



**UNIVERSIDADE DO ESTADO DO AMAZONAS
ESCOLA SUPERIOR DE TECNOLOGIA**

MARIO FERNANDO MOURA DAS NEVES NETO

**DESENVOLVIMENTO DE PROTÓTIPO DE SISTEMA DE BAIXO
CUSTO PARA MONITORAMENTO DA QUALIDADE DA ÁGUA DE
BACIAS AMAZÔNICAS**

Manaus
2022

MARIO FERNANDO MOURA DAS NEVES NETO

**DESENVOLVIMENTO DE PROTÓTIPO DE SISTEMA DE BAIXO
CUSTO PARA MONITORAMENTO DA QUALIDADE DA ÁGUA DE
BACIAS AMAZÔNICAS**

Pesquisa desenvolvida durante a disciplina de Trabalho de conclusão de curso II e apresentada à banca avaliadora do Curso de Engenharia Eletrônica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas como pré-requisito para a obtenção do título de Engenheiro em Eletrônica.



Orientador: Fábio de Sousa Cardoso, Prof. Dr.

Manaus
2022

Universidade do Estado do Amazonas – UEA
Escola Superior de Tecnologia – EST

Reitor:

Cleinaldo de Almeida Costa, Dr.

Vice-Reitor:

Cleto Cavalcante de Souza Leal, Me.

Diretora da Escola Superior de Tecnologia:

Ingrid Sammyne Gadelha Figueiredo, Me.

Coordenador do Curso de Engenharia Eletrônica:

Bruno da Gama Monteiro MSc.

Banca Avaliadora composta por:

Data da defesa: 21/01/2022.

Prof. Fábio de Sousa Cardoso, Dr. (Orientador)

Prof. Cláudio Gonçalves, Dr. (Avaliador 1)

Prof. Bruno da Gama Monteiro MSc. (Avaliador 2)

CIP – Catalogação na Publicação

Neto, Mario Fernando

Desenvolvimento de protótipo de sistema de baixo custo para monitoramento da qualidade da água de bacias amazônicas / Mario Fernando Moura das Neves Neto; [orientado por] Prof. Fábio de Sousa Cardoso – Manaus: 2022.

68 f. p.: il.

Trabalho de Conclusão de Curso (Graduação em Engenharia Eletrônica). Universidade do Estado do Amazonas, 2022.

1. Medição de parâmetros da água, 2. Qualidade da água
3. ESP32 LoRa, 4. Monitoramento em tempo real, 5. Interface
Web.

I. Sousa Cardoso, Fábio.

MARIO FERNANDO MOURA DAS NEVES NETO

**DESENVOLVIMENTO DE PROTÓTIPO DE SISTEMA DE BAIXO
CUSTO PARA MONITORAMENTO DA QUALIDADE DA ÁGUA
DE BACIAS AMAZÔNICAS**

Pesquisa desenvolvida durante a disciplina de Trabalho de Conclusão de Curso II e apresentada à banca avaliadora do Curso de Engenharia Eletrônica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para a obtenção do título de Engenheiro em Eletrônica.

Nota obtida: 9,5 (Nove pontos e cinco décimos)

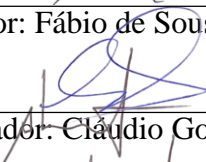
Aprovada em: 21/01/2022.

Área de concentração: Sistemas embarcados, Internet das Coisas.

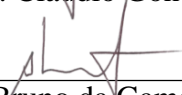
BANCA EXAMINADORA



Orientador: Fábio de Sousa Cardoso, Dr.



Avaliador: Cláudio Gonçalves, Dr.



Avaliador: Bruno da Gama Monteiro MSc.

Manaus
2022

DEDICATÓRIA

Dedico este trabalho primeiramente a Deus, pois sem ele nada disso seria possível, a minha avó que não está mais presente entre nós, porém tenho certeza que está muito orgulhosa, aos meus pais, por todo o suporte durante toda a trajetória na universidade, a minha noiva, por todo o apoio e a todos os meus familiares e amigos que direta ou indiretamente me apoiaram nesta caminhada. Meu muito obrigado.

AGRADECIMENTOS

A Deus por me manter saudável e firme em meus objetivos. À minha família que sempre me apoiou e confiou em minhas escolhas. A minha noiva e amigos, por compartilharem seus conhecimentos quando mais precisei. Ao meu orientador pela paciência, pelo compartilhamento de suas experiências, ensinamentos, bons direcionamentos e incentivo. Ao professor da disciplina de TCC pelas instruções de formatação do trabalho escrito.

RESUMO

Este documento apresenta o projeto de desenvolvimento de um protótipo de sistema de baixo custo para monitoramento da qualidade da água de bacias amazônicas. O estudo científico tem por objetivo a busca de soluções que possam monitorar a qualidade da água dos rios, especificamente da região amazônica, uma vez que não há sistemas de monitoramento similares a esse, principalmente capazes de receberem informações dos parâmetros da água em tempo real. São apresentados, de forma estruturada, todos os componentes do sistema, os princípios teóricos e práticos do seu funcionamento, os objetivos e ganhos decorrentes do projeto. Primeiramente, mostra-se a fundamentação teórica das tecnologias utilizadas, apresentando os sensores que serão utilizados para cada parâmetro determinado (temperatura, pH, TDS e turbidez), baseado na plataforma ESP32 LoRa para aquisição, processamento e envio dos dados, primeiramente via protocolo de comunicação LoRaWan e em seguida, via rede de comunicação Wi-Fi ou via redes de dados móveis para uma central de monitoramento. Estes componentes estarão confinados em caixa de acrílico desenvolvida para comportar o circuito eletrônico, na qual a mesma flutuará sobre as águas do rio devido a utilização de macarrão de piscina. As informações pertinentes serão exibidas em uma plataforma IoT (Arduino IoT Cloud), que é um *software* próprio para desenvolvimento IoT. Em seguida, em métodos e materiais, são apresentadas as etapas de desenvolvimento do projeto, os materiais, equipamentos e *softwares* utilizados. Posteriormente, em implementação do projeto, são detalhadas as metodologias utilizadas para estudo e como os componentes devem interagir para a coleta dos dados necessários ao estudo científico. Também serão informadas as condições e limitações do modelo teórico e suas diferenças em relação as condições reais do ambiente natural. Na sequência, em resultados obtidos, são mostrados e analisados os dados do projeto em seu pleno funcionamento. Finalmente, conclui-se por meio dos dados obtidos que a solução proposta, aplicadas as melhorias e alterações necessárias ao seu funcionamento no ambiente natural, pode ser testada em condições reais de forma que possa ser aproveitada em outras regiões do mundo.

Palavras-chaves: Qualidade da água, sensores, ESP32 LoRa, Monitoramento em tempo real, Interface Web.

ABSTRACT

This document presents the project for the development of a low-cost system prototype for monitoring the water quality of Amazon Hydrographics basins. The scientific study aims to find solutions that can monitor the river water quality, specifically in the Amazon Region, since there is no monitoring system similar to this one, mainly capable of receiving information of water parameters in real time. All the components of the system, the theoretical and practical principles of its operation, the objectives and gains resulting from the project are presented in a structured way. First of all, the theoretical foundation of the technologies used is shown, presenting the sensors that will be used for each determined parameter (temperature, pH, TDS and turbidity), based on the ESP32 LoRa platform for data acquisition, processing and sending, firstly via LoRaWan communication protocol and then via Wi-Fi communication network or via mobile data networks to a monitoring center. These components will be confined in an acrylic box developed to hold the electronic circuit, in which it will float on the river water due to the use of pool noodles. The pertinent information will be displayed on an IoT platform (Arduino IoT Cloud), which is exclusive software for IoT development. Then, in methods and materials, the stages of project development, materials, equipment and software used are presented. Later, in project implementation, the methodologies used for the study are detailed and how the components must interact to collect the data necessary for the scientific study. The conditions and limitations of the theoretical model and their differences in relation to the real conditions of the natural environment will also be informed. Afterwards, in the results obtained, the data of the project in its full functioning are shown and analyzed. Finally, it is concluded from the data obtained that the proposed solution, applied the improvements and changes necessary for its functioning in the natural environment, can be tested in real conditions so that it can be used in others regions of the world.

Keywords: Water Quality, Sensors, ESP32 LoRa, Real Time Monitoring, Web Interface.

LISTA DE FIGURAS

Figura 1 – Sensor de Temperatura DS18B20	20
Figura 2 - Sensor de pH PH4502C.....	20
Figura 3 – Esquema Elétrico Sensor de pH PH4502C.....	21
Figura 4 – Sensor de Turbidez SEN0189.....	21
Figura 5 – Gráfico de Resposta do Sensor SEN0189	22
Figura 6 – Sensor TDS.....	22
Figura 7- Circuito Eletrônico do Módulo TDS	23
Figura 8 - Comparação entre ESP32, ESP8266 e Arduino UNO.	24
Figura 9 - Diagrama de Pinos ESP 32 LoRa.....	24
Figura 10 - Estrutura de uma rede LoRaWAN.....	25
Figura 11 - Diagrama Funcional do Protótipo.	28
Figura 12 - Pagina Web Arduino.	29
Figura 13 - Página Web VS Code	30
Figura 14 - Tela de Instalação	31
Figura 15 - Tela Inicial do Fritzing.....	32
Figura 16 - Bibliotecas no IDE Arduino	32
Figura 17 - Circuito interno de alimentação do ESP 32.....	33
Figura 18 - Pinagem GPIO do módulo ESP 32.....	34
Figura 19 - Esquemático das conexões – Sensores e Módulo ESP 32.....	35
Figura 20 - Fluxograma do dispositivo Slave.	36
Figura 21 - Fluxograma do dispositivo Master.	36
Figura 22 - Tela de configuração principal (Arduino IoT Cloud).....	37
Figura 23 - Tela de importação de código (Arduino IoT Cloud).....	38
Figura 24 - Protótipo Mecânico.	38
Figura 25 - Protótipo Completo	41
Figura 26 - Página do Dashboard.....	44
Figura 27 - Medidor Multiparâmetro HI98195.....	45
Figura 28 - Medidor de Turvação HI98703-02.....	45
Figura 29 - Validação das medições do sensor de pH PH4502C.....	46
Figura 30 - Validação das medições do sensor de temperatura DS18B20.....	47
Figura 31 - Solução Padrão TDS.....	48

Figura 32 - Validação das Medições do Sensor TDS.....48

SUMÁRIO

INTRODUÇÃO	10
1 REFERENCIAL TEÓRICO.....	15
1.1 IMPORTÂNCIA DA ÁGUA.....	15
1.2 QUALIDADE DA ÁGUA.....	15
1.3 PARÂMETROS DE QUALIDADE DA ÁGUA.....	16
1.3.1 Temperatura	17
1.3.2 Potencial Hidrogeniônico (pH)	17
1.3.3 Turbidez	18
1.3.4 Sólidos Totais Dissolvidos	18
1.4 SENSORES.....	19
1.4.1 Sensor DS18B20	19
1.4.2 Sensor PH-4502C.....	20
1.4.3 Sensor SEN0189	21
1.4.4 Sensor TDS SEN0244.....	22
1.5 MICROCONTROLADOR: MÓDULO ESP 32 LORA	23
1.6 LoRaWAN E LPWAN	25
1.7 PROTOCOLO DE COMUNICAÇÃO WIFI.....	26
2 MÉTODOS E MATERIAIS.....	27
3 IMPLEMENTAÇÃO DO PROJETO	29
3.1 INSTALAÇÃO DOS <i>SOFTWARES</i> UTILIZADOS	29
3.1.1 Arduino IDE.....	29
3.1.2 Visual Studio Code	30
3.1.3 Fritzing	30
3.2 INSTALAÇÃO DAS BIBLIOTECAS NO IDE ARDUINO.....	31
3.3 IMPLEMENTAÇÃO DO DISPOSITIVO DE AQUISIÇÃO DE DADOS E COMUNICAÇÃO.....	33
3.4 FIRMWARES DOS DISPOSITIVOS DE AQUISIÇÃO DE DADOS E COMUNICAÇÃO.....	35
3.5 CONFIGURAÇÃO DA PLATAFORMA ARDUINO IOT CLOUD	36
3.6 DESENVOLVIMENTO DO PROTÓTIPO MECÂNICO	38
3.7 MONTAGEM DO PROTÓTIPO.....	39
3.7.1 Idealização do Modelo	39
3.7.2 Seleção de Componentes e Sistemas.....	39
3.7.3 Aquisição de Materiais.....	39
3.7.4 Montagem e Integração.....	40
3.7.5 Comissionamento.....	40
4 RESULTADOS OBTIDOS	43
4.1 RESULTADO DO SISTEMA DE MONITORAMENTO	43
4.2 INTERFACE GRÁFICA	43
4.3 TESTES E CALIBRAÇÃO DOS SENSORES	44
CONCLUSÃO	50
REFERÊNCIAS	52

APÊNDICE A - CÓDIGO USADO NO MICROCONTROLADOR MASTER PARA RECEBER A LEITURA DOS SENSORES E ENVIAR PARA A PLATAFORMA WEB.....56

APÊNDICE B - CÓDIGO USADO NO MICROCONTROLADOR SLAVE PARA REALIZAR A LEITURA DOS SENSORES DE TEMPERATURA, PH, TDS E TURBIDEZ..... 60

INTRODUÇÃO

Este trabalho tem como tema o desenvolvimento de protótipo de sistema de baixo custo para monitoramento da qualidade da água de bacias amazônicas. As águas dos rios influenciam os sistemas naturais e atividades humanas no contexto de uma bacia hidrográfica. Por outro lado, as águas dos rios também são afetadas pelas atividades humanas, por mudanças ambientais e por alterações climáticas. A Amazônia, a maior floresta tropical do mundo com uma das maiores biodiversidades do planeta, está em perigo com um assustador aumento no desmatamento. A interferência danosa do homem nos últimos anos se intensificou com o avanço da atividade agropecuária, construções de hidrelétricas, garimpos clandestinos, roubo de madeiras, invasão de terras indígenas e de unidades de conservação.

