microcontrolador lê esses valores analógicos e os converte em uma faixa numérica de 0 a 1023 para uma alimentação de 5V em que, cada valor corresponde a 4,88mV, isso é 5/1024, o que permite o processamento dos dados pelo programa.

Figura 3.41 – Variáveis.

```
//-----  
// Variáveis Auxiliares.  
bool flag = 1, corrigecodut = 1;  
unsigned long tempo = 0;  
  
// Variáveis dos sensores.  
double pH = 0;  
double NTU = 0;  
float TEMP = 0;  
  
int analogBuffer[SCOUNT]; // Armazena os valores analógicos lidos do sensor.  
int analogBufferTemp[SCOUNT]; // Cópia temporária do buffer para processamento.  
int analogBufferIndex = 0; // Índice do buffer.  
int copyIndex = 0;  
  
double tensaoEstavel = 0;  
double TDS = 0;
```

Fonte: Elaborada pelos autores.

Na imagem acima, são declaradas as variáveis que armazenarão as leituras dos sensores e ajudarão no processamento dos dados. A variável “*flag*” controla o fluxo de leituras dos sensores, enquanto outras variáveis armazenam os valores dos sensores (pH, turbidez, temperatura e condutividade), além dos “buffers” que servem como amortecedores para armazenar as leituras analógicas e selecionar os melhores parâmetros para tornar a precisão das medições mais eficazes.

Figura 3.42 – Algorítimo.

```
//-----  
// Filtragem média do sensor de condutividade.  
int getMedianNum(int bArray[], int iFilterLen) {  
    int bTab[iFilterLen];  
    for (int i = 0; i < iFilterLen; i++) {  
        bTab[i] = bArray[i];  
    }  
    int i, j, bTemp;  
    for (j = 0; j < iFilterLen - 1; j++) {  
        for (i = 0; i < iFilterLen - j - 1; i++) {  
            if (bTab[i] > bTab[i + 1]) {  
                bTemp = bTab[i];  
                bTab[i] = bTab[i + 1];  
                bTab[i + 1] = bTemp;  
            }  
        }  
    }  
    if ((iFilterLen & 1) > 0) bTemp = bTab[(iFilterLen - 1) / 2];  
    else bTemp = (bTab[iFilterLen / 2] + bTab[iFilterLen / 2 - 1]) / 2;  
  
    return bTemp;  
}
```

Fonte: Elaborada pelos autores.

Nessa parte do código, representada pela imagem acima, há a presença de uma função, sendo ela, “*getMedianNum*”, que realiza a filtragem das leituras do sensor de condutividade utilizando o algoritmo de mediana. As leituras do sensor são armazenadas em um array, e depois são ordenadas em ordem crescente. A mediana é calculada para minimizar a influência de leituras extremas ou ruidosas,

proporcionando um valor de leitura mais confiável.

Além disso, é importante entender que, no Arduino, o código é estruturado em dois ciclos básicos: *SETUP* e *LOOP*.

A função *SETUP* (Figura 3.43) é executada uma única vez, quando o sistema é iniciado. Ela é responsável pela configuração dos pinos a serem utilizados, pela definição da velocidade de comunicação serial, pela configuração dos pinos digitais e analógicos e pela inicialização do Wi-Fi no ESP01.

```
//-----  
void setup() {  
  
    // Inicializa o Monitor Serial para a conexão com o ESP8266 01.  
    Serial.begin(115200);  
  
    // Define os pinos como entrada.  
    pinMode(pHPin, INPUT);           // pH.  
    pinMode(turbidezPin, INPUT);     //Turbidez.  
    pinMode(temperaturaPin, INPUT);   //Temperatura.  
    pinMode(condutividadePin, INPUT); //Condutividade.  
  
    // Inicializações do sensor de temperatura.  
    sensors.begin();                //Temperatura.  
    sensors.getAddress(sensor1, 0);   //Temperatura.  
}  
]  
----- SETUP -----
```

Figura 3.43 – Void setup

Fonte: Elaborada pelos autores.

A partir disso, a função *LOOP* (Figura 3.44) é responsável pelas atividades contínuas durante o funcionamento do protótipo. Ela define os momentos e as condições em que os dispositivos devem operar, permitindo a coleta de dados e a ativação dos sensores conforme necessário. No código desenvolvido pela equipe, a função *LOOP* serve como uma chamada para as funções de leitura. A cada iteração, as funções de leitura de pH, turbidez, temperatura e condutividade são chamadas, permitindo a leitura dos dados analisados e sua exibição no monitor serial. Além disso, na mesma função há a presença das variáveis que contêm os valores lidos dos sensores (Ph, NTU, TEMP e TDS) que serão como argumentos para a função “Conexao()”, que formata e envia os dados para o ESP8266, como descrito anteriormente.

Figura 3.44 – Void loop.

```
//---  
void loop() {  
    // Lê os dados continuamente e envia a cada 1 segundo.  
    tempo = millis();  
    while (millis() - tempo < 1000) {  
        PH();  
        Turbidez();  
        Temperatura();  
        Condutividade();  
    }  
    Conexao(pH, NTU, TEMP, TDS);  
    flag = 1;  
}
```

Fonte: Elaborada pelos autores.

A seguir, é necessário detalhar cada função retornada no *LOOP*, conforme exemplificado na figura acima. Como mencionado anteriormente, o código referente a cada sensor foi baseado no datasheet do fabricante, com algumas alterações feitas pela equipe para otimizar a programação. A primeira função executada no loop é a do sensor de pH. Inicialmente, ela realiza 10 leituras, e em seguida, organiza esses valores em ordem crescente usando o método bubble sort. Após isso, é descartado as duas menores e as duas maiores leituras, para eliminar possíveis ruídos ou inconsistências, e então, é calculado a média dos valores restantes. A partir deste cálculo, é utilizada uma equação de primeiro grau

$(y = ax + b)$ , na qual “ $a$ ” equivale a -5,7 e “ $b$ ” equivale ao fator de calibração realizado de forma empírica, utilizando um pH já conhecido, enquanto “ $x$ ” se refere ao valor da medição.

Figura 3.45 – Função do sensor de pH.

```

//-----FUNÇÕES-----
void pH() {

    const float calibracao = 22.0;
    unsigned long somaLeiturasValidas = 0;
    unsigned int leituras[10];

    // Realiza 10 leituras do sensor.
    for (byte i = 0; i < 10; i++) {
        leituras[i] = analogRead(pHPin);
    }

    // Organiza os valores em ordem crescente.
    for (byte i = 0; i < 9; i++) {
        for (byte j = i + 1; j < 10; j++) {
            if (leituras[i] > leituras[j]) {
                int temp = leituras[i];
                leituras[i] = leituras[j];
                leituras[j] = temp;
            }
        }
    }

    for (byte i = 2; i < 8; i++) somaLeiturasValidas += leituras[i]; // Descarta as duas primeiras e duas últimas leituras e soma.
    double pHVol = somaLeiturasValidas * leituraAnalogica / 6; // Faz a média entre as 6 leituras restantes.
    pH = -5.70 * pHVol + calibracao; // Retorna um valor de pH.
}

```

Fonte: Elaborada pelos autores.

A segunda função, referente ao sensor de turbidez também faz uma leitura analógica, mas, ao contrário do sensor de pH, a conversão do sinal lido para o valor final de turbidez em NTU (Unidade Nephelométrica de Turbidez) é feita por meio de uma fórmula de calibração específica. O valor da leitura analógica é convertido em uma tensão proporcional e, com base nessa tensão, é aplicado o cálculo que gera o valor de turbidez. O dado adquirido é exibido no monitor serial.

Figura 3.46 – Função do sensor de turbidez.

```

void Turbidez() {

    double tensao = (analogRead(turbidezPin) * leituraAnalogica);
    NTU = (double)((-1.1204 * tensao * tensao) + (5.7423 * tensao) - 4.3529); // Calculo NTU.
}

```

Fonte: Elaborada pelos autores.

O terceiro componente trabalho é o sensor de temperatura DS18B20, que se comunica com o Arduino utilizando o protocolo *OneWire*. Diferente dos sensores de pH e turbidez, que geram sinais analógicos, o DS18B20 transmite dados digitais. O Arduino solicita as leituras de temperatura do sensor através da função “*requestTemperatures()*”, e o valor da temperatura em graus Celsius é obtido

utilizando a função “*getTempC()*”. Este valor de temperatura é armazenado na variável “TEMP” para ser utilizado no processo de compensação da condutividade posteriormente. O valor obtido é exibido no monitor serial.

Figura 3.47 – Função do sensor de temperatura.

```
void Temperatura() {
    sensors.requestTemperatures(); // Le a informação do sensor.
    TEMP = sensors.getTempC(sensor1);
}
```

Fonte: Elaborada pelos autores.

Por fim, o último componente inserido nessa programação foi o sensor de condutividade. Ele lê os valores do parâmetro, utilizando um algoritmo de filtragem para obter um valor e em seguida apresenta variações significativas de leitura, causadas por ruídos ou interferências externas. Para contornar isso, o código utiliza um buffer, no qual várias leituras do sensor são armazenadas.

Após reunir essas leituras, o sensor realiza uma compensação de temperatura e converte a leitura de TDS (Total de Sólidos Dissolvidos), em ppm (partes por milhão). É válido destacar que essa compensação de temperatura é realizada a partir das informações fornecidas pelo sensor ds18b20 que também se encontra na plataforma.

Figura 3.48 – Função do sensor de condutividade.

```
void Condutividade() {
    analogBuffer[analogBufferIndex] = analogRead(condutividadePin); //Lê o valor analógico e armazena no buffer.
    analogBufferIndex++;

    if (analogBufferIndex == SCOUNT) analogBufferIndex = 0;

    if(flag) {
        flag = 0;
        for (copyIndex = 0; copyIndex < SCOUNT; copyIndex++) {
            analogBufferTemp[copyIndex] = analogBuffer[copyIndex];
        }
    }

    // Lê o valor analógico mais estável pelo algoritmo de filtragem média e converte para um valor de tensão.
    tensaoEstavel = getMedianNum(analogBufferTemp, SCOUNT) * leituraAnaloga;

    // Compensação do valor de condutividade em função da temperatura.
    float compensationCoefficient = 1.0 + 0.02 * (TEMP - 25.0); // Resultadofinal(25°C) = Resultadofinal(atual)/(1,0 + 0,02 * (temperatura - 25,0));
    float compensation = tensaoEstavel / compensationCoefficient;

    // Converte o valor da tensão em um valor de condutividade. O valor calculado é multiplicado por 0.5 para converter para TDS em ppm.
    TDS = ((133.42 * compensation * compensation * compensation) - 255.86 * compensation + 857.39 * compensation) * 0.5;
}
```

Fonte: Elaborada pelos autores.