No período de agosto de 2018 a julho de 2019, o desmatamento da Amazônia cresceu 30%, o equivalente a 1,4 milhões de campos de futebol, segundo dados do Projeto de Monitoramento do Desmatamento na Amazônia Legal por Satélite (PRODES), medido pelo Instituto Nacional de Pesquisas Espaciais (INPE, 2019). O desmatamento da Amazônia em abril de 2020 foi o maior dos últimos dez anos, com 529 km² da floresta derrubada, conforme os dados do Sistema de Alerta de Desmatamento (SAD), do Instituto do Homem e Meio Ambiente da Amazônia (Imazon). No ano de 2011, o Fórum Econômico Mundial identificou que questões de água, de energia e de alimentos possuem forte correlação, formando um sistema, que se negligenciado gera um risco global sério.

Outra atividade ilegal bastante desenvolvida na Amazônia é o garimpo. Recentemente foi publicada uma matéria na qual onde pode-se ter o conhecimento sobre o quão grande é a contaminação que pode ser gerada por essa atividade, devido a utilização de mercúrio, afetando não somente a comunidade local, mas também outras regiões do país e até mesmo outros países. Uma tecnologia barata, mas com um custo socioambiental altíssimo. Essa é a definição do uso do mercúrio na mineração do Brasil. Para os garimpeiros, é a forma mais rápida para separar o ouro do cascalho. Para a sociedade, representa uma série de danos que geram efeitos em cascata (LOBATO, 2021).

O mercúrio, que serve de ímã para grudar os pedaços menores de ouro, tornando-os mais visíveis e fáceis de serem separados, ainda é usado em larga escala pelos garimpeiros ilegais. Esta atividade causa impacto em todo o ecossistema e a cadeia biológica do entorno: contamina as águas e degrada as condições de saúde de comunidades ribeirinhas, cidades próximas e se estende para além da região (LOBATO, 2021).

As dragas funcionam da seguinte forma: as mangueiras são jogadas no fundo do rio e, por meio de bombas, a água é sugada, passando por uma esteira. É nessa etapa que o mercúrio é usado, separando o ouro das pedras e lama. A água contaminada é devolvida ao rio novamente. A inalação do produto já é um risco, podendo causar danos nos pulmões. Mas quando despejado num curso d'água, onde encontra um ambiente sem oxigênio, ele se transforma no metilmercúrio, substância muito perigosa para o organismo humano (LOBATO, 2021).

Os peixes carnívoros têm a maior probabilidade de serem os que vão acumular mais metilmercúrio no organismo, pois se alimentam de outros peixes pequenos que podem estar contaminados, como é o caso da piracatinga. Muitos pescadores utilizam da carne do boto ou do jacaré como iscas para capturar essa espécie de pescado que, por serem animais grandes, podem ter uma elevada contaminação de metilmercúrio. Mesmo não sendo tão consumidos na região, há um mercado em potencial para a piracatinga, principalmente, fora do país. A piracatinga é um peixe de couro que tem pouco consumo na Amazônia, mas encontra amplo mercado em outras partes do Brasil e em países como a Colômbia. No mercado de outras regiões, ele é conhecido como 'douradinha'(LOBATO, 2021).

Nas pessoas, a contaminação pode causar desde dificuldades motoras, a problemas de visão e alterar outros sentidos, além de afetar o campo neurológico, comprometendo a memória. Algumas das pesquisas realizadas pela Fundação Oswaldo Cruz (FioCruz), como reportou o site Infoamazonia, mostram que em crianças esses casos tornam-se mais graves. Em um estudo na Terra Indígena Sawré Muybu, no oeste do Pará, foi constatado sérios atrasos motores e anemia grave no povo Munduruku. Moradores de comunidades ribeirinhas e indígenas têm o seu modo de vida afetado por peixes contaminados e a violência social que acompanha o garimpo (LOBATO, 2021).

O garimpo não prejudica apenas o meio ambiente, é um trabalho que não segue nenhuma regulamentação, e que precisa ser interrompido imediatamente, como afirma Carlos Durigan. “É uma atividade que tem que ser banida. Para explorar o ouro na Amazônia, temos que recomeçar do zero, construir uma nova agenda e discutir com especialistas para promover alguns estudos para ver se é viável, possível”, afirma, alertando que atualmente ainda não existe nenhum modelo de exploração de ouro na Amazônia de forma sustentável (LOBATO, 2021).

Gerenciar um aspecto deste sistema sem considerar os demais, aumenta o risco para as civilizações. Com o aumento da população, a gestão da água se tornou uma questão de extrema importância para nossa sociedade. Muitas pessoas no mundo sequer têm acesso à água potável. Todo ano milhares de pessoas sofrem de várias doenças fatais causadas pela

poluição da água. Segundo a Organização Mundial da Saúde (OMS), pelo menos 2 milhões de pessoas, principalmente crianças com menos de 5 anos de idade, morrem por ano no mundo devido a doenças causadas pela água contaminada (ASBRAN, 2011). Porém, os problemas poderiam ser evitados por meio de políticas públicas eficientes e a adoção de um plano de gestão de água potável de qualidade. De acordo com a World Health Organization aproximadamente 1.4 milhões mortes de crianças poderiam ser evitadas se tivessem acesso a água potável. A OMS dispõe de um plano denominado Planejamento de Água Saudável, que define mudanças na gestão da água potável em vários países, visando definir procedimentos de segurança para assegurar a qualidade da água usada na alimentação, orientações à população e recomendações sobre os riscos envolvidos (ASBRAN, 2011). Conforme o plano, as autoridades devem estar atentas às mudanças climáticas, que provocam alterações de temperatura da água, às ameaças de escassez do produto e o controle no uso de substâncias químicas para o armazenamento de água potável.

Segundo Fisch, Marengo e Nobre (1998), a Bacia Amazônica possui uma área estimada de 6,3 milhões de quilômetros quadrados, sendo aproximadamente 5 milhões em território brasileiro e o restante entre os países da Bolívia, Colômbia, Equador e Peru. Segundo o Instituto Brasileiro de Geografia e Estatística (IBGE), a área da Amazônia Legal no Brasil é de 5.032.925 km², compreendidos pelos estados do Pará, Amazonas, Rondônia, Roraima, Acre e Amapá e parte dos estados do Tocantins, Mato Grosso e Maranhão. A bacia amazônica envolve um conjunto de recursos hídricos, pequenas bacias hidrográficas, que convergem para o rio Amazonas, sendo a maior bacia hidrográfica do mundo (FERREIRA, 2013; NEILL et al., 2006). Ecossistemas terrestres adjacentes desempenham um papel dominante na formação da hidrologia e biogeoquímica desses pequenos rios e suas bacias hidrográficas. Para Neill et al. (2006), é em pequenas bacias hidrográficas onde as consequências das mudanças no uso do solo na Amazônia são mais agudas. É em pequenas bacias hidrográficas que as consequências das mudanças no uso do solo na Amazônia são mais agudas. A Amazônia está mudando rapidamente porque as florestas estão sendo derrubadas pela exploração madeireira e pela expansão da agricultura e das áreas urbanas.

De acordo com o Sistema Nacional de Informações sobre Saneamento (SNIS), em 2017, apenas 57,3% dos brasileiros do Norte recebiam água tratada, apenas 10,2% da população tinha acesso à rede de coleta de esgotos e 17,4% do esgoto gerado era tratado. As perdas de água potável nos sistemas de distribuição chegam aos incríveis 55,1% (TRATA BRASIL, 2019). Isso significa que mais de 7,5 milhões de pessoas vivem da água coletada diretamente de rios e igarapés da Amazônia. Giatti e Cutolo (2012), comentando os dados do IBGE no

período entre 2002 e 2009, confirmam esse sério déficit para a Amazônia Legal quanto ao percentual de domicílios permanentes ligados à rede de abastecimento de água na região, que em 2009 foi de 58,5%, bastante inferior ao nacional no mesmo ano (83,1%). O percentual de domicílios em estados amazônicos com ligação à rede de esgotos sanitários, que em 2009 foi de apenas 9,4%, enquanto o nacional figurou 52,5%.

Diante deste cenário de ameaças, e sendo a água doce um recurso natural tão importante, não só do ponto de vista ambiental como social, humano e econômico, é extremamente relevante o controle e monitoramento de sua qualidade nas bacias amazônicas. As informações disponibilizadas através de sensores devem permitir uma avaliação dinâmica consistente e fornecer alertas indispensáveis à ação das autoridades. Até mesmo um sistema simples, composto de alguns sensores, posicionados em regiões sob cobertura de rede de telefonia móvel, trará dados importantes sobre a qualidade e outros parâmetros físicos da água dos rios amazônicos.

Assim, o que se propõe neste trabalho, é desenvolver e montar um protótipo de sistema baixo custo em laboratório, localizado na Escola Superior de Tecnologia (EST) da Universidade do Estado do Amazonas (UEA), baseado na plataforma ESP32 LoRaWan para aquisição, processamento e envio de dados via rede de comunicação LoRaWan e Wi-Fi para uma central de monitoramento. Para isso, serão utilizados sensores capazes de mensurar parâmetros de qualidade da água como temperatura, pH, turbidez e sólidos totais dissolvidos.

O estudo proposto se justifica, uma vez que pode trazer uma solução eficaz, capaz de monitorar e alertar as autoridades ambientais responsáveis, para que sejam tomadas as decisões cabíveis o mais rápido possível e sejam feitas fiscalizações mais rigorosas. Além disso, o projeto explora conhecimentos adquiridos durante o curso de Engenharia Eletrônica, principalmente das disciplinas: Microprocessadores e Microcontroladores, Sensores, Instrumentação Eletrônica, Linguagem de Programação, Eletrônica Aplicada e Redes de comunicação.

Este trabalho está dividido nos seguintes capítulos: referencial teórico, métodos e materiais, implementação do projeto e resultados obtidos.

No primeiro capítulo é abordada uma revisão dos assuntos referentes aos componentes eletrônicos, *softwares* utilizados e conceitos dos protocolos de comunicação. No segundo capítulo são mostrados os métodos utilizados para se obter o objetivo final. No terceiro capítulo, implementação do projeto, que aborda a execução dos passos citados no capítulo de

métodos. Mostra-se todas as etapas de construção do sistema, assim como a criação da interface de acesso ao usuário. No quarto capítulo, resultados obtidos, mostram-se os resultados dos experimentos realizados na implementação, e faz-se análises baseadas no referencial teórico e no conhecimento adquirido durante todo o processo de pesquisa e implementação do projeto. E, por fim, é apresentada a conclusão que aborda a análise do comportamento do módulo utilizado e a discussão acerca da utilização destes dispositivos para a aplicação em monitoramento.

1 REFERENCIAL TEÓRICO

1.1 IMPORTÂNCIA DA ÁGUA

A água é um dos principais recursos naturais presentes no planeta, sendo essencial para a subsistência do homem, principalmente pelo fato de diversas atividades humanas serem dependentes deste recurso, como o abastecimento humano, o uso industrial, o uso na agricultura, a geração de energia elétrica, turismo, lazer, dessedentação animal, pesca, dentre outros. Apesar disso, como afirmam Saad et al. (2007), a água é uma das principais formas de difusão de agentes patogênicos.

A água é um bem de domínio público, um recurso limitado e que possui valor econômico sendo destinada prioritariamente ao consumo humano e à dessedentação de animais em caso de escassez (BRASIL, 1997).

1.2 QUALIDADE DA ÁGUA

Para Myers (2017), a qualidade da água é definida como uma medida das características físicas, químicas, biológicas e microbiológicas da água, e é um desafio crescente devido ao grande número de produtos químicos usados em nossa vida cotidiana, no comércio, na agricultura, no extrativismo mineral e na indústria que podem chegar até nossas águas. O monitoramento fornece evidências para apoiar tomar decisões sobre o gerenciamento da qualidade da água, sobre questões de saúde pública e ambientais. O monitoramento pode ser usado para alertar sobre problemas atuais, contínuos e emergentes, visando controlar e garantir padrões seguros de água potável. Os dados coletados e processados podem ajudar gestores e legisladores a medir a eficácia das políticas hídricas, ou seja, avaliar se a qualidade da água está melhorando ou piorando e formular novas políticas visando garantir saúde às populações e proteção ao meio ambiente.

Conforme a Agência Nacional de Águas e Saneamento Básico (ANA), a Política Nacional de Recursos Hídricos estabelece que a gestão dos recursos hídricos deve proporcionar o uso múltiplo das águas, os quais possuem diferentes requisitos de qualidade. Cita-se como exemplo que para se preservar as comunidades aquáticas é necessária uma água com certo nível de oxigênio dissolvido, temperatura, pH, nutrientes, entre outros. Já para a navegação os requisitos de qualidade da água são bem menores, devendo estar ausentes os materiais flutuantes e os materiais sedimentáveis que causem assoreamento do corpo d'água. Assim, os usos da água são condicionados pela sua qualidade, ou seja, as águas com maior qualidade

permitem a existência de usos mais exigentes. Segundo a ANA, o enquadramento de um rio, ou de qualquer corpo d'água, deve considerar três aspectos principais: o rio que temos; o rio que queremos; e o rio que podemos ter. O enquadramento é influenciado por aspectos técnicos, econômicos, sociais e políticos. O processo de enquadramento deve considerar estes aspectos para que sejam estabelecidas metas de qualidade das águas factíveis de serem alcançadas. Observa-se que a qualidade da água é função da quantidade de água disponível no rio, que varia ao longo do ano (épocas de cheias e estiagens), assim, deve-se definir ou empregar vazões de referência para se efetuar medidas. A agência enfatiza que o Brasil adota o enquadramento por classes de qualidade.

O Índice de Qualidade das Águas (IQA) foi criado em 1970, nos Estados Unidos, pela National Sanitation Foundation. A partir de 1975 começou a ser utilizado pela Companhia Ambiental do Estado de São Paulo (CETESB). Nas décadas seguintes, outros Estados brasileiros adotaram o IQA, que hoje é o principal índice de qualidade da água utilizado no país (AGÊNCIA, 2005a).

O IQA foi desenvolvido para avaliar a qualidade da água bruta visando seu uso para o abastecimento público, após tratamento. Os parâmetros utilizados no cálculo do IQA são em sua maioria indicadores de contaminação causada pelo lançamento de esgotos domésticos. O IQA é composto por nove parâmetros mais representativos para a caracterização da qualidade das águas: oxigênio dissolvido, coliformes termotolerantes, pH, demanda bioquímica de oxigênio, nitrato, fosfato total, variação da temperatura da água, turbidez e sólidos totais. (AGÊNCIA, 2005a).

1.3 PARÂMETROS DE QUALIDADE DA ÁGUA

De acordo com o Índice de Qualidade das Águas, que foi adotado por grande parte dos estados brasileiros como o principal índice de qualidade da água no país, existem nove parâmetros principais utilizados para avaliar a qualidade da água bruta visando seu uso para o abastecimento público, após tratamento. São eles: oxigênio dissolvido, coliformes termotolerantes, pH, demanda bioquímica de oxigênio, temperatura da água, nitrogênio total, fósforo total, turbidez e sólidos totais dissolvidos (AGÊNCIA, 2005b).

Para este trabalho, foram escolhidos quatro destes parâmetros físico-químicos: temperatura, turbidez, sólidos dissolvidos totais e pH.

1.3.1 Temperatura

Pinto (2007) define temperatura como a medida da intensidade de calor expresso em uma determinada escala. As escalas mais usadas para esta grandeza são grau centígrado e grau Celsius ($^{\circ}\text{C}$), podendo ser medida por diferentes dispositivos, como, por exemplo, termômetro ou sensor. A utilização de sensores para monitorar a temperatura da água em rios e também em reservatórios vem se tornando algo recorrente, visto que a temperatura da água interfere diretamente no rendimento de cultivos, afetam os organismos aquáticos e pode influenciar na ocorrência de doenças.

A alteração da temperatura da água pode ser causada por antropogênicas (despejos industriais e águas de resfriamento de máquinas) ou por fontes naturais, principalmente energia solar. De acordo com o manual de controle de qualidade da água, publicado pela Fundação Nacional de Saúde (2014), a temperatura pode influenciar na velocidade das reações químicas, nas atividades metabólicas dos organismos e na solubilidade de substâncias. Os ambientes aquáticos brasileiros apresentam, em geral, temperaturas na faixa de 20°C a 30°C . Entretanto, em períodos de inverno, nas regiões mais frias, como no sul do país, a temperatura da água pode baixar a valores entre 5°C e 15°C , atingindo, em alguns casos, até o ponto de congelamento.

Em relação às águas para consumo humano, temperaturas elevadas aumentam as perspectivas de rejeição ao uso, e pode ocasionar efeitos laxativos. Águas subterrâneas captadas a grandes profundidades frequentemente necessitam de unidades de resfriamento, a fim de adequá-las ao abastecimento (FUNDAÇÃO, 2014).

Além disso, verificou-se que há uma relação entre o aumento da temperatura da água e outros parâmetros físico-químicos, como por exemplo, o oxigênio dissolvido e a quantidade de sólidos dissolvidos e suspensos nas águas (CARVALHO; SCHLITTLER; TORNISIELO, 2000).

1.3.2 Potencial Hidrogeniônico (pH)

“O potencial hidrogeniônico representa a intensidade das condições ácidas ou alcalinas do meio líquido através da medição da presença de íons hidrogênio (H^+) expressa em mols de íons de hidrogênio por litro de solução” (INSTITUTO, 2015). Esteves (1998) afirma que o pH pode ser considerado como uma das variáveis ambientais mais importantes, e ao mesmo tempo uma das mais difíceis de se interpretar em função do grande número de fatores que

podem influenciá-lo. No entanto, na maioria das águas 16 naturais o pH da água é influenciado pela concentração de íons H^+ originados da ionização do ácido carbônico que gera valores baixos de pH, e das reações de íons carbonato e bicarbonatos com a água, que elevam os valores de pH para a faixa alcalina (NOGUEIRA, COSTA, PEREIRA, 2015).

O potencial hidrogeniônico é representado em uma escala de 0 a 14, para temperaturas de 25°. Esses valores indicam se a solução é ácida (valores inferiores a 7), básicos ou alcalinos (valores superiores a 7), e neutra (valor igual a 7). Os valores recomendados para o consumo humano estão entre 6 e 9 (SOUZA FILHO, 2018).

1.3.3 Turbidez

De acordo com Correia *et al.* (2008), turbidez é a medida da dificuldade de um feixe de luz atravessar certa quantidade de água. Este fenômeno se dá pela presença de substâncias em suspensão, ou seja, sólidos suspensos, finamente divididos ou em estado coloidal, e de organismos microscópicos. A presença destas partículas provoca a dispersão e a absorção da luz, deixando a água com aparência turva, esteticamente indesejável e potencialmente perigosa (NOGUEIRA; COSTA; PEREIRA, 2015).

Os valores de turbidez são expressos em Unidade Nefelométrica de Turbidez (UNT). A cor da água interfere negativamente na medida da turbidez devido à sua propriedade de absorver luz. Segundo a Organização Mundial da Saúde, o limite máximo de turbidez em água potável deve ser 5 UNT (Correia *et. al.*, 2008).

No processo de tratamento, as partículas presentes na água, as quais estão associadas à turbidez, são retiradas nas etapas de coagulação e floculação. Porém, algumas vezes o processo de tratamento não consegue remover todos os materiais em suspensão presentes na água, ao passo que até chegar nas residências, a turbidez pode tender a aumentar, tendo em vista também possíveis vazamentos e partículas suspensas nas tubulações (RODRIGUES; DINIZ, 2019).

1.3.4 Sólidos Totais Dissolvidos

Os sólidos presentes na água podem ser classificados de duas formas: sólidos suspensos (sedimentáveis e não sedimentáveis) e sólidos dissolvidos (voláteis e fixos). Os sólidos

suspensos podem ser definidos como as partículas passíveis de retenção por processos de filtração. Já os sólidos dissolvidos são constituídos por partículas de diâmetro inferior a 10^{-3} μm e que permanecem em solução mesmo após a filtração (FUNDAÇÃO, 2014).

Os sólidos totais dissolvidos é um dos principais parâmetros de avaliação da qualidade da água, e são dados em mg/L ou parte por milhões (ppm), onde 1ppm equivale a 1mg/L. Conforme o Conselho Nacional de Meio Ambiente, o valor limite de concentração de sólidos totais dissolvidos, para as classes de água destinadas ao consumo humano, é de até 500mg/L (SOUZA FILHO, 2018).

1.4 SENSORES

“Na eletrônica, um sensor é conhecido como qualquer componente ou circuito eletrônico que permita a análise de uma determinada condição do ambiente, podendo ela ser algo simples como temperatura ou luminosidade” (PATSKO, 2006, p. 1).

De modo geral, os sensores podem ser classificados como analógicos ou digitais, de acordo com a forma como eles respondem à variação de estímulos. Os sensores analógicos podem assumir infinitos valores intermediários na sua saída, conforme a variação na entrada. Já os sensores digitais baseiam-se em níveis de tensão bem definidos, onde podem assumir valores quantizados, utilizando a lógica binária (PATSKO, 2006).

Existem sensores de água capazes de medir parâmetros como nível de água, temperatura, pH, condutividade, oxigênio dissolvido, e assim por diante. Geralmente esses sensores são utilizados por indústrias, de acordo com a sua aplicação (RIKA, 2019).

1.4.1 Sensor DS18B20

O termômetro digital DS18B20, ilustrado na figura 1, fornece de 9 *bits* a 12 *bits* para medições de temperatura em graus *Celsius* e tem uma função de alarme programável pelo usuário e pontos de gatilho mais baixos. O DS18B20 comunica em um barramento de 1 fio que, por definição, requer apenas uma linha de dados e aterramento para comunicação com uma central microprocessador. Além disso, o DS18B20 pode derivar energia diretamente da linha de dados, eliminando a necessidade de uma fonte de alimentação externa (MAXIM, 2019).

Figura 1 - Sensor de temperatura DS18B20.



Fonte: Autoria própria.

1.4.2 Sensor PH-4502C

O sensor PH-4502C, ilustrado na figura 2, é um sensor funcional e prático, desenvolvido para trabalhar em conjunto com microcontroladores. O conjunto é formado por um eletrodo pH e um módulo eletrônico que faz a intermediação com o microcontrolador, por exemplo, podendo o eletrodo ficar submerso no recipiente com água, ficando somente com o cabo exposto. O sensor é indicado para verificar e permitir a manutenção dos níveis de pH dentro de escalas seguras de uso em piscinas, aquários, caixas de água, rios, lagos e muito mais, permitindo manter proporções aceitáveis para pessoas, animais e plantas (USINAINFO, 2021a).

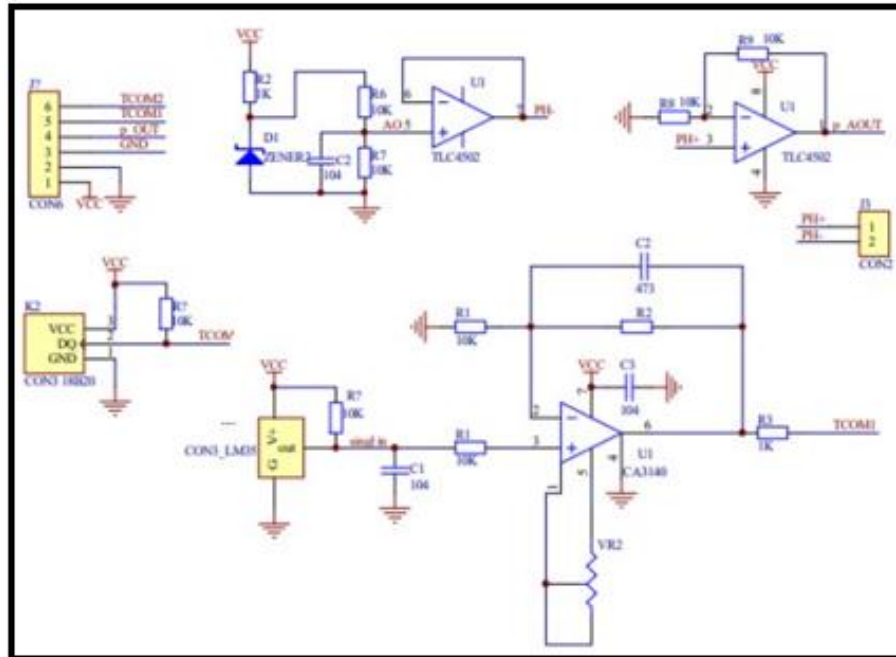
Figura 2 - Sensor de pH PH4502C.



Fonte: Autoria própria.

Na figura 3 é possível visualizar uma descrição mais detalhada dos circuitos eletrônicos que o módulo possui, assim como a funcionalidade de cada parte do mesmo, desde a parte de alimentação até a parte de leitura da sonda.

Figura 3 - Esquema Elétrico Sensor de pH PH4502C.



Fonte: (USINAINFO, 2021a).

1.4.3 Sensor SEN0189

O sensor de turbidez, ilustrado na figura 4, é capaz de medir os níveis de turbidez ou opacidade. Ele usa luz para detectar partículas suspensas na água medindo a transmitância da luz e a taxa de espalhamento, que muda com a quantidade de sólidos suspensos totais (TTS – *Total Suspense Solids*) na água. Conforme o TTS aumenta, o nível de turbidez do líquido aumenta. Sensores de turbidez são usados para medir a qualidade da água em rios e riachos, medições de águas residuais e efluentes, instrumentação de controle para tanques de decantação, pesquisa de transporte de sedimentos e medições de laboratório. Este sensor de líquido fornece modos de saída de sinal analógico e digital (USINAINFO, 2021b).

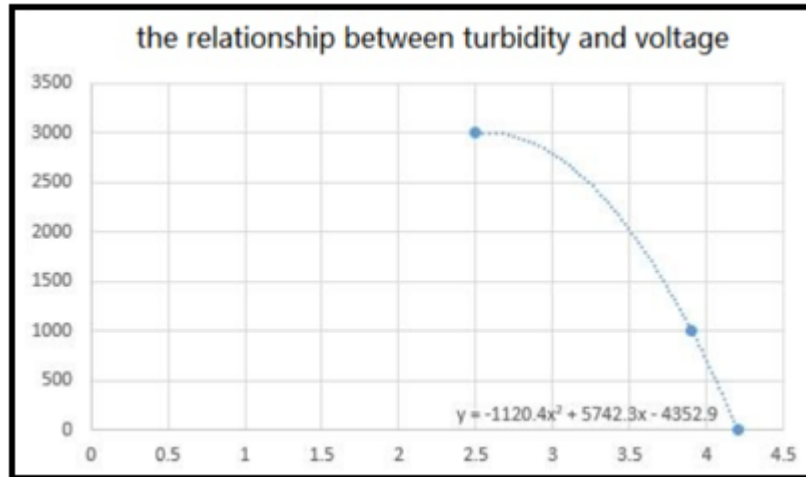
Figura 4 - Sensor de turbidez SEN0189.



Fonte: (DFROBOT, 2021, p.1).

Na figura 5 é possível visualizar o gráfico que demonstra a curva de resposta do sensor, onde estão relacionados os parâmetros de tensão por turbidez.

Figura 5 - Gráfico de Resposta do Sensor SEN0189.



Fonte: (DFROBOT, 2021, p.4).

1.4.4 Sensor TDS SEN0244

O sensor TDS, traduzido do inglês, sólidos dissolvidos totais, ilustrado na figura 6, indica quantos miligramas de sólidos solúveis são dissolvidos em um litro de água. Em geral, quanto mais alto o valor de TDS, mais sólidos solúveis são dissolvidos na água e menos limpa a água é, portanto, o valor TDS pode ser usado como um ponto de referência para refletir a limpeza da água. Este sensor pode ser utilizado em aplicações de qualidade de água, como análise de água doméstica ou hidropônica (DFROBOT, 2021).

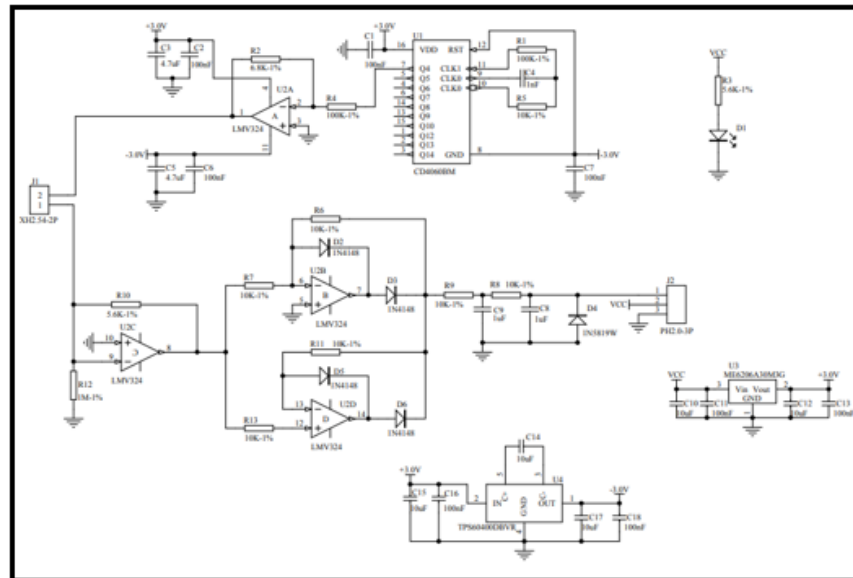
Figura 6 - Sensor TDS SEN0244.