A partir da explicação de como cada sensor lê os valores recebidos, é

importante explicar como o arduino realiza a conexão com o ESP01 na programação da primeira placa microcontroladora. (Figura 3.40).

Figura 3.49 – Função da conexão entre o arduino e ESP01.

```
void Conexao(double pH, double turbidez, float temperatura, double condutividade) {  
    // Envia as informações lidas para o ESP8266 01.  
    String mensagem = "ph=" + String(pH) + "&turbidez=" + String(turbidez) + "&temperatura=" + String(temperatura) + "&condutividade=" + String(condutividade);  
    Serial.print(mensagem);  
    Serial.println();  
}
```

Fonte: Elaborada pelos autores.

A função conexão, exemplificada acima, tem como objetivo enviar as informações coletadas pelos sensores para o ESP8266 (ESP01). Dentro dessa função, os dados de cada sensor (pH, turbidez, temperatura e condutividade) são formatados como uma *string* (variável que armazena caracteres consecutivos), e são enviados via comunicação serial, enquanto as próximas linhas, usam a função “*Serial.print()*” para enviar a *string* mensagem para o monitor serial do Arduino. O “*Serial.println()*” adiciona uma quebra de linha ao final da *string*, o que ajuda na formatação dos dados enviados, caso seja necessário para o processamento.

O ESP8266 irá ler essa mensagem através da comunicação serial, e a partir do código implementado no ESP, transmitirá os dados para a plataforma web desenvolvida pela equipe.

### b) Programação da conexão entre o ESP01 e arduino nano.

Agora que foram descritas as programações dos sensores e a conexão entre as duas placas controladoras, é necessário explicar como as informações adquiridas pelo Arduino são enviadas ao ESP32 e transmitidas para a plataforma web desenvolvida pela equipe.

Figura 3.50 – Declaração das bibliotecas.

```
#include <Arduino.h>  
#include <ESP8266WiFi.h>  
#include <ESP8266WiFiMulti.h>  
#include <ESP8266HTTPClient.h>  
#include <WiFiClient.h>
```

Fonte: Elaborada pelos autores.

Conforme exemplificado na figura acima, as bibliotecas necessárias para o funcionamento do código são importadas. A biblioteca “Arduino.h” é usada como a base para programar o microcontrolador, fornecendo funcionalidades essenciais para a comunicação e manipulação de hardware, já a a “ESP8266WiFi.h” permite que o ESP8266 se conecte a uma rede Wi-Fi, enquanto a “ESP8266WiFiMulti.h” possibilita que o dispositivo se conecte a várias redes Wi-Fi, tentando uma conexão até que consiga estabelecer uma com sucesso. Além dessas, a “ESP8266HTTPClient.h” é utilizada para enviar requisições HTTP do tipo GET, permitindo a comunicação com um servidor, e a bblioteca “WiFiClient.h” fornece funções para gerenciar a comunicação TCP/IP, criando uma conexão cliente com o servidor.

Logo após, como representado na figura abaixo, são definidas algumas constantes. A constante “*USE\_SERIAL*” é configurada como “Serial”, isso é, um objeto que permite a comunicação serial entre o microcontrolador e um computador ou outro dispositivo. As variáveis “STASSID” e “STAPSK” são definidas com os dados da rede Wi-Fi à qual o ESP8266 se conectará, ou seja, o nome da rede (SSID) e a senha (STAPSK). A constante “*SERVERNAME*” armazena a URL do servidor, para onde os dados serão enviados. Esta URL é composta por um endereço específico (<http://makesoft.net.br/src/getinfo.php?>), ao qual será concatenado um parâmetro que representa os dados a serem enviados.

Figura 3.51 – Definição de constantes.

```
#define USE_SERIAL Serial
#define STASSID "LabMaker" //LabMaker
#define STAPSK "53471409000152" //53471409000152
#define SERVERNAME "http://makesoft.net.br/src/getinfo.php?"
ESP8266WiFiMulti WiFiMulti;
```

Fonte: Elaborada pelos autores.

Dentro do código, é criado um objeto “WiFiMulti” da classe “ESP8266WiFiMulti”, que possibilita ao dispositivo se conectar a várias redes Wi-Fi. Nesse sentido, o ESP01 tentará se conectar à rede configurada nas constantes “STASSID” e “STAPSK” citadas anteriormente. Caso uma rede Wi-Fi adicional seja configurada, o objeto “WiFiMulti” tentará se conectar à primeira rede disponível.

Figura 3.52 – Função para comandos do ESP01.

```
void sendEsp(String baglink){
    WiFiClient client;
    HTTPClient http;

    String finalLink = SERVERNAME + baglink;
    if (http.begin(client, finalLink)) { // HTTP
        int httpCode = http.GET();

        if (httpCode > 0) {
            USE_SERIAL.printf("[HTTP] GET... código: %d\n", httpCode);

            if (httpCode == HTTP_CODE_OK || httpCode == HTTP_CODE_MOVED_PERMANENTLY) {
                String payload = http.getString();
                USE_SERIAL.println(payload);
            }
        } else {
            USE_SERIAL.printf("[HTTP] GET... falhou, erro: %s\n", http.errorToString(httpCode).c_str());
        }

        http.end();
    } else {
        USE_SERIAL.println("[HTTP] Impossível conectar");
    }
}
```

Fonte: Elaborada pelos autores.

Ademais, a função “sendEsp(String baglink)”, exemplificada acima, é responsável por enviar os dados ao servidor. Ela recebe um parâmetro do tipo String, chamado “*baglink*”, que contém os dados a serem enviados. O código cria uma URL final concatenando a URL base do servidor com a string recebida. Em seguida, cria-se um objeto “*WiFiClient*” para gerenciar a conexão com o servidor e um objeto “*HTTPClient*” para enviar a requisição HTTP. A partir disso, a função “*http.begin()*” inicia a conexão com a URL construída. Caso a conexão seja bem-sucedida, a requisição “*GET*” é enviada para o servidor e o código HTTP de resposta é verificado. Se a resposta for positiva (código HTTP 200 ou 301), o conteúdo da resposta (*payload*) é impresso na porta serial. Caso contrário, se ocorrer um erro na requisição, uma mensagem de falha é exibida, informando o tipo de erro. Após a operação, a função *http.end()* encerra a comunicação HTTP.

Diante disso, por um lado, no bloco de código “*setup()*” (Figura 3.53), são realizadas as configurações iniciais. A comunicação serial é iniciada a 115200 (números de bits enviados por segundo), e em seguida, o modo Wi-Fi do ESP8266 é configurado para o modo estação (WIFI\_STA), o que significa que o dispositivo se conectará a uma rede Wi-Fi existente, enquanto a função “*WiFiMulti.addAP(STASSID, STAPSK)*” adiciona a rede Wi-Fi à qual o dispositivo tentará se conectar. Caso o ESP8266 já tenha uma rede configurada previamente, ele tentará se conectar automaticamente à rede configurada. Se não houver uma conexão, o código continuará tentando se conectar até encontrar uma rede válida.

Figura 3.53 – Void setup.

```
void setup() {  
    USE_SERIAL.begin(115200);  
    WiFi.mode(WIFI_STA);  
    WiFiMulti.addAP(STASSID, STAPSK);  
}
```

Fonte: Elaborada pelos autores.

Por outro lado, na função “loop()” (Figura 3.54), o dispositivo realiza verificações e operações continuamente. Dentro do “loop()”, o código verifica se o ESP8266 está conectado à rede Wi-Fi e se o dispositivo estiver conectado, ele aguarda dados na porta serial. Quando dados estão disponíveis na porta serial, o código lê os caracteres e os armazena na variável “phrase”, que é construída com os caracteres lidos. Após coletar todos os dados disponíveis, a função “sendEsp(phrase)” é chamada, enviando a *string* coletada como parâmetro para o servidor. O código então continua executando esse processo de captura e envio de dados, garantindo que os dados da porta serial sejam enviados continuamente para a plataforma web.

Figura 3.54 – Void loop.

```
void loop() {  
    if ((WiFiMulti.run() == WL_CONNECTED)) {  
        String phrase;  
        if (USE_SERIAL.available() > 0){  
            delay(5);  
            while (USE_SERIAL.available() > 0) { //  
                char caractere; //  
                caractere = USE_SERIAL.read (); //  
                int enter = caractere;  
                phrase = phrase + caractere;  
            }  
            sendEsp(phrase);  
        }  
    }  
}
```

Fonte: Elaborada pelos autores.

Este fluxo de trabalho permite que o dispositivo, ao ser conectado a uma rede Wi-Fi, funcione como um "ponteiro" entre a comunicação serial e o servidor remoto. Dados podem ser enviados para o servidor sem a necessidade de interação direta com um computador, tornando o sistema autônomo e eficiente para aplicações em que o ESP8266 precisa coletar e enviar dados sem a presença constante de um dispositivo controlador.

### 3.8 Site

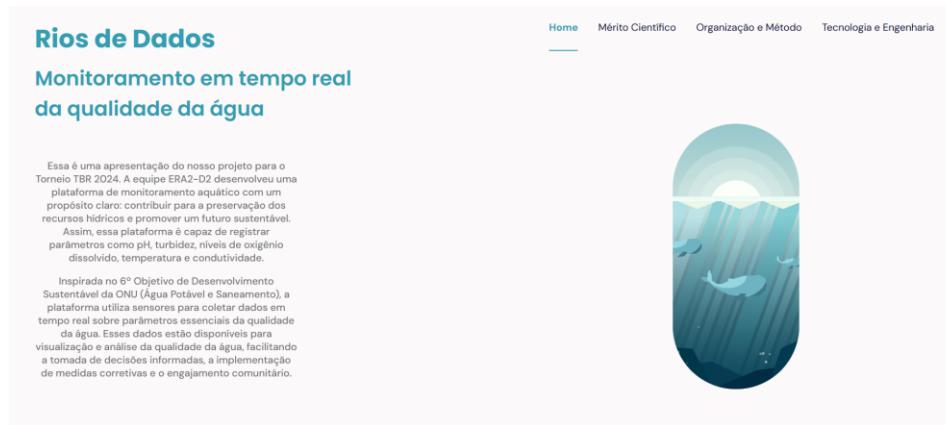
Desde a fase regional do TBR, na qual a equipe ERA2-D2 foi campeã, o desenvolvimento do site planejado passou por transformações significativas. Inicialmente, a proposta da plataforma web (Figura 3.55) ainda não apresentava uma ideia concretizada de design, estrutura e funcionalidade. Com a transição para a etapa nacional, as ideias foram reformuladas, resultando na versão atual da plataforma (Figura 3.56). Essa reformulação foi motivada pela necessidade de profissionalizar o projeto, tornando-o mais moderno, sério e atrativo, sem perder a essência original da equipe.