Fonte: Autoria própria.

Na figura 7 é possível visualizar uma descrição mais detalhada do circuito eletrônico que o módulo possui, assim como a funcionalidade de cada parte do mesmo, desde a parte de alimentação até a parte de leitura da sonda.

Figura 7 - Circuito Eletrônico do Módulo TDS.



Fonte: (DFROBOT, 2021).

1.5 MICROCONTROLADOR: MÓDULO ESP 32 LORA

Microcontrolador é um dispositivo que agrupa todos os componentes para o funcionamento de um chip, podendo também ter periféricos extras que auxiliam na aplicação para o qual foi definido, bem como maior quantidade de pinos de entrada e saída (JUNIOR, 2013).

Neste trabalho, foi utilizado o módulo microcontrolado ESP32, que possui recursos de comunicação *wireless* e *bluetooth*, sendo comercializado tanto separado quanto em placas já com alguns componentes, como, por exemplo, regulador de tensão, porta micro usb e diferentes números de portas de entrada e saída, sendo que essas portas podem ser utilizadas para os sensores e atuadores (ESPRESSIF, 2021).

O ESP32 também traz recursos para economia de energia, como o modo de hibernação, que permite desligá-lo em momentos em que não houver uso, reduzindo, assim, a demanda de energia para mantê-lo em atividade. Outra característica é que seus pinos de entradas e saídas de propósitos geral têm maior precisão se comparados com outros microcontroladores, como o Arduino, pois permitem receber uma sequência de caracteres de 12 bits, enquanto que, no Arduino, a cadeia de caracteres é de 10 bits (ESPRESSIF, 2021).

Comparando-se o ESP32 com o módulo microcontrolador Arduino Uno, possui vantagens de possuir 2 núcleos, podendo, então, trabalhar com execuções paralelas, além da arquitetura ser de 32bits, diferentemente do Arduino Uno, já possui tecnologias *wireless* e *bluetooth* (BORGES, 2019). A figura 8 demonstra essa comparação.

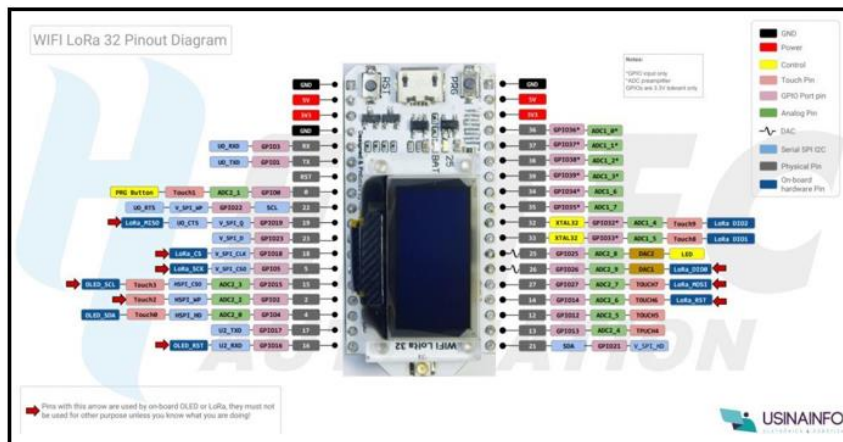
Figura 8 - Comparação entre ESP32, ESP8266 e Arduino UNO.

SPECS/BOARD	ESP32	ESP8266	ARDUINO UNO
Number of Cores	2	1	1
Architecture	32 Bit	32 Bit	8 Bit
CPU Frequency	160 MHz	80 MHz	16 MHz
WiFi	YES	YES	NO
BLUETOOTH	YES	NO	NO
RAM	512 KB	160 KB	2 KB
FLASH	16 MB	16 MB	32 KB
GPIO PINS	36	17	14
Busses	SPI, I2C, UART, I2S, CAN	SPI, I2C, UART, I2S	SPI, I2C, UART
ADC Pins	18	1	6
DAC Pins	2	0	0

Fonte: (GHOSH, 2019).

O ESP32 LoRa possui três formas distintas de comunicação: Wifi, Bluetooth e a rede de comunicação de longo alcance – LoRa, devido a utilização do chip o chip SX1278. Na figura 9 abaixo está ilustrado o diagrama *pinout* (USINAINFO, 2019).

Figura 9 - Diagrama de Pinos ESP 32 LoRa.



Fonte: (USINAINFO, 2019, p1).

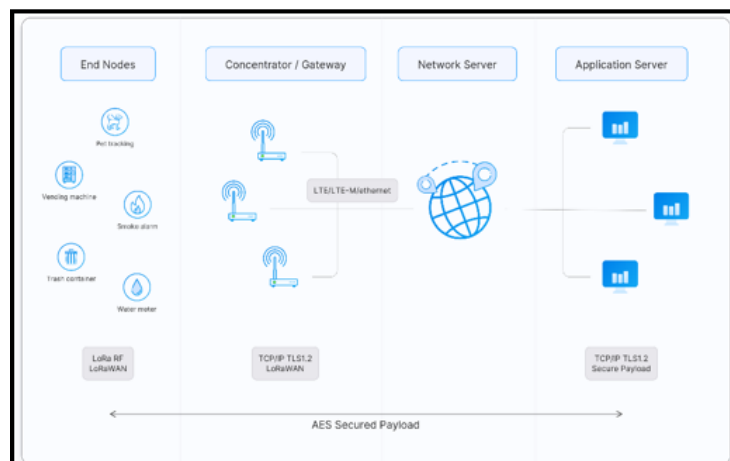
1.6 LoRaWAN E LPWAN

A especificação LoRaWAN® é um protocolo de rede de baixa potência, área ampla (LPWA) projetado para conectar 'coisas' operadas por bateria sem fio à internet em redes regionais, nacionais ou globais, e visa os principais requisitos da Internet das coisas (IoT), como comunicação bidirecional, segurança ponta a ponta, mobilidade e serviços de localização (LORA ALLIANCE, 2021).

Com o objetivo de realizar a comunicação entre os microcontroladores utilizados no projeto, e enviar os dados a uma central (nesse caso, um concentrador - *gateway*), uma rede de baixo consumo (LPWAN, do inglês *Low Power Wide Area Network*) foi escolhida. Em comparação com outras redes, como por exemplo Zigbee, Wi-Fi e Bluetooth, as redes LPWAN apresentam vantagens, quando se trata de envio de dados a longa distância (MUTHANNA *et al.*, 2018). As redes LPWAN apresentam características diferentes das outras tecnologias sem fio tradicionais, principalmente referente à área de cobertura, tecnologias tradicionais se estendem, geralmente, por questão de dezenas de metros nos melhores dos casos (RAZA; KULKARNI; SOORIYABANDARA, 2017).

A organização de uma rede LoRa se dá através de uma topologia de estrela. Nessa estrutura, os nós se conectam com um servidor central, chamado de *gateway*, através do protocolo LoRaWAN. Os *gateways*, por sua vez, estão conectados a um servidor de rede através de uma rede não-LoRaWAN, geralmente *Wi-Fi*, *Ethernet*, 3G ou 4G. Todos os dados que são enviados através do protocolo LoRaWAN possuem encriptação AES-128, o que significa que apenas a aplicação que possui as chaves corretas consegue decifrar os dados (Adelantado *et al.*, 2017). A seguir na figura 10, é possível visualizar a estrutura de uma rede LoRaWAN.

Figura 10 - Estrutura de uma rede LoRaWAN.



Fonte: (THETHINGSNETWORK, 2019, p.2).

1.7 PROTOCOLO DE COMUNICAÇÃO WIFI

De acordo com Rose (2010), WiFi é um conjunto de particularidades para redes WLAN - (Wireless Local Area Network) baseada no padrão IEEE 802.11. O nome Wi-Fi é tido como uma abreviatura do termo inglês Wireless Fidelity, embora a WiFiAlliance, empresa responsável principalmente pelo licenciamento de produtos fundamentados na tecnologia, nunca tenha afirmado tal conclusão.

Os padrões IEE 802.11b e IEE 802.11g operam em uma frequência de 2,4 GHz não licenciada chamada de ISM (industrial, scientific and medical), aplicada para fins industriais, científicos e médicos (GORON, 2013).

2 MÉTODOS E MATERIAIS

Neste capítulo é mostrado o desenvolvimento do projeto desde a pesquisa teórica em conjunto com as primeiras medições em laboratório, passando pela montagem do protótipo e testes de comunicação entre o módulo ESP 32 LoRa, até a criação do sistema e sua integração, assim estabelecendo a interface do sistema de monitoramento.

O sistema é monitorado através de quatro sensores, são eles: sensor de temperatura DS18B20, sensor de pH PH4502C, sensor de turbidez SEN0189 e o sensor TDS. Os mesmos serão instalados no protótipo feito de acrílico, onde a fiação ficará na parte interna do protótipo e os sensores ficarão na parte inferior e externa ao protótipo, em contato com a água, para que os parâmetros possam ser medidos. Em seguida, os valores de leitura são injetados nas entradas analógicas de um ESP32 LoRa, este funcionará como *slave*, onde os dados são convertidos para um valor digital e enviados para um outro ESP32 LoRa, este funcionará como *gateway*, utilizando a protocolo de comunicação LoRaWAN. O microcontrolador que possui a função de *gateway* estará, por sua vez, localizado próximo a uma rede de comunicação de dados com acesso à *internet*, para a partir daí, enviar os dados para a plataforma Arduino Iot Cloud, através de comunicação WI-FI via protocolo IEE 802.11.

Os valores discretos dos parâmetros são processados pelo *software*, que gera um *dashboard* exibindo os dados através de um gráfico, sendo possível acompanhar qualquer variação de pressão do sistema, fornecendo assim um histórico da medição. O sistema foi desenvolvido nas seguintes etapas.

Na primeira etapa, foi realizada uma revisão de literatura nas áreas de microcontroladores e microprocessadores, redes sem fio, protocolo LoRaWAN, sensores e armazenamento em banco de dados.

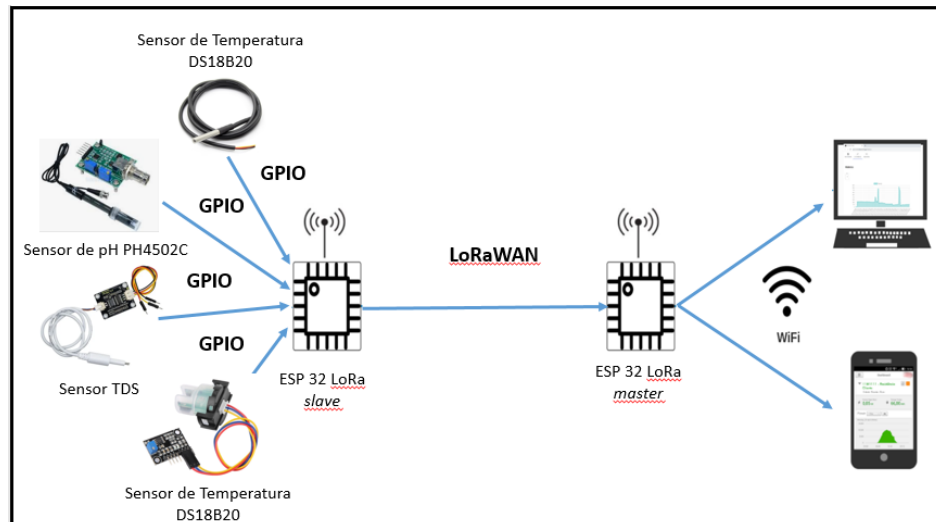
Na segunda etapa, para a implementação do protótipo, foi feito primeiramente o levantamento dos equipamentos e ferramentas a serem utilizadas, bem como uma lista de materiais e componentes para aquisição.

Na terceira etapa, montou-se o protótipo utilizando uma placa de ensaio (em inglês, *proto-board*), dois ESP32 LoRa e os sensores de temperatura, pH, TDS e turbidez. O projeto foi montado no laboratório localizado na Universidade do Estado do Amazonas, bem como o desenvolvimento do *firmware* a ser embarcado nos microcontroladores.

Na quarta etapa, foi realizada a integração dos dados com o *software* Arduino Iot Cloud, onde será possível visualizar as leituras dos sensores. E em seguida, o protótipo foi desenvolvido de forma que o mesmo flutue sobre a água, não necessitando assim de instalação.

Na quinta etapa, foram feitos os testes de validação no laboratório de biologia da Universidade do Estado do Amazonas, na unidade Ensino Normal Superior (ENS), testes de repetibilidade dos resultados, e testes robustez do protótipo, e por fim, a análise e interpretação dos resultados. A figura 11 ilustra o diagrama funcional do protótipo.

Figura 11 - Diagrama Funcional do Protótipo.



Fonte: Autoria própria.

Durante a montagem do protótipo de sistema de monitoramento foram utilizados os materiais listados abaixo.

- 1 *protoboard*.
- 1 caixa de acrílico para protótipo mecânico.
- 5 *jumper wires* macho-macho 12cm.
- 2 módulos ESP32 LoRa.
- 1 sensor de temperatura DS18B20.
- 1 sensor de pH PH4502C.
- 1 sensor de turbidez SEN0189.
- 1 sensor de TDS.
- 1 resistor de 3.3KΩ.
- 1 bateria de Recarregável 5V/1A.

3 IMPLEMENTAÇÃO DO PROJETO

Este capítulo apresenta o processo de desenvolvimento do projeto. Foram aplicadas diversas tecnologias e utilizadas diferentes plataformas de programação.