Figura 3.55 – Página inicial do site na etapa regional do TBR.



Fonte: Elaborada pelos autores.

Figura 3.56 – Página inicial do site na etapa nacional do TBR.



Fonte: Elaborada pelos autores.

Nesse cenário, para explicar o desenvolvimento de tal plataforma, é importante contextualizar algumas outras mudanças significativas ocorridas no intervalo entre a etapa regional e a fase nacional do TBR, que impactaram diretamente a evolução do site. Há

cerca de três meses, o sistema do site, mantido pela MakeSoftSolutions — patrocinadora responsável pela criação e manutenção da plataforma —, utilizava *PHP* puro. Recentemente, entretanto, foi realizada a migração para o *framework Laravel*, que segue o padrão arquitetônico *MVC* (*Model-View-Controller*), trazendo maior organização e eficiência ao projeto.

Diante disso, tal arquitetura separa as responsabilidades de manipulação de dados, interface do usuário e controle de lógica de negócios. Os *Models* são responsáveis pelo gerenciamento do acesso e manipulação de dados no banco, permitindo operações como inserções, consultas e atualizações. Já as *Views* representam a interface visual, interagindo diretamente com o usuário, enquanto os *Controllers* intermediários processam os dados dos *Models* e os exibem nas *Views*. Esse sistema em camadas facilita o desenvolvimento e a manutenção da aplicação.

### a) Desenvolvimento da programação

O mapeamento de URLs ocorre no arquivo *web.php*, onde são definidas as rotas para os *Controllers* e suas funções associadas. Por exemplo, ao acessar *makesoftsolutions.net/era2d2*, a requisição é direcionada ao *EraController*, que executa a função *index*. Esta função processa os dados necessários e retorna uma *View* para exibição ao usuário. A linguagem *Blade*, integrada ao *Laravel*, facilita a reutilização de componentes como o cabeçalho (*header*) em várias páginas, garantindo uma aparência visual consistente e minimizando a duplicação de código.

Ademais, uma funcionalidade crucial é o link “*makesoft.net.br/era2d2/store*”, configurado para receber dados transmitidos diretamente por dispositivos como o *Arduino*. Na função *store*, implementada no *EraController*, são recebidos valores sequenciais de pH, turbidez, temperatura e condutividade, que são armazenados no banco de dados através de *Models* específicos, como *phCreate* e *turbidezCreate*. Cada valor é inserido em sua tabela correspondente, promovendo organização e eficiência no gerenciamento dos dados.

Além disso, para a exibição de documentos PDF, foi utilizado o visualizador do Google, mas para lidar com falhas ocasionais de carregamento, foi necessário implementar um código em *JavaScript* que verifica se o PDF foi carregado corretamente e, em caso de erro, realiza um recarregamento automático para garantir uma experiência mais estável ao usuário. Em conjunto da exibição, a biblioteca “*Apex Charts*” foi adotada para criar gráficos interativos. Configurando adequadamente os dados e implementando

um script JavaScript, os gráficos são gerados automaticamente, facilitando a visualização clara e atrativa das informações.

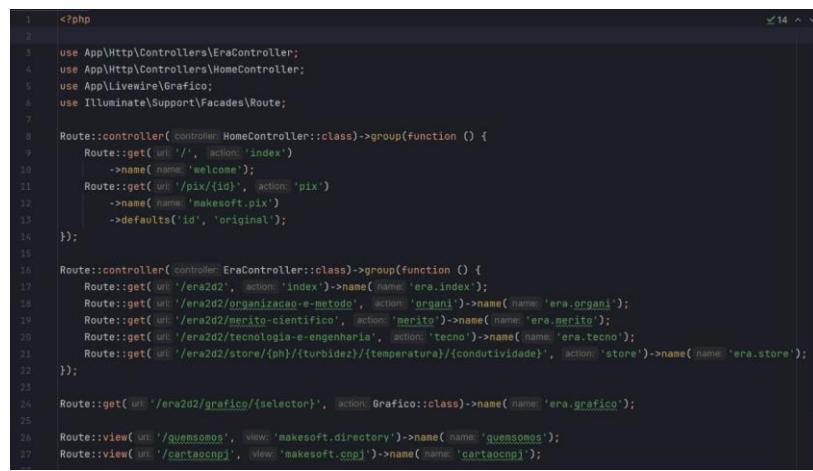
A partir disso, a estruturação do banco de dados foi feita através do uso das Migrations do Laravel. Cada tabela foi definida em arquivos específicos, detalhando suas colunas e propriedades. O comando “*php artisan migrate*” permite a criação automática dessas tabelas no banco de dados. Os *Models* foram vinculados a essas tabelas seguindo uma convenção de nomenclatura padrão, com ajustes específicos para garantir o correto mapeamento quando necessário.

Por motivos de segurança, o Laravel organiza os arquivos de forma hierárquica. Recursos públicos, como arquivos CSS, imagens e ícones, são armazenados no diretório “*Public*”, enquanto as *Views* e outras partes críticas do sistema ficam protegidas em diretórios internos. Isso evita o acesso não autorizado a componentes sensíveis.

### b) Código explicado

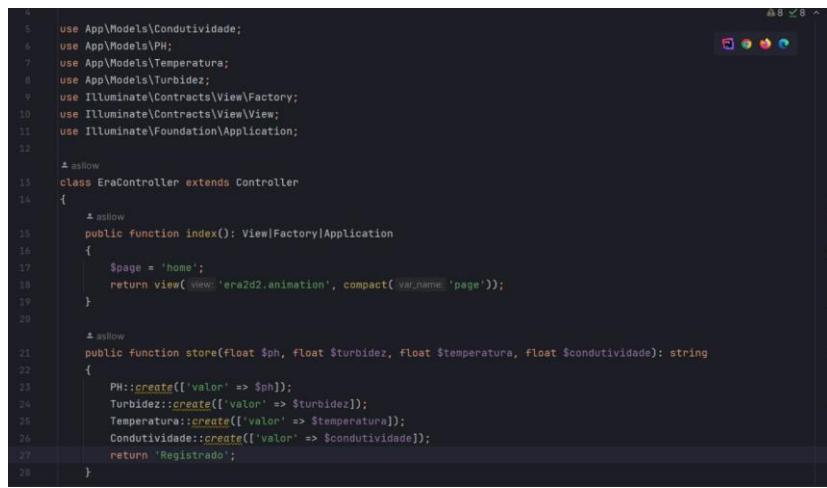
A função *index* declara uma variável chamada *\$page* com o valor ‘home’, que é compactada e passada para a View *era2d2.animation*. Esta View, por sua vez, exibe a interface inicial do site, incluindo a animação dos peixes na página principal. O arquivo *era2d2.animation* utiliza a diretiva @extends para herdar elementos estruturais do arquivo pai *era2d2.index*, que também herda o layout básico do arquivo principal *era2d2.header*.

Figura 3.57 – Código que define as rotas e os controllers.



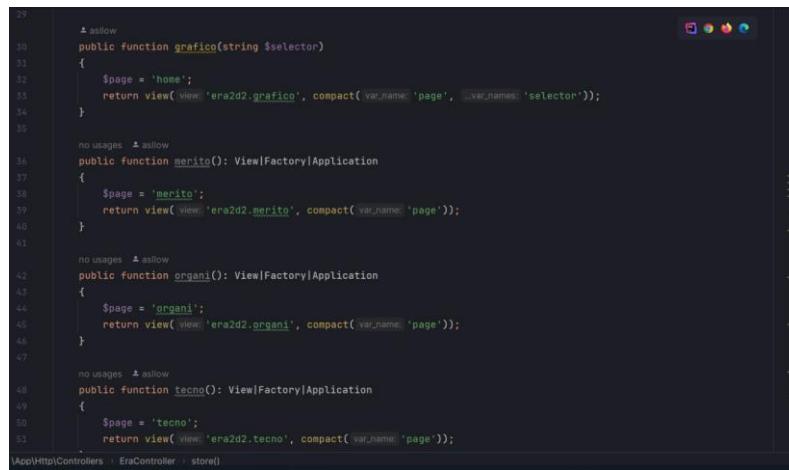
```
1 <?php
2
3 use App\Http\Controllers\EraController;
4 use App\Http\Controllers\HomeController;
5 use App\Livewire\Grafico;
6 use Illuminate\Support\Facades\Route;
7
8 Route::controller(HomeController::class)->group(function () {
9     Route::get('/', action: 'index')
10        ->name('welcome');
11     Route::get('/pix/{id}', action: 'pix')
12        ->name('makesoft.pix')
13        ->defaults('id', 'original');
14 });
15
16 Route::controller(EraController::class)->group(function () {
17     Route::get('/era2d2', action: 'index')->name('era.index');
18     Route::get('/era2d2/organizacao-e-metodo', action: 'organiz')->name('era.organiz');
19     Route::get('/era2d2/merito-cientifico', action: 'merito')->name('era.merito');
20     Route::get('/era2d2/tecnologia-e-engenharia', action: 'tecnico')->name('era.tecnico');
21     Route::get('/era2d2/store/{ph}/{turbidez}/{condutividade}', action: 'store')->name('era.store');
22 });
23
24 Route::get('/era2d2/grafico/selector', action: Grafico::class)->name('era.grafico');
25
26 Route::view('/quemsomos', view: 'makesoft.directory')->name('quemsomos');
27 Route::view('/cartaoCNPJ', view: 'makesoft.cnpj')->name('cartaoCNPJ');
```

Fonte: Elaborada pelos autores.



```
4
5     use App\Models\Condutovidade;
6     use App\Models\PH;
7     use App\Models\Temperatura;
8     use App\Models\Turbidez;
9     use Illuminate\Contracts\View\Factory;
10    use Illuminate\Contracts\View\View;
11    use Illuminate\Foundation\Application;
12
13    ▲ asallow
14    class EraController extends Controller
15    {
16        ▲ asallow
17        public function index(): View|Factory|Application
18        {
19            $page = 'home';
20            return view( view: 'era2d2.animation', compact( var_name: 'page' ) );
21
22        ▲ asallow
23        public function store( float $ph, float $turbidez, float $temperatura, float $condutividade ): string
24        {
25            PH::create([ 'valor' => $ph ]);
26            Turbidez::create([ 'valor' => $turbidez ]);
27            Temperatura::create([ 'valor' => $temperatura ]);
28            Condutovidade::create([ 'valor' => $condutividade ]);
29            return 'Registrado';
30        }
31    }
```

Fonte: Elaborada pelos autores.



```
29
30    ▲ asallow
31    public function grafico( string $selector )
32    {
33        $page = 'home';
34        return view( view: 'era2d2.grafico', compact( var_name: 'page', ...var_names: 'selector' ) );
35    }
36
37    no usages ▲ asallow
38    public function merito(): View|Factory|Application
39    {
40        $page = 'merito';
41        return view( view: 'era2d2.merito', compact( var_name: 'page' ) );
42    }
43
44    no usages ▲ asallow
45    public function organi(): View|Factory|Application
46    {
47        $page = 'organi';
48        return view( view: 'era2d2.organi', compact( var_name: 'page' ) );
49    }
50
51    no usages ▲ asallow
52    public function tecno(): View|Factory|Application
53    {
54        $page = 'tecno';
55        return view( view: 'era2d2.tecno', compact( var_name: 'page' ) );
56    }
57
```

Fonte: Elaborada pelos autores.



```
bubbleY.style.left = ` ${Math.random() * 156}px`;
bubbleY.style.animationDelay = `${Math.random() * 3}s`;
bubbleY.appendChild(bubbleX);

// Adiciona bolha no contêiner
bubbleContainer.appendChild(bubbleY);

// Adiciona animação dinâmica para bubbleX
const keyframes =
  @keyframes bubbleX${i} {
  0% { transform: translateX(0); }
  20% { transform: translateX(-${Math.random() * 15}px); }
  40% { transform: translateX(${Math.random() * 15}px); }
  60% { transform: translateX(-${Math.random() * 15}px); }
  80% { transform: translateX(${Math.random() * 15}px); }
  100% { transform: translateX(-${Math.random() * 15}px); }
}
addKeyframes(keyframes);
}

// Gera 10 bolhas
createBubbles(10);
</script>
@endsession
```

Fonte: Elaborada pelos autores.

```

@extends('era2d2.header')

@section('title', 'ERA2-D2')

@section('description', 'Esse é o site da equipe de robótica ERA2D2, destinada ao Torneio Brasil de Robótica - TBR, categoria Robótica Submarina. O projeto visa monitorar a qualidade da água em tempo real para promover a preservação dos recursos hídricos e contribuir para um futuro sustentável.')

@section('main')
    <main id="main">
        <section class="section" id="home">
            <div class="container page1">
                <h1 class="title">Rios de Dados</h1>
                <h2 class="title subtitle">Monitoramento em tempo real da qualidade da água</h2>
                <div class="project-presentation">
                    <div class="text">
                        <p>Essa é uma apresentação do nosso projeto para o Torneio TBR 2024. A equipe ERA2-D2 desenvolveu uma plataforma de monitoramento aquático com um propósito claro: contribuir para a preservação dos recursos hídricos e promover um futuro sustentável. Assim, essa plataforma é capaz de registrar parâmetros como pH, turbidez, níveis de oxigênio dissolvido, temperatura e condutividade.</p>
                        <p>Inspirada no 6º Objetivo de Desenvolvimento Sustentável da ONU (Água Potável e Saneamento), a plataforma utiliza sensores para coletar dados em tempo real sobre parâmetros essenciais da qualidade da água. Esses dados estão disponíveis para visualização e análise da qualidade da água, facilitando a tomada de decisões informadas, a implementação de medidas corretivas e o engajamento comunitário.</p>
                    </div>
                    @yield('animation')
                </div>
            </div>
        </section>
    </main>

```

Fonte: Elaborada pelos autores.

O era2d2.header é o componente geral do site, contendo as configurações completas de HTML, CSS e JavaScript, aplicados a todas as páginas. Inclui diretrizes como @yield('title'), que define o título da página, e @yield('main'), onde o conteúdo principal de cada seção é inserido. Esse uso hierárquico de Views permite a integração de pedaços de código de arquivos filhos em arquivos pais. No caso da animação dos peixes, o conteúdo HTML, CSS e JavaScript necessário é encapsulado entre as diretivas @section('animation') e @endsection, sendo "despejado" no local indicado pelo @yield('animation') no arquivo pai era2d2.index.

Figura 3.57 – Código que define os gráficos e banco de dados.

```

1  <!DOCTYPE html>
2
3  <html>
4      <head>
5          {-- Configuration --}
6          <meta charset="UTF-8">
7          <meta http-equiv="X-UA-Compatible" content="IE=edge">
8          <meta name="viewport" content="width=device-width, initial-scale=1.0">
9          <meta name="description" content="@yield('description')">
10         <title>@yield('title')</title>
11         @if($page == 'home')
12             <link href="{{asset('public/css/animation.css')}}" rel="stylesheet" type="text/css"/>
13         @endif
14         <link href="{{asset('public/css/app.css')}}" rel="stylesheet" type="text/css"/>
15         <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.min.css">
16         @yield('stylesheet', '')
17         <script src="https://cdn.jsdelivr.net/gh/alpinejs@v2.8.2/dist/alpine.min.js"></script>
18         <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
19         <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
20         {-- Favicon --}
21         <link rel="apple-touch-icon" sizes="180x180" href="{{asset('public/favicon/apple-touch-icon.png')}}">
22         <link rel="icon" type="image/png" sizes="32x32" href="{{asset('public/favicon/favicon-32x32.png')}}">
23         <link rel="icon" type="image/png" sizes="16x16" href="{{asset('public/favicon/favicon-16x16.png')}}">
24         <link rel="manifest" href="{{asset('public/favicon/site.webmanifest')}}">
25         {-- Fonts --}
26         <link rel="preconnect" href="https://fonts.googleapis.com">
27         <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
28         <link href="https://fonts.googleapis.com/css2?family=DM+Sans:ital,wght@0..400,1000;1..400,1000&family=Pop"

```

Fonte: Elaborada pelos autores.

```

4   <html>
5     </head>
6   <body>
7     <header id="header">
8       <nav class="container">
9         <div class="menu">
10           <ul class="grid">
11             <li><a class='title @if($page=='home') active @endif' id='index' href='{{ route('era.index') }}'>Home</a>
12             </li>
13             <li><a class='title @if($page=='merito') active @endif' id='merito' href='{{ route('era.merito') }}'>Mérito
14               Científico</a></li>
15             <li><a class='title @if($page=='organico') active @endif' id='organico' href='{{ route('era.organico') }}'>Organiza
16               e Método</a>
17             </li>
18             <li><a class='title @if($page=='tecnico') active @endif' id='tecnico' href='{{ route('era.tecnico') }}'>Tecnologia
19               e Engenharia</a>
20           </ul>
21           <div class="background"></div>
22         </div>
23       </nav>
24     </header>
25     @yield('main')
26     <script type="text/javascript" src="{{ asset("public/js/app.js")}}"></script>
27   </body>
28 </html>

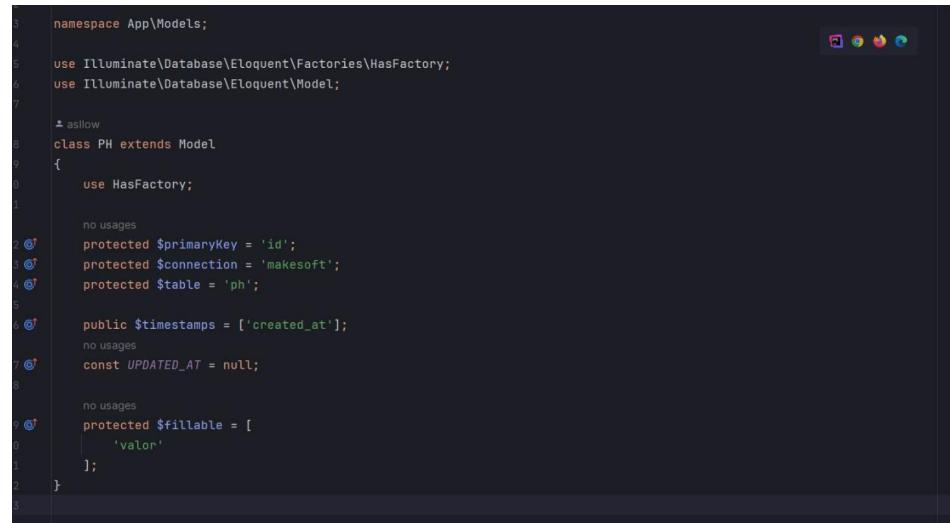
```

Fonte: Elaborada pelos autores.

Além da organização das Views, outra funcionalidade central do site está na recepção e processamento de dados. Por meio da rota makesoft.net.br/era2d2/store/{ph}/{temperatura}/{turbidez}/{condutividade}, dispositivos como o Arduino enviam valores de sensores diretamente para o site. A função store, localizada no EraController, captura esses dados e os atribui a variáveis específicas para cada parâmetro. Por exemplo, o comando PH::create() chama o Model PH, acessando a tabela correspondente no banco de dados e inserindo o valor recebido na coluna valor. Esse processo se repete para os outros parâmetros como turbidez, temperatura e condutividade.