São apresentados neste capítulo os seguintes tópicos:

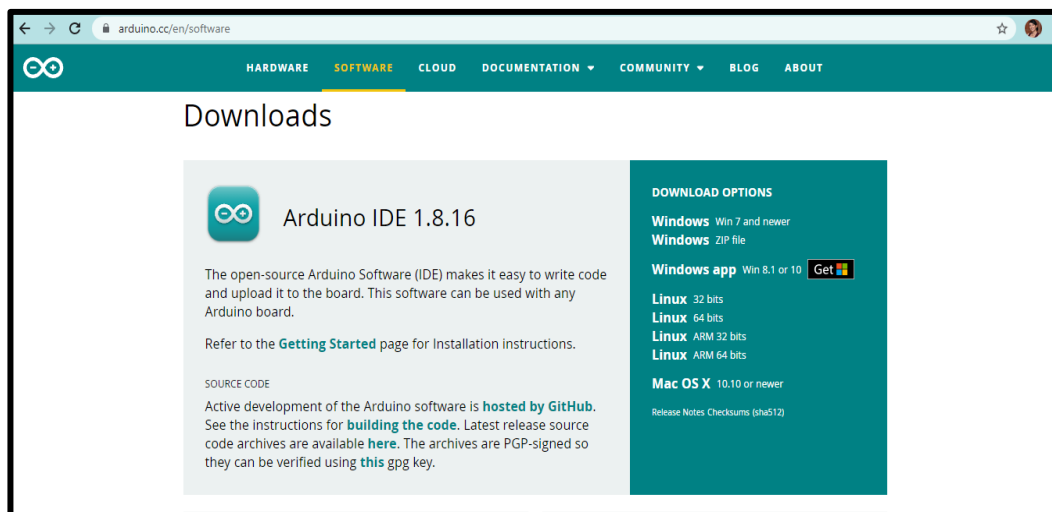
- a) instalação dos *softwares* utilizados;
- b) instalação das bibliotecas no IDE Arduino;
- c) implementação do dispositivo de aquisição de dados e comunicação;
- d) firmware do dispositivo de aquisição de dados e comunicação;
- e) desenvolvimento do protótipo;
- f) montagem do protótipo;
- e) implementação com o software Arduino IoT Cloud.

3.1 INSTALAÇÃO DOS *SOFTWARES* UTILIZADOS

3.1.1 Arduino IDE

Para a implementação do projeto foi necessário escrever um código na linguagem C, utilizando o *software* Arduino IDE, devido a compatibilidade com o microcontrolador escolhido para o projeto. O *download* da versão 1.8.16 pode ser realizado pela URL (<https://www.arduino.cc/en/software>). Na figura 12 abaixo está sendo exibido o site de onde é possível realizar o *download*.

Figura 12 - Pagina Web Arduino.



Fonte: Aatoria própria.

Na figura 11 é possível notar a disponibilidade de versões para os sistemas operacionais mais comumente utilizados e suas respectivas configurações. É necessário escolher uma das versões disponíveis de acordo com as configurações do usuário.

O *download* e a instalação são processos bem simples de serem realizados. Seguindo somente os passos da tela que irão ser mostrados durante a instalação, sendo possível concluí-la com sucesso. Após estes procedimentos de *download* e instalação, é possível ter acesso ao ambiente para o desenvolvimento do código para o projeto.

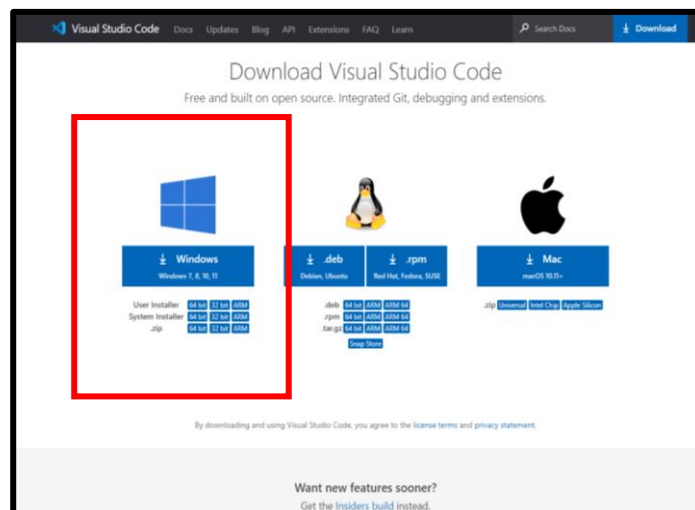
3.1.2 Visual Studio Code

Outro *software* usado no projeto foi o VS Code, ele é altamente customizável e extensível, além disso possui as seguintes funcionalidades: *autocomplete*, *intellisense*, depuração do código direto do editor e suporte nativo ao repositório git, que é um sistema de controle de versões distribuídos, usado no desenvolvimento de software, mas pode ser empregado para registrar o histórico das edições de qualquer tipo de arquivo.

Existem duas maneiras de realizar o processo de instalação: a primeira é a convencional que depende do sistema operacional, ou a segunda, utilizando o suporte ao modo portátil, ou seja, para dispositivos móveis.

Neste projeto, foi realizado o *download* pelo modo convencional, onde o instalador do *software* pode ser encontrado na URL (<https://code.visualstudio.com/Download>). O processo de instalação é bem simples, basta apenas seguir o modo padrão de instalação de programas no Windows. A figura 13 apresenta a página inicial do VS Code.

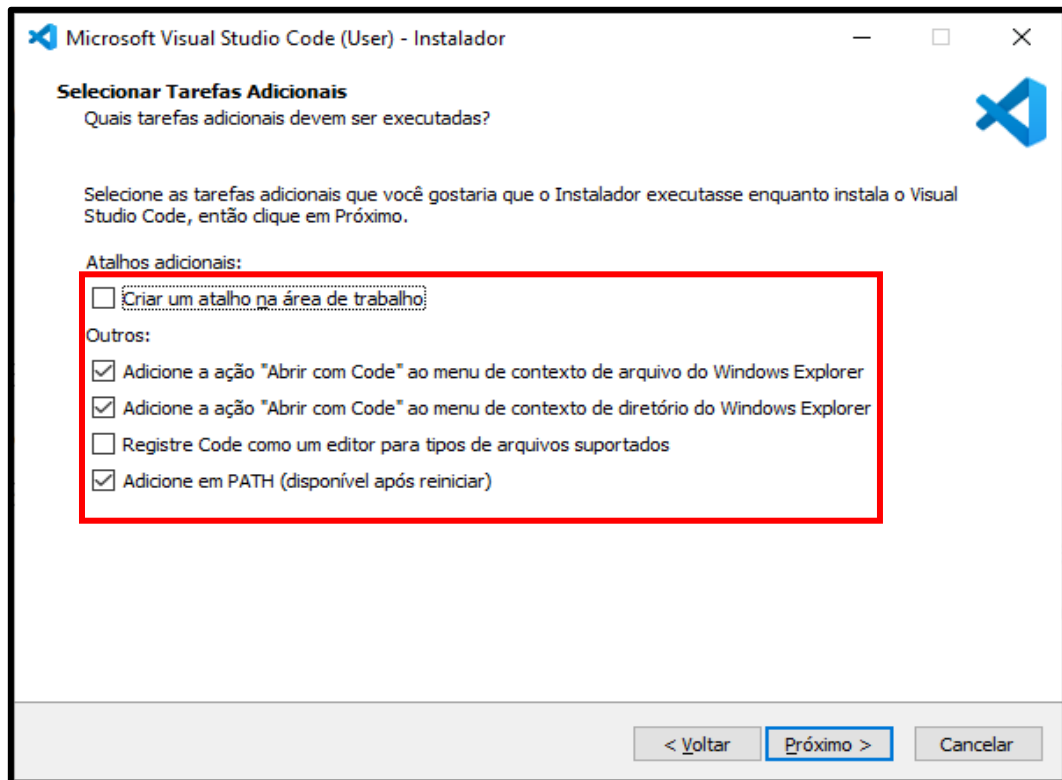
Figura 13 - Página Web VS Code.



Fonte: Autoria própria.

Deve-se garantir que na segunda tela do assistente a opção “Adicione em PATH (disponível após reiniciar)” esteja marcada, dessa maneira será possível abrir o editor pelo terminal com o comando *code*, conforme figura 14.

Figura 14 - Tela de Instalação.



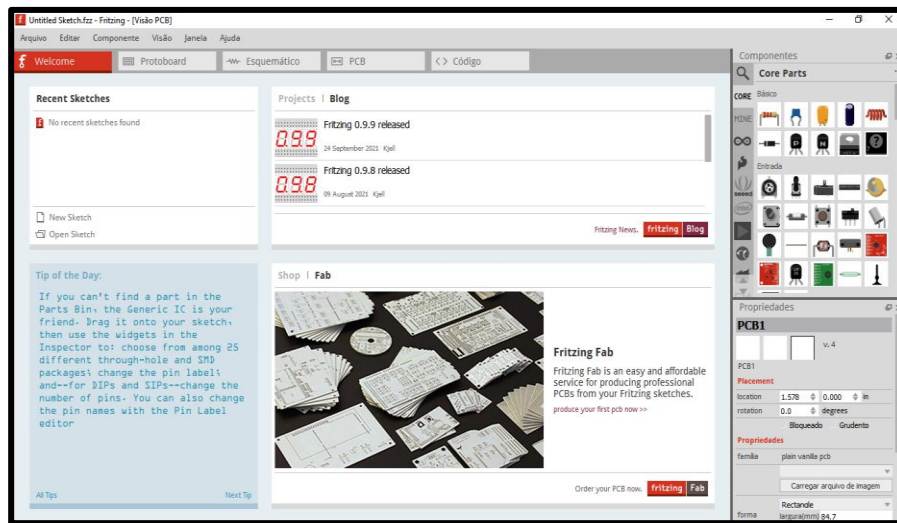
Fonte: Autoria própria.

3.1.3 Fritzing

É possível fazer o *download* do Fritzing pela URL (<https://fritzing.org/download/>), para qualquer sistema operacional, seja Windows, Mac ou Linux. Os procedimentos a seguir são para instalação do *software*:

Para instalar o Fritzing, executa-se o arquivo de instalação baixado, seguindo as instruções apresentadas. O instalador pode solicitar direitos do administrador ("UAC") durante a instalação. Deve-se confirmar esta solicitação clicando em "sim". Isso permitirá que o instalador baixe bibliotecas adicionais da Microsoft. A figura 15 apresenta a página inicial do *software*.

Figura 15 - Tela Inicial do Fritzing.



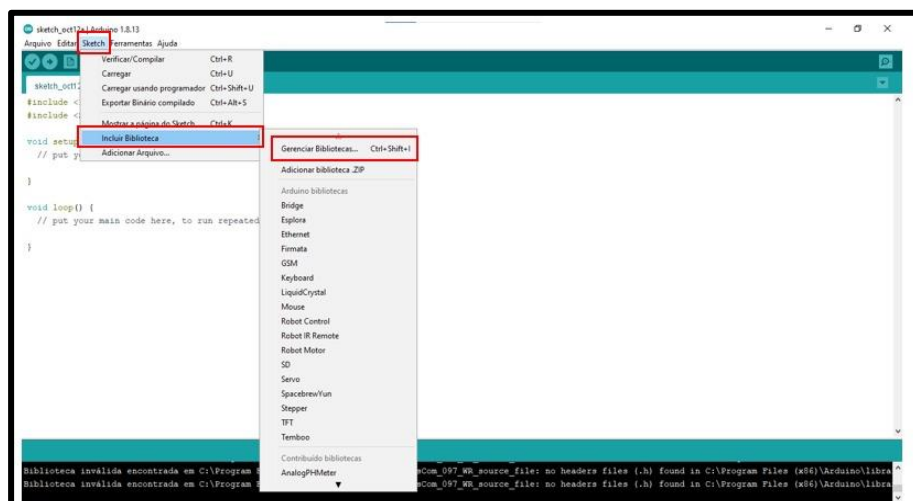
Fonte: Autoria própria.

3.2 INSTALAÇÃO DAS BIBLIOTECAS NO IDE ARDUINO.

Levando em consideração que é de suma importância simplificar os códigos visando economizar espaço na memória do programa, a IDE Arduino possui um recurso onde pode-se adicionar uma biblioteca específica para cada dispositivo capaz de facilitar a programação, ou seja, ao invés de se realizar uma longa e extensa programação, pode-se utilizar os comandos existentes na biblioteca para facilitar a execução de determinado comando e simplificar o desenvolvimento do código.

Para realização das instalações das bibliotecas no IDE Arduino, é necessário seguir os passos conforme a figura 16.

Figura 16 - Bibliotecas no IDE Arduino.



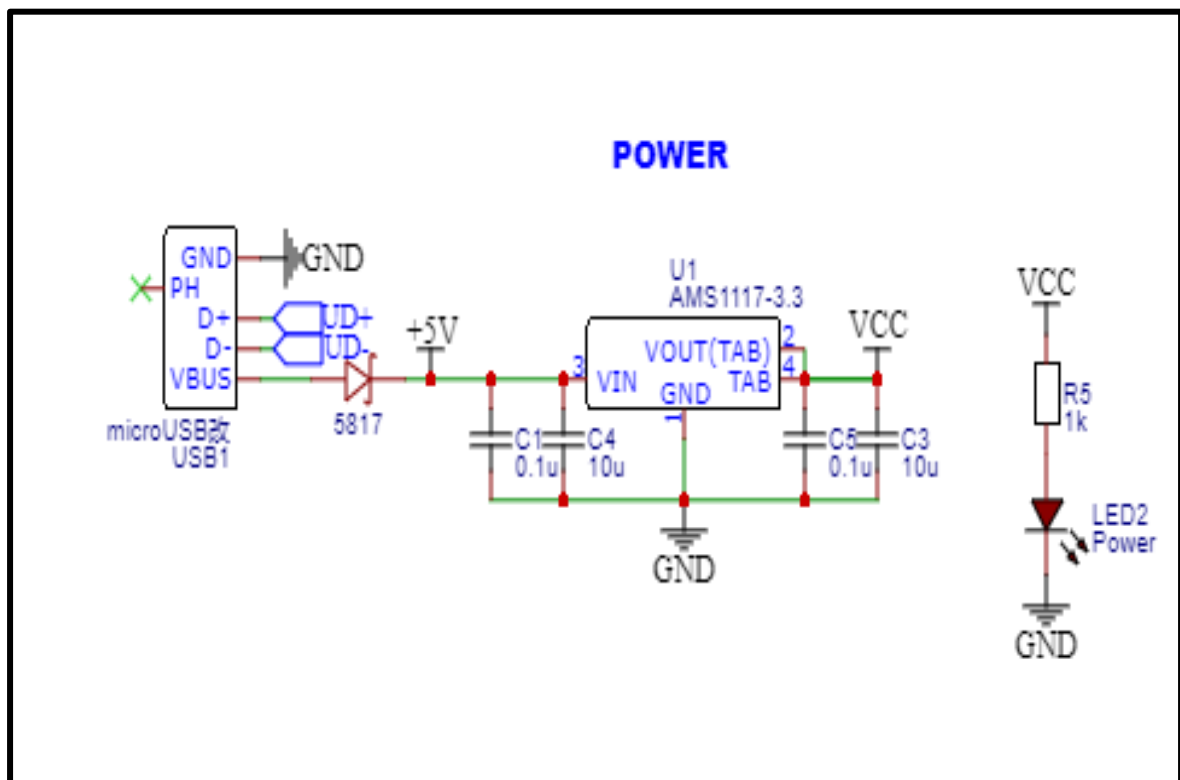
Fonte: Autoria própria.