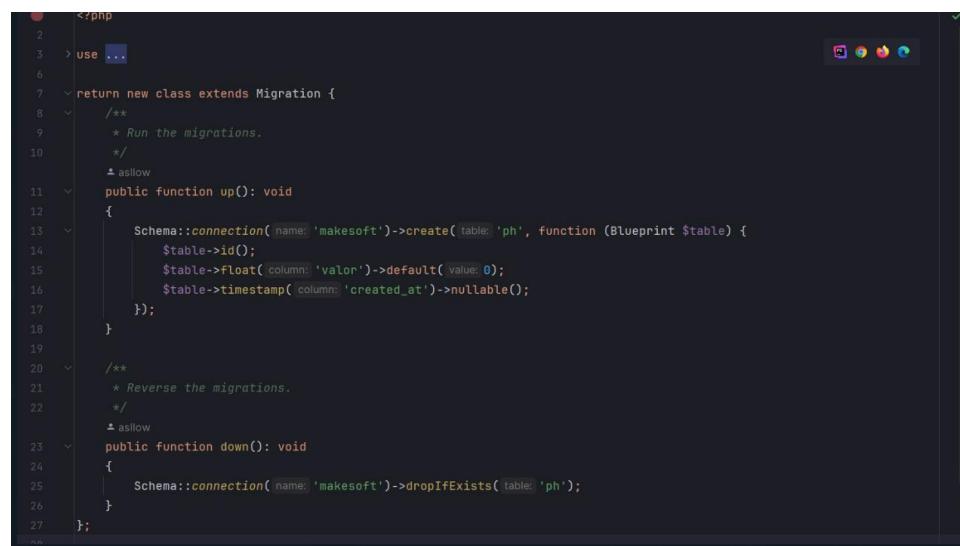
Os Models, que representam as tabelas do banco de dados, foram configurados previamente com Migrations. Essas Migrations definem a estrutura das tabelas e permitem a criação ou exclusão delas conforme necessário. A função up em uma Migration cria a tabela com colunas como id, valor e created\_at, enquanto a função down remove a tabela caso ela exista, garantindo flexibilidade e organização na gestão do banco de dados. Por exemplo, a tabela ph foi configurada com as colunas id (chave primária), valor (float) e created\_at (timestamp), otimizadas para armazenar os dados dos sensores.

Figura 3.5 – Código do migration.



```
1 namespace App\Models;
2
3 use Illuminate\Database\Eloquent\Factories\HasFactory;
4 use Illuminate\Database\Eloquent\Model;
5
6 /**
7 * @allow
8 */
9 class PH extends Model
10 {
11     use HasFactory;
12
13     no usages
14     protected $primaryKey = 'id';
15     protected $connection = 'makesoft';
16     protected $table = 'ph';
17
18     public $timestamps = ['created_at'];
19     no usages
20     const UPDATED_AT = null;
21
22     no usages
23     protected $fillable = [
24         'valor'
25     ];
26 }
27
```

Fonte: Elaborada pelos autores.



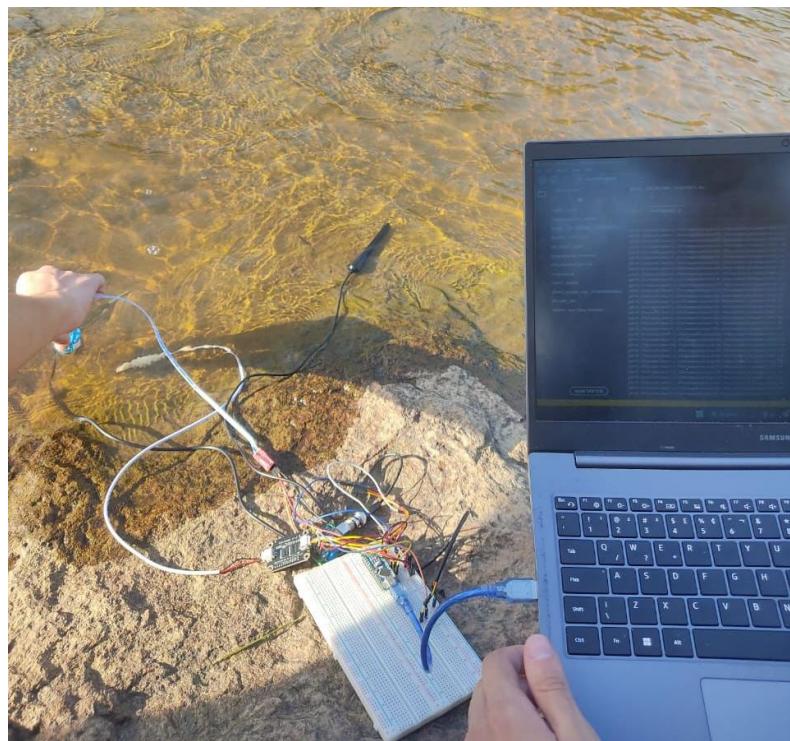
```
1 ?php
2
3 > use ...;
4
5
6 return new class extends Migration {
7     /**
8      * Run the migrations.
9      */
10    /**
11     * @allow
12     */
13    public function up(): void
14    {
15        Schema::connection('name: "makesoft")->create('table: 'ph', function (Blueprint $table) {
16            $table->id();
17            $table->float('valor')->default(0);
18            $table->timestamp('created_at')->nullable();
19        });
20    }
21
22    /**
23     * Reverse the migrations.
24     */
25    /**
26     * @allow
27     */
28    public function down(): void
29    {
30        Schema::connection('name: "makesoft')->dropIfExists('table: 'ph');
31    }
32};
```

Fonte: Elaborada pelos autores.

## 4. RESULTADOS

Buscando validar o protótipo criado pela equipe e iniciar o monitoramento em diferentes locais, a equipe dirigiu-se ao Rio Itapecerica, um afluente da margem esquerda do Rio Pará e subafluente do Rio São Francisco. O rio tem sua origem na confluência dos ribeirões Vermelho, do Gama e Santo Antônio, no município de Itapecerica, e percorre cerca de 123 quilômetros até desaguar no Rio Pará, no município de Divinópolis, onde os estudos são realizados. A plataforma foi testada em três pontos equidistantes ao longo do rio (Figuras x, x e x), o que revelou resultados diferentes em cada ponto, mas permitiu à equipe chegar a uma conclusão geral sobre a qualidade da água do rio.

Figura 4.1 – Entrada do rio na cidade. Medido às 16:55.



Fonte: Elaborada pelos autores

Figura 4.2 – Após encontro com rio Pará. Medido às 18:23.



Fonte: Elaborada pelos autores.

Figura 4.3 – Após encontro com rio Pará. Medido às 9:43.



Fonte: Elaborada pelos autores.

A análise dos dados coletados pela plataforma de monitoramento hídrico revelou informações cruciais sobre a qualidade e a dinâmica dos corpos d'água monitorados. Abaixo estão os principais resultados obtidos durante o período de

monitoramento no primeiro ponto:

Figura 4.4 – Resultados no primeiro ponto.

```
ph=8.92&turbidez=1.29&temperatura=24.94&condutividade=1657.27  
ph=8.89&turbidez=1.34&temperatura=24.88&condutividade=1680.66  
ph=8.84&turbidez=1.99&temperatura=24.94&condutividade=1657.27  
ph=8.86&turbidez=1.99&temperatura=24.88&condutividade=1649.25  
ph=8.84&turbidez=1.99&temperatura=24.94&condutividade=1644.81  
ph=8.83&turbidez=1.16&temperatura=24.88&condutividade=1649.25  
ph=8.84&turbidez=1.05&temperatura=24.94&condutividade=1595.80  
ph=8.80&turbidez=1.28&temperatura=24.94&condutividade=1583.75  
ph=8.81&turbidez=1.09&temperatura=24.88&condutividade=1564.02
```

Fonte: Elaborada pelos autores.

As medições revelaram valores de pH consistentemente próximos de 8,80, indicando um ambiente levemente alcalino. Embora esteja dentro do limite aceitável para a maioria dos usos da água, valores elevados de pH podem sugerir a presença de compostos químicos alcalinos, frequentemente associados a efluentes industriais ou agrícolas. Os valores de turbidez variaram entre 1,09 e 2,00 NTU (Unidade de Turbidez Nephelométrica). Em geral, os níveis de turbidez são considerados elevados, o que pode estar associados à erosão do solo, sedimentos ou atividades humanas locais. Já a temperatura se manteve estável, em torno de 24,88 °C a 24,94 °C. Esse padrão é típico de regiões tropicais e tem influência direta nos processos biológicos e químicos do ecossistema aquático, como a solubilidade de gases e o metabolismo dos organismos aquáticos. Os valores de TDS variaram entre 1499,81 ppm e 1680,66 ppm. Níveis elevados como esses indicam altas concentrações de íons dissolvidos, possivelmente resultantes de descargas de efluentes domésticos ou industriais, ou do escoamento de fertilizantes agrícolas

No segundo ponto, foram obtidos os seguintes parâmetros:

Figura 4.5 – Resultados no segundo ponto.

```
ph=4.88&turbidez=0.70&temperatura=26.25&condutividade=43.60  
ph=4.89&turbidez=0.70&temperatura=26.25&condutividade=43.60  
ph=4.92&turbidez=0.70&temperatura=26.31&condutividade=43.55  
ph=4.94&turbidez=0.70&temperatura=26.25&condutividade=43.60  
ph=4.94&turbidez=0.70&temperatura=26.31&condutividade=43.55  
ph=4.93&turbidez=0.70&temperatura=26.31&condutividade=43.55  
ph=4.93&turbidez=0.70&temperatura=26.31&condutividade=43.55  
ph=4.91&turbidez=0.70&temperatura=26.25&condutividade=43.60
```

Fonte: Elaborada pelos autores.

Os valores de pH variaram entre 4,37 e 4,96 durante o período de medição. Isso indica um ambiente ácido, uma vez que valores abaixo de 7 são considerados ácidos. Já a turbidez apresentou valores relativamente constantes, entre 0,68 e 0,70

NTU, indicando baixa presença de partículas suspensas na água. A temperatura permaneceu estável, com variação em torno de 26,25°C a 26,31°C, o que é típico para corpos d'água em ambientes tropicais. E por fim, o total de sólidos dissolvidos oscilou significativamente, variando de 43,00 a 43,60 ppm, o que sugere diferenças na presença de íons dissolvidos durante a medição. Nesse cenário, os dados avaliados neste ponto apresentam diferentes valores devido ao ambiente aquático contaminado pelos poluentes advindos do esgoto urbano. Isso se comprova nos valores de turvidez, que variam entre valores menores devido ao horário no qual foram realizadas as medições. Além disso, é válido destacar, que o aumento de temperatura evidente neste local também é proveniente da rede de drenagem contaminada pelo esgoto. Ademais, os valores de TDS não atenderam as expectativas esperadas pela equipe, em virtude de problemas no qual o grupo ainda não identificou. Já no terceiro ponto de coleta foram registrados os seguintes resultados:

Figura 4.6 – Resultados no terceiro ponto.

```
ph=9.51&turbidez=1.91&temperatura=25.94&condutividade=1336.61
ph=9.67&turbidez=1.91&temperatura=25.94&condutividade=1336.61
ph=9.57&turbidez=1.92&temperatura=25.94&condutividade=1336.61
ph=9.67&turbidez=1.92&temperatura=25.94&condutividade=1341.68
ph=9.66&turbidez=1.92&temperatura=25.94&condutividade=1341.68
ph=9.65&turbidez=1.92&temperatura=25.94&condutividade=1341.68
ph=9.39&turbidez=1.93&temperatura=25.94&condutividade=1341.68
ph=9.70&turbidez=1.91&temperatura=25.94&condutividade=1346.77
ph=9.60&turbidez=1.93&temperatura=25.94&condutividade=1346.77
ph=9.58&turbidez=1.93&temperatura=25.94&condutividade=1346.77
ph=9.46&turbidez=1.94&temperatura=25.94&condutividade=1346.77
ph=9.42&turbidez=1.93&temperatura=25.94&condutividade=1346.77
ph=9.40&turbidez=1.93&temperatura=25.94&condutividade=1346.77
ph=9.24&turbidez=1.93&temperatura=25.94&condutividade=1346.77
ph=9.42&turbidez=1.94&temperatura=25.94&condutividade=1346.77
ph=9.35&turbidez=1.94&temperatura=25.94&condutividade=1346.77
ph=9.42&turbidez=1.94&temperatura=25.94&condutividade=1346.77
ph=9.41&turbidez=1.94&temperatura=25.94&condutividade=1341.68
ph=9.53&turbidez=1.95&temperatura=25.94&condutividade=1341.68
ph=9.17&turbidez=1.95&temperatura=25.94&condutividade=1341.68
ph=9.03&turbidez=1.96&temperatura=25.94&condutividade=1341.68
ph=8.95&turbidez=1.96&temperatura=25.94&condutividade=1341.68
ph=8.71&turbidez=1.97&temperatura=25.94&condutividade=1336.61
```

Fonte: Elaborada pelos autores.

Ao analisar as informações, foi possível chegar nas seguintes conclusões sobre cada parâmetro. Os valores de pH variaram entre 8,00 e 9,70, com uma média de 8,9, indicando que a água está em um intervalo alcalino. Valores de pH acima de 7,0 indicam condições básicas, e os valores observados sugerem possíveis influências de processos naturais ou atividades antrópicas, como descargas de resíduos industriais ou agrícolas. A turvidez oscilou entre 1,91 e 1,97 NTU, o que indica uma água relativamente limpa, com baixa quantidade de partículas em suspensão. Este parâmetro é um indicativo da presença de sólidos

suspensos, como sedimentos ou matéria orgânica, que podem impactar a penetração de luz e a qualidade do habitat aquático. A temperatura manteve-se constante em 25,94°C durante o período de coleta. Esse valor é típico para águas superficiais em regiões tropicais, influenciando diretamente a solubilidade de gases, como oxigênio, e os processos biológicos de organismos aquáticos. E avariou entre 1328,32 ppm e 1346,77 ppm. Valores elevados de TDS indicam a presença de sais dissolvidos, que podem ser resultado de descargas industriais, agrícolas ou efluentes urbanos.

#### **4.1 Discussão**

As medições realizadas no Rio Itapecerica evidenciaram diferenças significativas nos parâmetros de qualidade da água em comparação com dados anteriores. A acidez observada na medição atual pode ser atribuída à decomposição de matéria orgânica ou ao lançamento de efluentes ácidos, enquanto os níveis alcalinos registrados anteriormente indicam a possível influência de compostos básicos provenientes de descargas industriais ou agrícolas.

A turbidez manteve-se baixa em ambas as medições, indicando uma água relativamente limpa. A variação observada entre os períodos pode estar relacionada a maior movimentação de sedimentos ou presença de partículas orgânicas, influenciadas por chuvas, atividades humanas ou diferenças nos pontos de coleta.

A temperatura permaneceu estável e dentro da faixa esperada para corpos d'água tropicais, influenciando diretamente processos biológicos e a solubilidade de gases, como o oxigênio. Pequenas variações entre as medições podem ser explicadas por condições climáticas ou diferenças nos horários de coleta.

Os valores obtidos a partir do sensor de condutividade (TDS Meter V1.0) elétrica apresentou a maior discrepância. Enquanto os valores comparativos oscilaram entre 1328,32 ppm e 1346,77 ppm, refletindo altas concentrações de sais dissolvidos frequentemente associados a descargas industriais, agrícolas ou urbanas, a medição atual apresentou ausência de leituras consistentes, sugerindo interferências ou condições incomuns de baixa mineralização.

Os resultados refletem características típicas de um corpo hídrico em área

urbana, com significativa influência antrópica. O pH alcalino registrado anteriormente pode ser associado à entrada de detergentes, efluentes industriais ou fertilizantes, enquanto a baixa turbidez sugere ausência de aumento significativo de sedimentos no momento da coleta, dependente do regime de chuvas e da erosão local. Os altos valores de TDS reforça a presença de contaminantes dissolvidos, como sais ou metais, que são indicadores relevantes da qualidade da água.

Por outro lado, a estabilidade térmica sugere a ausência de descargas térmicas, o que é benéfico para organismos sensíveis a variações de temperatura. Apesar disso, a acidez e as oscilações de sólidos dissolvidos na medição atual indicam possíveis comprometimentos na qualidade da água, tornando-a inadequada para usos como abastecimento direto sem tratamento prévio.

## **4.2 Implicações Ambientais**

Os resultados indicam que o Rio Itapecerica está sendo impactado por atividades humanas que ameaçam sua qualidade. A alta condutividade elétrica e o pH alcalino sugerem riscos à biodiversidade aquática e à saúde pública, especialmente se a água for utilizada para abastecimento ou recreação. Flutuações térmicas podem reduzir os níveis de oxigênio dissolvido, prejudicando espécies sensíveis, enquanto a turbidez reflete problemas de erosão e atividades agrícolas, que afetam tanto a estética quanto os processos de tratamento da água.

A presença de efluentes industriais contribui para a poluição química, com metais pesados e nutrientes que comprometem a saúde aquática e a potabilidade. A poluição térmica agrava a situação ao reduzir o oxigênio disponível, afetando a fauna local. Além disso, mudanças climáticas, como o aumento das temperaturas e chuvas intensas, exacerbam a carga de sedimentos e poluentes, tornando necessárias ações integradas para mitigar esses efeitos.

A interdependência entre os parâmetros analisados, como turbidez, consumo de oxigênio e hipoxia, reforça a importância de uma gestão sustentável dos recursos hídricos. A implementação de monitoramento contínuo, políticas públicas eficazes e ações de conscientização são essenciais para mitigar impactos

e preservar a qualidade da água.

Para garantir a qualidade do Rio Itapecerica, é crucial fortalecer o monitoramento, tratar os efluentes industriais e implementar medidas de controle de sedimentos e poluentes, assegurando a preservação dos ecossistemas e a saúde pública.

### **4.3 Embasamento Científico**

A partir de pesquisas já realizadas foi possível comprovar a eficácia da plataforma, assim como sua importância para a preservação da qualidade da água.

A expedição "Esse rio é meu", organizada pelo Comitê de Bacia Hidrográfica do Rio Pará e acompanhada pela equipe do *Estado de Minas* no ano de 2023, confirmou a presença de esgoto sanitário no rio, evidenciando a necessidade de intervenções imediatas.

Segundo ambientalistas, a análise de amostras revelou que a oxigenação do Rio Itapecerica está abaixo do recomendado pela resolução do Conselho Nacional do Meio Ambiente (Conama) nº 357, a qual determina que o índice de um rio de água doce não pode ser inferior ao índice 6. No caso do Rio Itapecerica, o nível demonstrado é 3,9, bem inferior, portanto, ao recomendado. Além disso, a análise para a presença de coliformes totais termotolerantes deu positivo, comprometendo a qualidade da água, o consumo e a vida aquática presente nele.

Esses resultados reforçam a necessidade de ações imediatas e eficazes para reverter a degradação do rio. Segundo Adriano Parreira, professor da Universidade Estadual de Minas Gerais (UEMG), que realizou a análise das amostras, é essencial a tomada de iniciativas do Comitê de Bacia Hidrográfica, como a expedição, ao proporcionar maior visibilidade à situação para a população. “Iniciativas como essa incentivam os cidadãos a exigirem do poder público medidas concretas para a melhoria da qualidade da água”.

Esse cenário evidencia que a integração entre monitoramento, conscientização pública e políticas públicas é essencial para reverter os impactos ambientais e garantir a sustentabilidade do Rio Itapecerica.

Assim, diante da presença de esgoto e da baixa oxigenação no Rio Itapecerica, conforme evidenciado pelas análises recentes, a plataforma projetada

pela equipe se torna uma ferramenta indispensável para fornecer dados precisos e consistentes, fundamentais para a formulação de tais políticas públicas. O uso contínuo da plataforma permite uma gestão integrada dos recursos hídricos, além de facilitar a implementação de ações corretivas e preventivas, garantindo a sustentabilidade e a qualidade da água para as futuras gerações.

## 5. RECURSOS UTILIZADOS

A seguir é apresentada uma tabela com os gastos que foram realizados para a construção do protótipo da plataforma de monitoramento do ar atmosférico.

Tabela 4.1 - Custos para o desenvolvimento do protótipo desenvolvido pela equipe ERA2D2

Descrição	Quantidade	Preço Unitário (R\$)	Preço Final (R\$)
KIT Módulo Sensor de Temperatura	1	13,23	13,23
Sensor de Condutividade de Água	1	17,19	17,19
Sensor de Turbidez	1	28,41	28,41
Fretes	-	59,80	59,80
Sensor pH + Eletrodo Sonda Bnc	1	87,10	87,10
Arduino mega 2560	1	0	0
Placa ESP32-WROOM	3	0	0
Resistores 4.7kΩ	1	0	0
Fios de ligação	-	0	0
Placa de cobre	2	0	0
<b>TOTAL (R\$)</b>			<b>205,73</b>

Fonte: Elaborada pelos autores.

Os resistores, fios de ligação, placa de cobre, placa ESP32-WROOM e o Arduino Mega 2560 não geraram custos, pois foram obtidos a partir de sucatas eletrônicas, e de doações feitas ao projeto.