Após entrar na tela de gerenciar bibliotecas, basta somente procurá-las pelo nome e clicar no ícone instalar. As bibliotecas instaladas foram: ESP 32, *WiFi*, *HTTPClient*, *WiFiClientSecure* e *ArduinoJson*

3.3 IMPLEMENTAÇÃO DO DISPOSITIVO DE AQUISIÇÃO DE DADOS E COMUNICAÇÃO.

O módulo ESP 32 tem um regulador de 3,3V (corrente max 1A). No *datasheet* do ESP32 especifica-se que a corrente fornecida para alimentação de 3,3V deve ser no máximo de 500 mA. Portanto, nunca se deve conectar um outro módulo ou dispositivo que consuma mais do que 0,5 A, para não sobrecarregar esse regulador interno. Na figura 17 a seguir, é possível verificar o circuito interno de alimentação do ESP32.

Figura 17 - Circuito interno de alimentação do ESP 32.

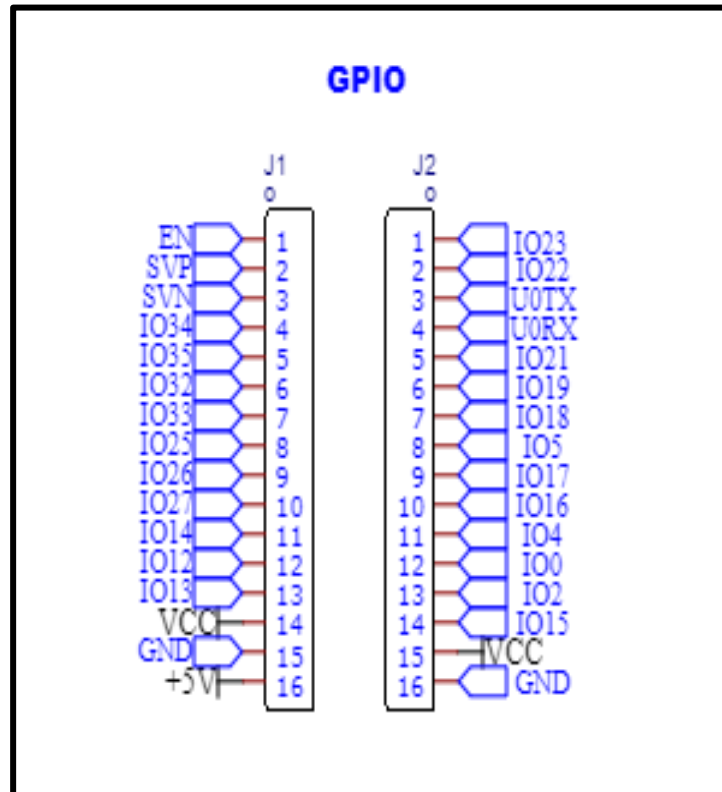


Fonte: Autoria própria.

Toda placa ESP32 já tem um regulador de 3,3V, um chip de interface Serial-USB, um conector micro-USB, um LED e botões de RESET (EN) e de LOAD (BOOT). A placa ESP32 pode ser usada similarmente como se fosse uma placa Arduino.

Os pinos de *General Purpose Input Output* (GPIO) são portas que servem como interface entre o microcontrolador e periféricos, como sensores, atuadores ou até outros circuitos integrados e microcontroladores. Na figura 18 é possível verificar a pinagem GPIO.

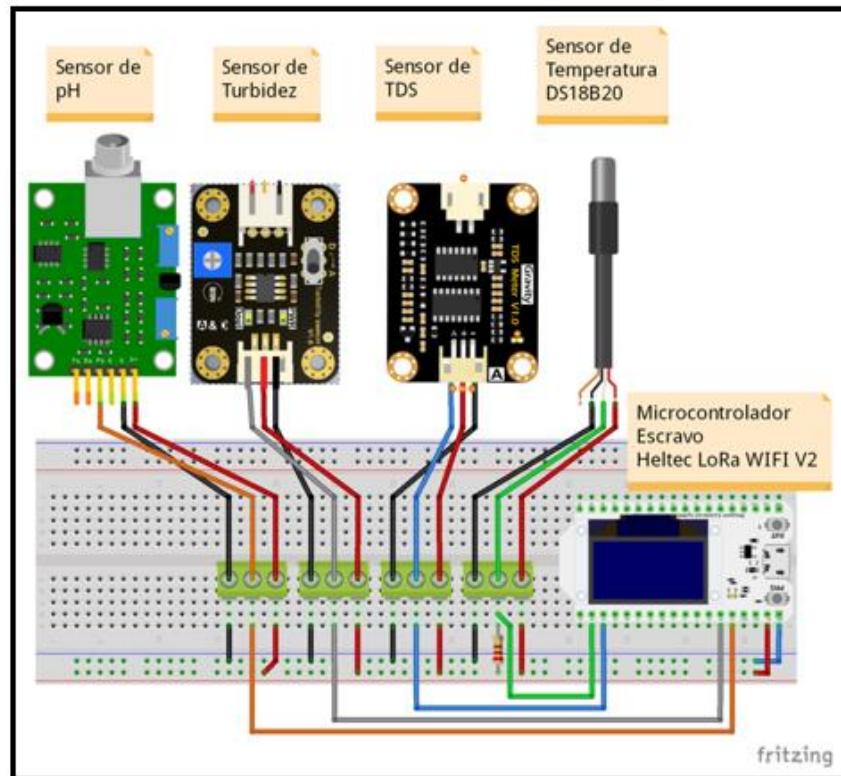
Figura 18 - Pinagem GPIO do módulo ESP 32.



Fonte: Autoria própria.

A figura 19 a seguir ilustra, a ligação elétrica que foi realizada utilizando um *protoboard*, conectores com parafusos, para evitar mal contato nas conexões, e os sensores, afim de garantir o correto funcionamento dos componentes utilizados no projeto. Os pinos utilizados foram os pinos GPIO, o pino de alimentação (5 Volts) e GND.

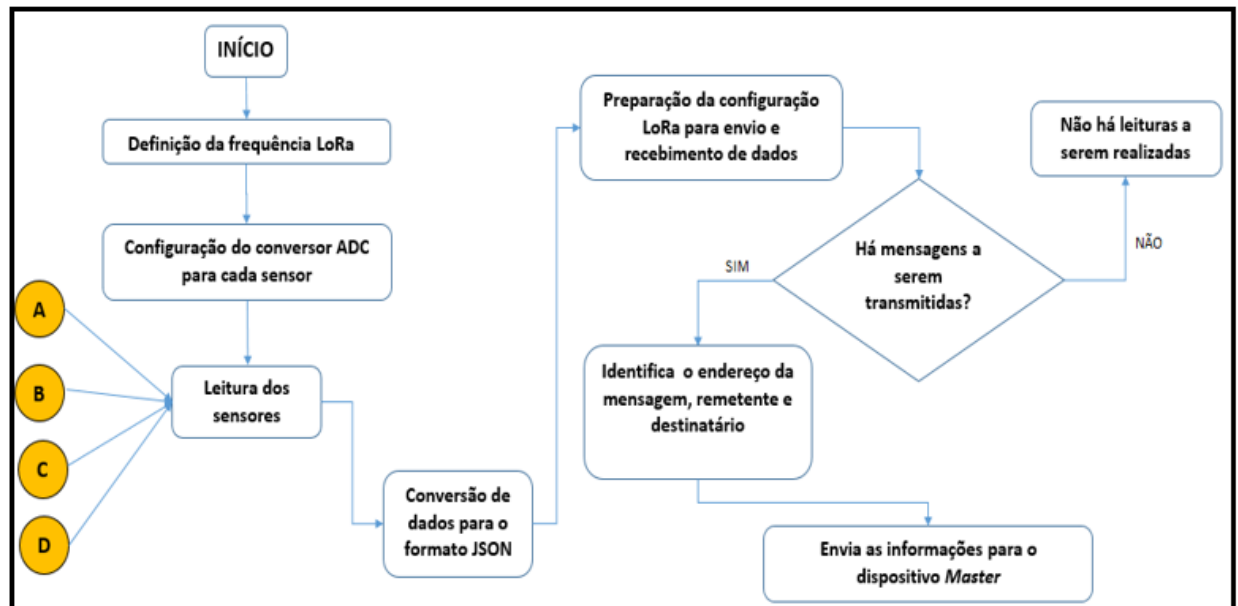
Figura 19 - Esquemático das conexões – Sensores e Módulo ESP 32.



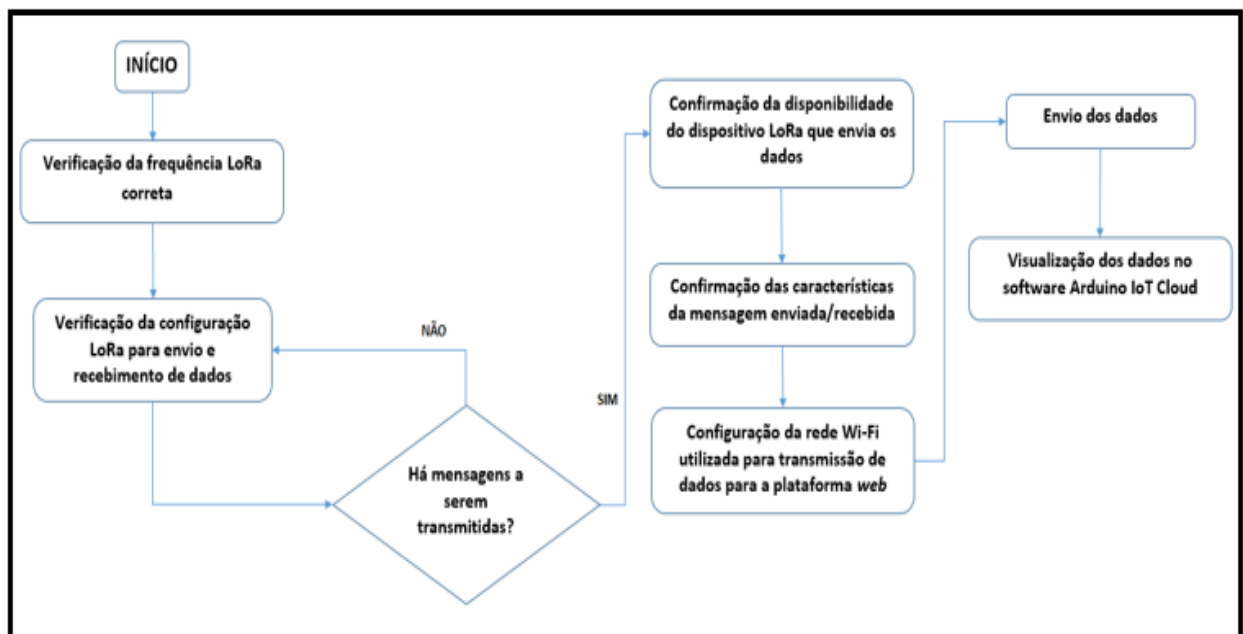
Fonte: Autoria própria.

3.4 FIRMWARES DOS DISPOSITIVOS DE AQUISIÇÃO DE DADOS E COMUNICAÇÃO

Após a escolha do microcontrolador ESP 32 foi iniciado o desenvolvimento do *firmware*. Seguindo a ordem do fluxograma, primeiramente foi realizada a configuração do conversor ADC, no qual permite ter uma precisão considerável devido a utilização do ESP 32. Em seguida, são realizadas as leituras dos sensores. Após a leitura analógica, este valor será convertido em um número que está na faixa de leitura do conversor ADC, e a partir daí, é convertido para o formato JSON, para poder ser enviado para a interface *WEB*, através de uma rede Wi-Fi. Há também a necessidade da criação de um cliente HTTP seguro, a fim de preservar os dados enviados na rede, e da definição da rota para envio dos dados. Após isto será possível visualizar as informações das leituras obtidas pelo sensor, na plataforma *WEB*. Nas figuras 20 e 21 tem-se o fluxograma que detalha a sequência de ações dos *firmwares*.

Figura 20 - Fluxograma do dispositivo *Slave*.

Fonte: Autoria própria.

Figura 21 - Fluxograma do dispositivo *Master*.

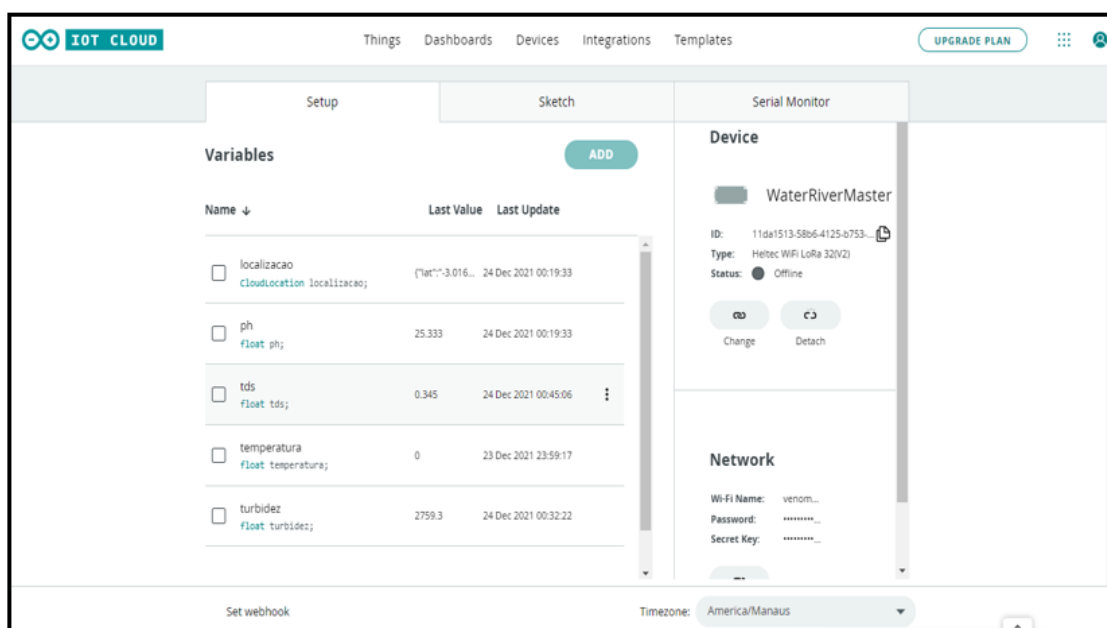
Fonte: Autoria própria.