A plataforma de monitoramento também pretende incorporar um sensor de oxigênio dissolvido, que ainda não foi adquirido devido ao seu alto custo (R\$284,37). A aquisição de um sensor nesse valor comprometeria um dos principais diferenciais da plataforma, que é ser uma solução de monitoramento hídrico de baixo custo. Porém, devido a importância desse parâmetro, o sensor não pode ser desprezado pois se configura como um dado importante para o projeto. Assim, alternativas mais acessíveis estão sendo estudadas para integrar ao protótipo. Caso não seja possível, a compra será uma futura implementação.

## 6. CRONOGRAMA

Quadro 5.1 - Cronograma de FEVEREIRO à MAIO de 2024.

AÇÕES PREVISTAS [EQUIPE ERA2D2 2024]	RESPONSÁVEL	PRAZO TOTAL (DIAS)	RESULTADOS ESPERADOS	MESES			
				FEV	MAR	ABR	MAI
Reunião de finalização da Temporada 2023	Equipe	1	Despedida de antigos membros e patrocinadores.				
Processo Seletivo de Novos Membros	Equipe(Técnico e mentor)	5	Seleção de novos membros comprometidos com o projeto.				
Reunião de Integração com Novos Membros	Equipe	1	Apresentação da equipe, do projeto, das regras e das funções de cada membro.				
Publicação de Vídeo de Agradecimento	Equipe	1	Agradecimento aos patrocinadores e colaboradores				
Abertura Oficial da temporada 2024	Equipe	1	Temporada 2024 iniciada				
Elaboração do orçamento da temporada	Equipe (Gestores)	10	Orçamento detalhado e realista para todo o projeto.				
Documentação para patrocínio	Equipe (Gestores)	20	Documentação pronta para a solicitação de patrocínio				
Definição do Design do Robô	Equipe (Robô)	20	Design funcional, considerando as regras da competição.				
Escolha do projeto para mérito científico	Equipe (Mérito científico)	20	Escolha do projeto com melhor custo-benefício				
Construção do robô	Equipe (Robô)	15	Robô construído e pronto para ser testado				
Início da documentação do TBR	Equipe (Gestores)	1	Documentação da temporada iniciada				
Início das programações do projeto científico	Equipe (Mérito científico)	1	Programações pronta para serem testadas				
Início das programações do robô	Equipe (Robô)	1	Programações pronta para serem testadas				

Fonte: Elaborado pelos autores.

Quadro 5.2 - Cronograma de JUNHO à SETEMBRO de 2024.

AÇÕES PREVISTAS [EQUIPE ERA2D2 2024]	RESPONSÁVEL	PRAZO TOTAL (DIAS)	RESULTADOS ESPERADOS	MESES			
				JUN	JUL	AGO	SET
Término da construção do robô	Equipe (Robô)	1	Robô finalizadas para testes	X			
Inicio do planejamento de abordagens para o desafio prático	Equipe (Robô)	30	Planejamento iniciado	X			
Inicio vendas da rifa	Equipe	1	Venda de rifas iniciados		X		
Término das estratégias de abordagens no desafio prático	Equipe (Robô)	1	Estratégias finalizadas		X		
Inicio dos testes do robô no tapete	Equipe (Robô)	1	Iniciados os testes do robô no tapete		X		
Inicio da construção do projeto científico	Equipe (Mérito científico)	1	Iniciada a construção do projeto científico		X		
Sorteio da rifa	Equipe	1	Rifa sorteada			X	
Término dos testes do robô	Equipe (Robô)	1	Teste concluidos			X	
Término da construção do projeto científico	Equipe (Mérito científico)	1	Projeto científico construído			X	
Teste do projeto científico	Equipe (Mérito científico)	20	Projeto científico testado e aprovado			X	
Término da documentação	Equipe (Gestores)	1	Documentação concluída			X	
Finalização e conferência de todo o progresso da equipe	Equipe	5	Progresso finalizado				X
Ensaio para apresentação	Equipe	10	Ensaio completo				X
Entrega de documentos com 7 dias de antecedência	Equipe (Gestores)	1	Documentos entregues				X
Participação da etapa regional do TBR 2024	Equipe	1	Participação da equipe				X

Fonte: Elaborado pelos autores.

## **7. CONSIDERAÇÕES FINAIS**

Este estudo destaca a relevância da plataforma desenvolvida pela equipe ERA2-D2, que está diretamente alinhada ao 6º Objetivo de Desenvolvimento Sustentável (ODS 6) da ONU, que visa a preservação dos recursos hídricos. A plataforma permite o monitoramento contínuo de parâmetros essenciais como pH, turbidez, oxigênio dissolvido, temperatura e condutividade da água, proporcionando detecção precoce de poluentes e anomalias. Isso o torna uma ferramenta valiosa para controlar a qualidade da água e promover a conscientização e preservação ambiental.

Sua implementação oferece uma solução acessível e prática, viável para adoção por comunidades e instituições de diversos portes. Além de fornecer dados fundamentais para a gestão dos recursos hídricos, a plataforma tem potencial para fortalecer políticas públicas e estratégias de proteção ambiental. Ao disponibilizar esta informação de forma acessível, sensibiliza também a população para a importância da preservação da água potável, recurso vital para o bem-estar social, econômico e ambiental.

O projeto ERA2-D2 demonstra como a ciência pode ser aplicada para resolver problemas ambientais prementes. Beneficia as gerações atuais e futuras, garantindo uma gestão mais eficaz dos recursos hídricos.

Por último, este trabalho destaca a importância das parcerias entre os setores público, privado e acadêmico, que são essenciais para enfrentar os desafios globais relacionados com a água. Só através da cooperação entre estas esferas será possível promover um futuro mais sustentável e seguro para todos.

## 8. REFERÊNCIAS BIBLIOGRÁFICAS

- AGÊNCIA NACIONAL DE ÁGUAS (ANA).** Conjuntura dos Recursos Hídricos no Brasil. Brasília: ANA, 2018.
- ALI, S.; KHUSHIK, A. G.; MEMON, A. H.** Water Resources Management: A Focus on Global Water Demand.
- ANTZOULATOS, G.; SKORDILIS, E.; MANATAKI, I.** Enhancing water management systems with IoT: A case study for urban water distribution. *Journal of Environmental Management*, 2020.
- ASSOCIAÇÃO BRASILEIRA DE ENGENHARIA SANITÁRIA E AMBIENTAL (ABES).** Perdas no sistema de distribuição de água no Brasil. Brasília: ABES, 2021.
- BORGES, P. R.; SOUZA, R. F.; ZANARDI, M. C.** Modelagem matemática para gestão de perdas em redes de abastecimento. 2017.
- BRITO, Luiza T. de L.; SRINIVASAN, Vajapeyam S.; SILVA, Aderaldo de S.; GHEYI, Hans R.; GALVÃO, Carlos de O.; HERMES, Luiz C.** Influência das atividades antrópicas na qualidade das águas da bacia hidrográfica do Rio Salitre. *Revista Brasileira de Engenharia Agrícola e Ambiental*, v.9, n.4, p.596-602, Campina Grande, PB, 2005.
- BUREK, P.; SATOH, Y.; FISCHER, G.; KAHIL, T.; SCHERZER, A.; TRAMBEREND, S.; WADA, Y.** Water Futures and Solutions: World Water Scenarios. IIASA, Laxenburg, Austria, 2016.
- CASSIDY, R.; DU, J.; MITCHELL, S.** Real-time water quality monitoring in distribution networks. 2021.
- DI MAURO, C.; BAROUDI, J.; HESS, B.** Water management and sustainability: Addressing growing global demand. 2021.
- FLORENCIO, A.; MALPASS, J.** Environmental laws in Brazil: A legislative review. 2014.
- FUNDAÇÃO GETÚLIO VARGAS (FGV).** Estudo sobre a distribuição da demanda de água no Brasil. Rio de Janeiro: FGV, 1998.
- GUERRA, A. J. T.; MENDONÇA, J. J.** Impactos ambientais urbanos: a questão das enchentes e inundações. 2010.
- HANAI, Frederico Yuri.** Caracterização qualitativa e quantitativa de esgotos sanitários. Universidade de São Paulo, Escola de Engenharia de São Carlos, Departamento de Hidráulica e Saneamento, São Paulo, 1997.

**INSTITUTO ÁGUA E SANEAMENTO.** Brasil retrocede em metas de água e saneamento da ONU, revela VII Relatório Luz 2023. Publicado em 03 out. 2023.

**INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA (IBGE).** Censo Demográfico 2011. Rio de Janeiro: IBGE, 2011.

**INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA (IBGE).** Censo Demográfico 2022. Rio de Janeiro: IBGE, 2022.

**INSTITUTO FEDERAL - SANTA CATARINA.** Escassez de água: entenda por que esse problema também é seu. Ana Paula Lückman - Jornalista do IFSC, 27 mar. 2023.

**JUNG, D.; LANSEY, K.** A simplified approach for water distribution system management. 2015.

**NASCIMENTO, Flávio Rodrigues do.** Categorização de usos múltiplos dos recursos hídricos e problemas ambientais. Universidade Federal Fluminense - Programa de Pós-Graduação em Geografia. *Revista da ANPEGE*, v. 7, n. 1, número especial, p. 81-97, out. 2011.

**NARAYANAN, A.; SANKARANARAYANAN, B.** Water Quality Monitoring in Smart Cities: Challenges and Opportunities. Nome do Periódico, 2020.

**OLIVEIRA, A. M.; HERRMANN, M. L.** A questão ambiental e a urbanização. Local de publicação: Editora, 2006.

**ORGANIZAÇÃO DAS NAÇÕES UNIDAS PARA AGRICULTURA E ALIMENTAÇÃO (FAO).** Relatório sobre a gestão dos recursos hídricos globais. Roma: FAO, 2020.

**ORGANIZAÇÃO MUNDIAL DA SAÚDE (OMS).** Environmental health risks: The global burden of disease. OMS, 2007.

**SANTIN, Janaína Rigo; GOELLNER, Emanuelle.** A Gestão dos Recursos Hídricos e a Cobrança pelo seu Uso. Universidade de Passo Fundo, Passo Fundo — RS, Brasil. *Revista Scielo*, 2013.

**SÉRIE DE VÍDEO-AULAS** para a turma de Saneamento Ambiental dos cursos de Engenharia Ambiental e Engenharia Civil da UFRN. Junho, 2020.

**SILVA, Valdir Martiniano Ferreira da.** Efeito das ações antrópicas na qualidade da água da bacia do Riacho do Silva, em Maceió-AL. Universidade Federal de Alagoas, Centro de Tecnologia, Programa de Pós-Graduação em Recursos Hídricos e Saneamento, Alagoas, 2011.

**SODRÉ, Fernando Fabriz.** Fontes Difusas de Poluição da Água: Características e métodos de controle. Grupo de Automação, Quimiometria e Química Ambiental (AQQUA), Instituto de Química, Universidade de Brasília, 2012.

**SISTEMA NACIONAL DE INFORMAÇÕES SOBRE SANEAMENTO (SNIS).** Diagnóstico dos serviços de água e esgoto — 2021. Brasília: Ministério do Desenvolvimento Regional, 2021.

**TUCCI, C. E. M.** Gerenciamento integrado de águas urbanas. Porto Alegre: Editora da UFRGS, 2002.

**TUCCI, C. E. M.** Hidrologia Urbana. Porto Alegre: Editora da UFRGS, 2003.

## 9. APÊNDICE

APÊNDICE A — Código referente a programação dos sensores utilizados na plataforma de monitoramento aquático.

// Bibliotecas.

```
#include <Wire.h>          // Biblioteca auxiliar para o protocolo de comunicação UART.
```

```
#include <OneWire.h>        // Biblioteca auxiliar para o sensor de temperatura.
```

```
#include <DallasTemperature.h> // Biblioteca auxiliar para o sensor de temperatura.
```

// Declaração dos pinos.

```
#define pHPin A0            // pH.
```

```
#define turbidezPin A1       // Turbidez.
```

```
#define temperaturaPin A2    // Sinal do sensor DS18B20 (Temperatura).
```

```
#define condutividadePin A3   // Condutividade.
```

```
//-----OBJETOS-----  
-----//
```

// Temperatura.

```
OneWire oneWire(temperaturaPin);
```

```
DallasTemperature sensors(&oneWire);
```

```
DeviceAddress sensor1;
```

```
//-----CONSTANTES-----  
-----//
```

```
#define SCOUNT 30 // Tamanho do vetor de soma.  
  
const double leituraAnalogica = ((double)5.0 / 1024.0); // 0,488 volts por valor.
```

```
//-----VARIÁVEIS-----  
-----//
```

```
// Variáveis Auxiliares.
```

```
bool flag = 1, corrigecodut = 1;
```

```
unsigned long tempo = 0;
```

```
// Variáveis dos sensores.
```

```
double pH = 0;
```

```
double NTU = 0;
```

```
float TEMP = 0;
```

```
int analogBuffer[SCOUNT]; // Armazena os valores analógicos lidos do sensor.
```

```
int analogBufferTemp[SCOUNT]; // Cópia temporária do buffer para processamento.
```

```
int analogBufferIndex = 0; // Índice do buffer.
```

```
int copyIndex = 0;
```

```
double tensaoEstavel = 0;
```

```
double TDS = 0;
```

//-----ALGORÍTIMOS-----

-----//

// Filtragem média do sensor de condutividade.

```
int getMedianNum(int bArray[], int iFilterLen) {
```

```
    int bTab[iFilterLen];
```

```
    for (int i = 0; i < iFilterLen; i++) {
```

```
        bTab[i] = bArray[i];
```

```
}
```

```
    int i, j, bTemp;
```

```
    for (j = 0; j < iFilterLen - 1; j++) {
```

```
        for (i = 0; i < iFilterLen - j - 1; i++) {
```

```
            if (bTab[i] > bTab[i + 1]) {
```

```
                bTemp = bTab[i];
```

```
                bTab[i] = bTab[i + 1];
```

```
                bTab[i + 1] = bTemp;
```

```
}
```

```
}
```

```
}
```

```
    if ((iFilterLen & 1) > 0) bTemp = bTab[(iFilterLen - 1) / 2];
```

```
    else bTemp = (bTab[iFilterLen / 2] + bTab[iFilterLen / 2 - 1]) / 2;
```

```
return bTemp;  
}
```

```
//-----FUNÇÕES-----  
-----//
```

```
void PH() {  
  
    const float calibracao = 22.0;  
  
    unsigned long somaLeiturasValidas = 0;  
  
    unsigned int leituras[10];
```

```
// Realiza 10 leituras do sensor.
```

```
for (byte i = 0; i < 10; i++) {  
  
    leituras[i] = analogRead(pHPin);  
  
}
```

```
// Organiza os valores em ordem crescente.
```

```
for (byte i = 0; i < 9; i++) {  
  
    for (byte j = i + 1; j < 10; j++) {  
  
        if (leituras[i] > leituras[j]) {  
  
            int temp = leituras[i];  
  
            leituras[i] = leituras[j];  
  
        }
```

```

leituras[j] = temp;

}

}

for (byte i = 2; i < 8; i++) somaLeiturasValidas += leituras[i]; // Descarta as duas
primeiras e duas últimas leituras e soma.

double pHVol = somaLeiturasValidas * leituraAnalogica / 6; // Faz a média entre as 6
leituras restantes.

pH = -5.70 * pHVol + calibracao; // Retorna um valor de pH.

}

void Turbidez() {

double tensao = (analogRead(turbidezPin) * leituraAnalogica);

NTU = (double)(((-1120.4 * tensao * tensao) + (5742.3 * tensao) - 4352.9) / 1000); //
((-1120.4 * volt * volt) + (5742.3 * volt) -4352.9)/1000;

}

void Temperatura() {

sensors.requestTemperatures(); // Le a informação do sensor.

TEMP = sensors.getTempC(sensor1);

```

```
}
```

```
void Condutividade() {
```

```
    analogBuffer[analogBufferIndex] = analogRead(condutividadePin); // Lê o valor  
    analógico e armazena no buffer.
```

```
    analogBufferIndex++;
```

```
    if (analogBufferIndex == SCOUNT) analogBufferIndex = 0;
```

```
    if(flag) {
```

```
        flag = 0;
```

```
        for (copyIndex = 0; copyIndex < SCOUNT; copyIndex++) {
```

```
            analogBufferTemp[copyIndex] = analogBuffer[copyIndex];
```

```
        }
```

```
}
```

```
// Lê o valor analógico mais estável pelo algoritmo de filtragem média e converte para  
um valor de tensão.
```

```
tensaoEstavel = getMedianNum(analogBufferTemp, SCOUNT) * leituraAnalogica;
```

```
// Compensação do valor de condutividade em função da temperatura.
```

```

float compensationCoefficient = 1.0 + 0.02 * (TEMP - 25.0); // ResultadoFinal(25°C)
= ResultadoFinal(atual)/(1,0 + 0,02 * (temperatura - 25,0));

float compensation = tensaoEstavel / compensationCoefficient;

// Converte o valor da tensão em um valor de condutividade. O valor calculado é
multiplicado por 0.5 para converter para TDS em ppm.

TDS = (133.42 * compensation * compensation * compensation - 255.86 *
compensation * compensation + 857.39 * compensation) * 0.5;

}

void Conexao(double pH, double turbidez, float temperatura, double condutividade) {

// Envia as informações lidas para o ESP8266 01.

String mensagem = "pH = " + String(pH) + "&turbidez = " + String(turbidez) +
"&temperatura = " + String(temperatura) + "&condutividade = " + String(condutividade);

Serial.print(mensagem);

Serial.println();

}

void TestePrint(){

Serial.println("O pH é: " + String(pH));

```

```
Serial.println("A turbidez é: " + String(NTU));

Serial.println("A temperatura é: " + String(TEMP));

Serial.println("O total de Sólidos Dissolvidos é: " + String(TDS, 0) + "ppm");

Serial.println();

}
```

```
//-----SETUP-----//
-----//
```

```
void setup() {
```

```
// Inicializa o Monitor Serial para a conexão com o ESP8266 01.
```

```
Serial.begin(115200);
```

```
// Define os pinos como entrada.
```

```
pinMode(pHPin, INPUT); // pH.
```

```
pinMode(turbidezPin, INPUT); //Turbidez.
```

```
pinMode(temperaturaPin, INPUT); //Temperatura.
```

```
pinMode(condutividadePin, INPUT); //Condutividade.
```

```
// Inicializações do sensor de temperatura.
```

```
sensors.begin(); //Temperatura.
```

```
sensors.getAddress(sensor1, 0); //Temperatura.
```

```

}

//-----LOOP-----
//-----//


void loop() {

    // Lê os dados continuamente e envia a cada 1 segundo.

    tempo = millis();

    while (millis() - tempo < 1000) {

        PH();

        Turbidez();

        Temperatura();

        Condutividade();

    }

    Conexao(pH, NTU, TEMP, TDS);

    flag = 1;

}

```

APÊNDICE B – Código referente a programação do microcontrolador ESP32 para transmissão dos dados para a plataforma WEB do protótipo.

```

#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>

#define USE_SERIAL Serial
#define STASSID "LabMaker" //LabMaker
#define STAPSK "53471409000152" //53471409000152

```

```

#define SERVERNAME "http://makesoft.net.br/era2d2/store/";
ESP8266WiFiMulti WiFiMulti;

void sendEsp(String baglink){
    WiFiClient client;
    HTTPClient http;

    String finalLink = SERVERNAME + baglink;
    if (http.begin(client, finalLink)) { // HTTP
        int httpCode = http.GET();

        if (httpCode > 0) {
            USE_SERIAL.printf("[HTTP] GET... código: %d\n", httpCode);

            if (httpCode == HTTP_CODE_OK || httpCode == HTTP_CODE_MOVED_PERMANENTLY) {
                String payload = http.getString();
                USE_SERIAL.println(payload);
            }
        } else {
            USE_SERIAL.printf("[HTTP] GET... falhou, erro: %s\n",
                http.errorToString(httpCode).c_str());
        }
        http.end();
    } else {
        USE_SERIAL.println("[HTTP] Impossível conectar");
    }
}

void setup() {
    USE_SERIAL.begin(115200);
    WiFi.mode(WIFI_STA);
    WiFiMulti.addAP(STASSID, STAPSK);
}

void loop() {
    if ((WiFiMulti.run() == WL_CONNECTED)) {
        String phrase;
        if (USE_SERIAL.available() > 0){
            delay(5);
            while (USE_SERIAL.available() > 0) { //
                char caractere; //
                caractere = USE_SERIAL.read (); //
                int enter = caractere;
                phrase = phrase + caractere;
            }
            sendEsp(phrase);
        }
    }
}

```

**APÊNDICE B** – Código referente a programação do microcontrolador ESP32 para transmissão dos dados para a plataforma WEB do protótipo.