3.5 CONFIGURAÇÃO DA PLATAFORMA ARDUINO IOT CLOUD

Para que fosse possível a visualização dos gráficos e dos valores gerados pelas medições dos sensores através da plataforma *web* foi necessário configurar uma série de

parâmetros, desde o tipo de microcontrolador utilizado, a rede Wi-Fi utilizada para enviar os dados, o tipo de sensor utilizado e sua função e o dashboard, que é o resultado final de saída do sistema. Na figura 22 a seguir, pode ser visualizada na aba de *setup* uma tela de configuração geral, onde pode-se visualizar os parâmetros configurados.

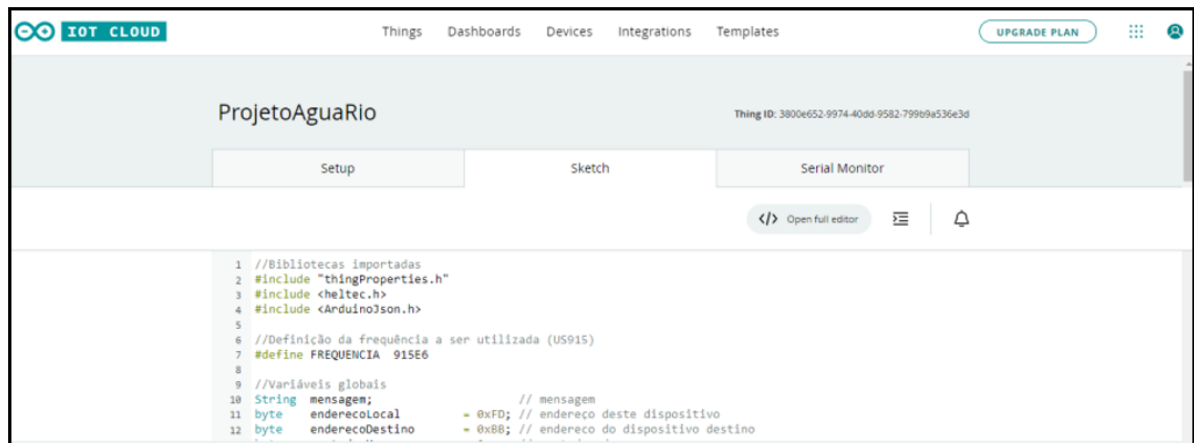
Figura 22 - Tela de configuração principal (Arduino IoT Cloud).



Fonte: Autoria própria.

Após realizada a configuração acima, a seguir na aba *Sketch* é possível adicionar o código do *firmware* que será executado pelo microcontrolador *master*. Na figura 23 a seguir, pode-se visualizar a tela onde foi adicionado o código e posteriormente o *upload* do código para gravação no microcontrolador.

Figura 23 - Tela de importação de código (Arduino IoT Cloud).

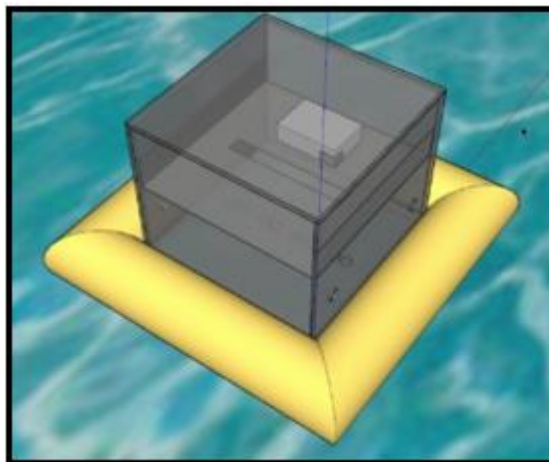


Fonte: Autoria própria.

3.6 DESENVOLVIMENTO DO PROTÓTIPO MECÂNICO

O protótipo mecânico foi desenvolvido com objetivo de acomodar o protoboard, juntamente com os sensores, visando a implementação de um modelo real do sistema de monitoramento. A referida caixa possui dimensões de 20 cm de comprimento, 20 cm de largura e 20 cm de altura, constituída por chapas de acrílico de 5 mm de espessura, além de ter acoplado em suas laterais na parte inferior, alguns pedaços de macarrão de piscina que foram cortados sob medida e colados, com o objetivo de fazer o protótipo flutuar sobre a água. O acrílico foi o material escolhido, uma vez que permite a visualização dos componentes no seu interior através de uma parede rígida e relativamente leve. A figura 24 ilustra a idealização do protótipo mecânico.

Figura 24 - Protótipo Mecânico.



Fonte: Autoria própria.

3.7 MONTAGEM DO PROTÓTIPO

Para a construção do protótipo, algumas etapas foram seguidas, dentre elas a idealização do modelo, seleção de componentes e sistemas, aquisição dos materiais, programação, montagem, integração e comissionamento. A seguir será detalhado cada uma delas.

3.7.1 Idealização do Modelo

A primeira etapa do processo de montagem do protótipo é a concepção do material que melhor se adapta ao modelo de baixo custo, uma vez que há limitações orçamentárias e de prazo. Assim, foi desenvolvido um projeto com dimensões e peso que viabilizassem a mobilidade, e ao mesmo tempo, permitissem a realização de testes e a acomodação de todos os componentes básicos necessários ao seu pleno funcionamento.

3.7.2 Seleção de Componentes e Sistemas

Foi de extrema importância a pesquisa, identificação e seleção de componentes e sistemas que compõem o modelo. Para isso, a pesquisa avançou sobre uma gama de sensores capazes de medir as características físico-químicas da água. Estes, por sua vez, foram cuidadosamente avaliados quanto a sua faixa de trabalho, precisão, custo e principalmente sobre a capacidade dos mesmos trabalharem imersos na água.

A principal dificuldade encontrada, foi o fato de existirem poucos sensores portáteis e de baixo custo, dificultando e limitando assim a escolha dos mesmos. Outro fator que dificultou a escolha dos parâmetros inicialmente, foi o valor dos sensores que medem certos parâmetros, como por exemplo o de oxigênio dissolvido. O custo de um sensor é praticamente o valor que foi gasto neste projeto por completo. Apesar das dificuldades e limitações encontradas, foi possível obter um resultado satisfatório com os sensores adquiridos.

3.7.3 Aquisição de Materiais

Os sensores selecionados foram importados, pois não foi possível encontrá-los nas lojas de componentes eletrônicos da cidade. Foi realizada uma pesquisa primeiramente em *sites* nacionais que também importavam os sensores, porém percebeu-se que o custo ainda

estava elevado. Então, todos os sensores e microcontroladores foram importados de *sites* chineses, no caso Aliexpress e Banggood. O tempo de chegada desses materiais até a cidade de Manaus, foi de aproximadamente de um mês e meio. A caixa de acrílico foi fabricada localmente, levando assim 2 dias para ficar pronta.

Os custos dos materiais utilizados para construção do protótipo na moeda Real (R\$) estão descritos conforme a tabela abaixo:

Tabela 1 - Custos do protótipo.

MATERIAL	UND	VALOR EM R\$	Frete	Total
kit Par ESP 32 LoRa	1	R\$214,87	R\$29,38	R\$244,25
DFROBOT - Sensor de turbidez SEN0189	1	R\$58,01	R\$32,01	R\$90,02
Sensor de temperatura DS18B20	1	R\$42,42	R\$5,36	R\$47,78
Sensor TDS	1	R\$106,12	R\$2,90	R\$109,02
Sensor de pH PH4502C	1	R\$158,70	R\$2,90	R\$161,60
Jumpers de conexão	15	R\$5,00	R\$2,90	R\$7,90
Protótipo Caixa de Acrílico	1	R\$250,00	-	R\$250,00
Bateria Recarregável	1	R\$30,00	-	R\$30,00
Conectores - Bourne	3	R\$3,50	-	R\$10,50
TOTAL:				R\$951,07

Fonte: Autoria própria.

Em se tratando de materiais e componentes importados, a entrega dos mesmos foi realizada pela Empresa Brasileira de Correios e Telégrafos (ECT). Com isso, o custo total do projeto foi de R\$ 951,07, sem considerar as mais de 300 horas de trabalho desde a seleção dos componentes, compra, recebimento e montagem do sistema.

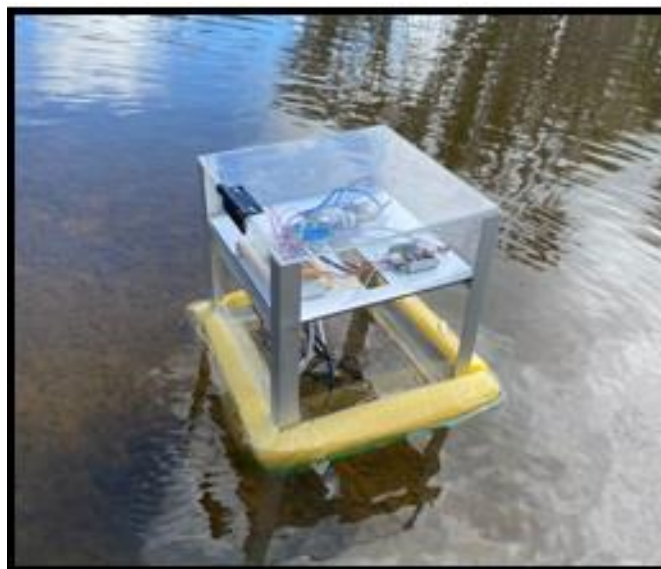
3.7.4 Montagem e Integração

Uma vez adquiridos os materiais, a montagem dos componentes foi executada com facilidade, uma vez que as conexões foram realizadas conforme as especificações dos respectivos *datasheets* disponibilizados por cada fabricante.

A integração dos sensores com o microcontrolador ESP 32 LoRa *Slave* foi realizada através do desenvolvimento de um código, utilizando a linguagem de programação C e o *software* Arduino IDE. Foi desenvolvido também um outro *firmware* para o ESP 32 LoRa *Master*, para que o mesmo execute a função de *gateway* e envie os dados recebido pelo ESP32 LoRa *Slave* para o plataforma IoT. Após o desenvolvimento dos *firmwares* foi necessário fazer a configuração na plataforma Arduino IoT Cloud, onde foi necessário configurar a rede Wi-Fi que seria utilizada para a comunicação, o tipo de microcontrolador a ser usado, e

parâmetros de comunicação, para que fosse possível estabelecer uma conexão entre o microcontrolador e a plataforma, com o objetivo de que fosse possível visualizar os dados coletados pelos sensores. A figura 25 demonstra a integração do protótipo completo.

Figura 25 - Protótipo Completo.



Fonte: Autoria própria.

3.7.5 Comissionamento

Por fim, foi realizada a etapa de comissionamento, onde foram testados e avaliados os pontos mais relevantes do projeto, tais como: testes de continuidade elétrica, teste de integridade de componentes, testes de comunicação, teste de leitura de dados, calibração e validação das leituras de dados e teste de estabilidade de medição.

Nos testes de continuidade elétrico foi verificado a integridade dos circuitos através da medição de continuidade, e foram realizadas medições de corrente e tensão, de forma a eliminar a possibilidade de curto, de fuga de corrente e de perda de tensão

Nos testes de integridade de componentes foi verificado o estado geral de cada elemento do sistema, observando possíveis danos físicos, presença de umidade, e avarias tanto no sensor, quanto no microcontrolador.

Testes de comunicação foram realizados com objetivo de verificar primeiramente a conexão e envio de dados entre os microcontroladores ESP32 LoRa, e em seguida com a rede WiFi. Em seguida, foram verificados o correto envio de dados por parte do microcontrolador, e a correta recepção dos mesmos dados na interface web. Neste ponto o ESP32 se mostrou

perfeitamente compatível com os sensores, podendo transmitir as informações via protocolo LoRaWAN e WiFi com velocidade aceitável e de forma praticamente instantânea.

Nos testes de leitura de dados, estes foram verificados e traduzidos com absoluta integridade e dentro das expectativas do modelo.

Na etapa de calibração e validação foram validadas as leituras dos sensores utilizando os equipamentos a seguir para comparar as medições: HI98194 – capaz de medir os parâmetros de temperatura, pH e TDS da água; e o HI98703-02 – capaz de medir a turbidez. Ambos os aparelhos foram utilizados nas dependências do laboratório de biologia da Universidade do Estado do Amazonas (UEA) na unidade Escola Normal Superior (ENS).

Nos testes de estabilidade de medição, o tempo de resposta da medição dos parâmetros através da leitura do sinal via interface de web, se mostrou estável e contínua. Os sensores sempre apresentaram poucas variações, consideradas normais e aceitáveis para medição da qualidade da água.

4 RESULTADOS OBTIDOS

Este capítulo mostra os resultados obtidos após a implementação do projeto. Importante frisar, a atenção que foi dedicada na coleta e registro de dados e dos custos envolvidos, uma vez que este seria apenas um primeiro passo para a implementação de um sistema mais robusto, e capaz de funcionar em um modelo de escala comercial.

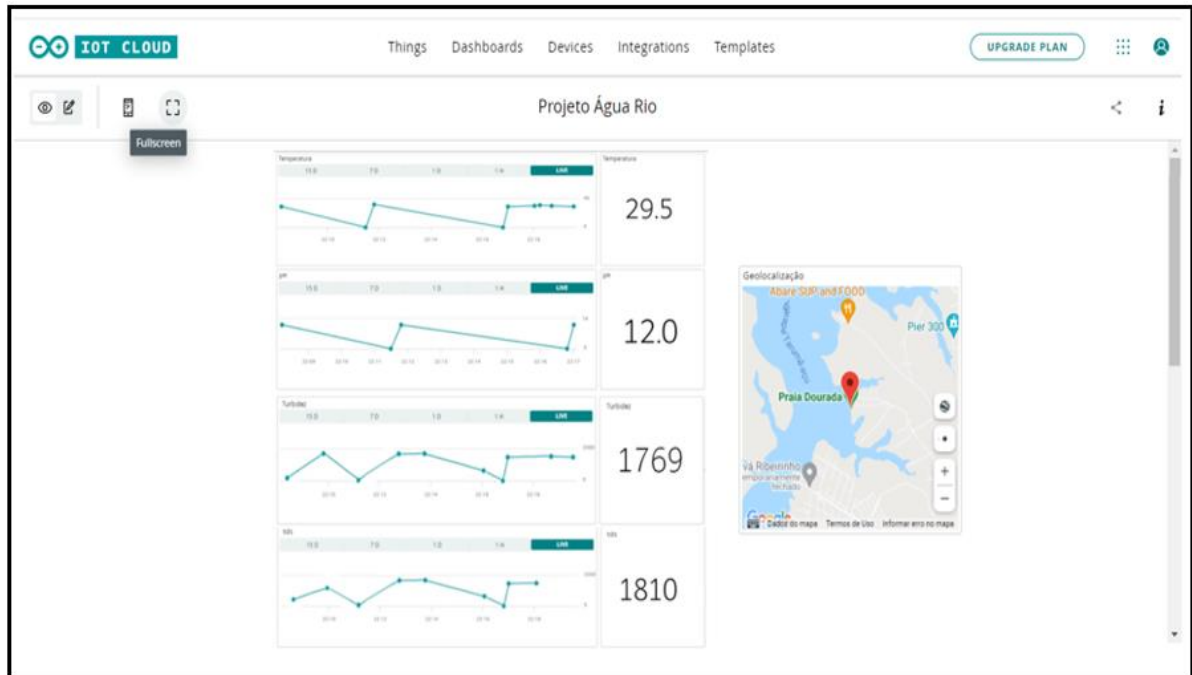
4.1 RESULTADO DO SISTEMA DE MONITORAMENTO

O sistema construído a partir do protótipo se mostrou de certa forma complexo, principalmente devido as adaptações de código implementadas para troca de dados entre os microcontroladores ESP32 utilizando o protocolo de comunicação LoRaWAN e, posteriormente o envio dessas informações utilizando o protocolo WiFi. Apesar da pouca variedade de sensores portáteis para mensurar os parâmetros de qualidade da água, foi possível adquirí-los e utilizá-los da maneira esperada. O funcionamento e medições corretas foram possíveis de ser comprovadas graças a execução dos testes de calibração e validação mencionados anteriormente, o que tornou o experimento bastante preciso e confiável, uma vez que não existem ainda projetos semelhantes a esse, principalmente para a região Amazônica. No entanto, apesar das poucas opções, para que houvesse precisão nos dados obtidos a partir das leituras, foi necessário escolher com certo critério os sensores para que fosse possível obter um boa precisão nas medições, uma vez que, tornam-se questionáveis as medições dos parâmetros da água em um protótipo que fica ao ar livre, porém, pode-se notar que obteve-se bastante sucesso no resultado, o qual será exposto a seguir.

4.2 INTERFACE GRÁFICA

A interface gráfica construída para a obtenção dos dados se deu através da criação de um *dashboard* utilizando o *software* Arduino IoT Cloud, onde a própria plataforma disponibiliza recursos para criação de tal. Através da figura 26 a seguir, é possível visualizar os gráficos obtidos de acordo com as medições realizadas de cada parâmetro. É possível também observar a hora que foram sendo realizaas as medições, além de ter acesso ao historico de medição conforme o período de tempo determinado pelo *software*. Foi adicionado também o parametro de geolocalização tornando possível observar a localização do protótipo, que utiliza o microcontrolador com a função *slave*.

Figura 26 - Página do Dashboard.



Fonte: Autoria própria.

4.3 TESTES E CALIBRAÇÃO DOS SENSORES

Diante da necessidade tornar confiáveis as leituras realizadas pelos sensores utilizados no protótipo, os mesmos foram submetidos a um processo de testes e calibração, onde foram realizadas leituras de amostras de água e comparações com aparelhos específicos para cada parâmetro.

O primeiro aparelho utilizado como referência foi o medidor multiparâmetro do fabricante HANNA INSTRUMENTS, modelo HI98194, ilustrado na figura 27 abaixo. Possui proteção à água que monitoriza até 12 parâmetros da qualidade da água, incluindo 6 medidos e 6 calculados, entre os quais estão três dos quatro parâmetros escolhidos para monitorar neste projeto: temperatura, pH e TDS.

Figura 27 - Medidor Multiparâmetro HI98194.



Fonte: (HANNA INSTRUMENTS, 2021a, p.1).

O segundo aparelho utilizado também é do fabricante HANNA INSTRUMENTS, porém é utilizado para medição do parâmetro turbidez. O modelo em questão é o HI98703-02, o qual é um medidor de turvação de elevada precisão. O HI98703-02 está em conformidade com o método da EPA 180.1 e com o método 2130 B do Standard Methods for the Examination of Water and Wastewater para medições da turvação. A seguir, na figura 28 é possível visualizar a ilustração do mesmo.

Figura 28 - Medidor de Turvação HI98703-02.



Fonte: (HANNA INSTRUMENTS, 2021b, p.1).

Primeiramente foi realizado o procedimento de calibração do sensor de pH, modelo PH4502C, onde foi utilizado como referência o medidor de modelo HI98194. A sonda do aparelho de referência e o sensor do projeto estão imersos na mesma amostra de água, onde o medidor indica a leitura de 6,99 e o sensor indica as medições de 6,99 a 7,00, conforme a figura 28 abaixo.

Figura 29 - Validação das medições do sensor de pH PH4502C.

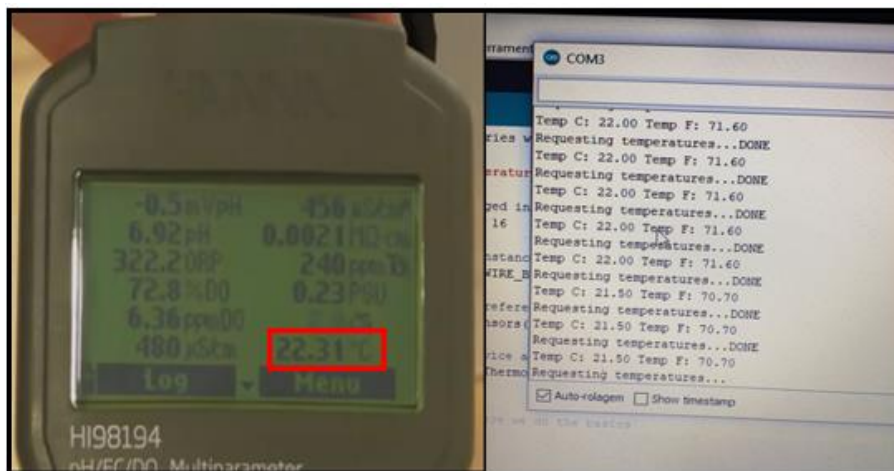


Fonte: Autoria própria.

Para a obtenção deste pH na amostra de água, foi utilizada uma solução de referência, a qual também podemos visualizar na imagem. A solução é comercializada pelo mesmo fabricante do medidor (HANNA, 2021b), possui um pH fixo, no caso em questão de 7.01. Houve essa pequena variação na medição devido a solução já ter sido aberta e utilizada anteriormente, o que pode ocasionar esta pequena variação, porém há legitimidade na validade dos resultados.

O processo se repete para a validação da medida de temperatura, onde utilizou-se também o medidor HI98194 para comparar as leituras obtidas pelo sensor DS18B20. A calibração de temperatura do medidor é automática, realizada através do próprio *software* do medidor que realiza as compensações necessárias. O resultado obtido foi bastante satisfatório, como podemos visualizar na figura 30 uma vez que o sensor possui precisão de 0,5 graus.

Figura 30 - Validação das medições do sensor de temperatura DS18B20.



Fonte: Autoria Própria.

Segue na sequência, conforme a tabela 2, os valores de temperatura utilizados para teste e comparação com o medidor de referência.

Tabela 2 – Tabela de valores utilizados para medir e comparar resultados.

Temperatura (°C)	Valor Medido (Medidor de Referência)	Valor Medido (Sensor)
10 °C	10.00 °C	10.0 °C
25 °C	25.02 °C	25.0 °C
50 °C	50.02 °C	50.0 °C
75 °C	75.03 °C	75.5 °C
100 °C	100.03 °C	100.5 °C

Fonte: Autoria Própria.

Para o sensor de TDS também foi utilizado o medidor HI98194 para validação das medições. Para obter medições precisas, inicialmente o medidor é calibrado com uma solução padrão de TDS, disponibilizada pelo mesmo fabricante do equipamento, de 1382 mg/L (ppm), como a da ilustração na figura 31 abaixo.

Figura 31 – Solução Padrão TDS.



Fonte: (HANNA INSTRUMENTS, 2021c, p.1).

Em seguida após a calibração foram realizados as medições e comparações com o sensor utilizado. Obteve-se um resultado satisfatório, pois as medições obtidas pelo sensor TDS, foram bem aproximadas às medidas obtidas pelo medidor de referência, como é possível observar abaixo na figura 32.

Figura 32 - Validação das Medições do Sensor TDS.



Fonte: Autoria Própria

E para finalizar os testes e calibração dos sensores , por ultimo foi testado o sensor de turbidez SEN0189, utilizando como referência na comparação dos resultados, o medidor HI98703-02. Na tabela 3 abaixo, estão descritos os valores utilizados como referência para calibração e obtidos através das medições.

Tabela 3 – Validação das Medições do Sensor de TDS.

Turbidez (ppm)	Valor Medido (Medidor de Referência)	Valor Medido (Sensor)
15 ppm	15.03 ppm	15 ppm
100 ppm	100.05 ppm	101 ppm
750 ppm	750.06 ppm	753 ppm
2000 ppm	2000.10 ppm	2002 ppm

Fonte: Autoria Própria

CONCLUSÃO

No desenvolvimento da presente pesquisa foram realizadas revisões acerca de assuntos como: sensores para medição de parâmetros da água, técnicas de conversão e transmissão de sinais, microcontroladores, protocolos de comunicação, além das tecnologias mais recentes aplicadas e seus parâmetros de configuração.

Os estudos e pesquisas realizados por mais de 1 (um) ano não apontaram qualquer dispositivo ou solução semelhante, que esteja sendo testado ou utilizado em aplicação do tipo. Atualmente o monitoramento das águas de qualquer que seja o rio das bacias amazônicas é realizado de forma convencional e manual, dispondo de pessoas treinadas para fazer a coleta das amostras, mas mesmo assim, podem comprometer os resultados obtidos por uma contaminação acidental ao coletar a amostra.

Houve uma certa dificuldade para selecionar os sensores, uma vez que ainda não existe uma variedade de sensores portáteis e com custo acessível para a aplicação do projeto em questão.

A maior dificuldade do trabalho foi a falta de uma base teórico-científica consolidada, ou estudos anteriores que pudessem ser o ponto de partida para as análises. Assim, todas as hipóteses testadas foram baseadas em inovação, desenvolvidas a partir de conhecimento teórico aplicado. No entanto, após a montagem do protótipo, é possível notar a qualidade dos componentes utilizados, e a sua fácil integração com os sistemas mais modernos de leitura e transmissão de dados. Foram obtidas leituras precisas, com tempo de resposta excelente e boa confiabilidade para um modelo em escala (protótipo).

Quanto a hipótese defendida, ela se mostrou plenamente factível, visto que foi, de fato, possível construir um sistema de baixo custo, para o monitoramento em tempo real da qualidade da água de bacias amazônicas capaz de mensurar os parâmetros determinados para este projeto.

Esta pesquisa viabiliza que a solução seja utilizada em um estudo mais amplo e robusto, considerando a possibilidade de utilização de outros sensores, uma rede maior e mais estruturada para comunicação dos dados entre os microcontroladores e redes de internet, podendo assim ter um sistema mais robusto e com maior área de cobertura, tornando assim, o monitoramento mais eficiente e adequado, informando assim as verdadeiras condições da água quando trata-se de mensurar a qualidade.

O próximo passo, que ainda pode ser desenvolvido em ambiente acadêmico, seria a ampliação da pesquisa para identificação ou desenvolvimento de novos sensores,

principalmente sensores portáteis, capazes de mensurar os parâmetros da qualidade da água, com características específicas que possibilitem a sua utilização em campo. A implementação de algoritmos em inteligência artificial que permitam que o sistema possa dar diagnósticos em um curto período de tempo ao medir os parâmetros da água em determinado trecho do rio e como fonte sustentável para sua bateria no protótipo, seria viável a utilização de uma placa de painel solar acoplada a bateria do circuito, dando assim uma vida útil maior ao protótipo.

Com os recursos disponíveis será possível reunir os recursos e a tecnologia necessários para levar a solução a um patamar que possa fazer a diferença na vida das pessoas e do meio ambiente, trazendo maior segurança, um tempo de ação mais rápido, das autoridades responsáveis, à possíveis contaminações ambientais que possam estar surgindo em determinada região do rio.

REFERÊNCIAS

- ADELANTADO, F. et al. **Understanding the limits of lorawan**. IEEE Communications magazine, IEEE, v. 55, n. 9, p. 34–40, 2017.
- AGÊNCIA NACIONAL DE ÁGUAS. **Indicadores de qualidade - Índice de Qualidade das Águas (IQA)**. Brasília, 2005a. Disponível em: <<http://pnqa.ana.gov.br/indicadores-indiceaguas.aspx>>. Acesso em: 15 de outubro de 2021.
- AGÊNCIA NACIONAL DE ÁGUAS. **Panorama da Qualidade das Águas Superficiais no Brasil**. Brasília, 2005b. Disponível em: <<http://www.ana.gov.br>>. Acesso em: 12 out. 2021.
- ASBRAN. **Cerca de 2 milhões de pessoas morrem no ano por causa de água**. Portal da ASBRAN - Associação Brasileira de Nutrição. 2011. Disponível em: <<https://www.asbran.org.br/noticias/cerca-de-2-milhoes-de-pessoas-morrem-no-ano-por-causa-de-agua>> Acesso em: 14 de dezembro de 2021.
- BORGES, E. **Sistema de automação residencial utilizando a tecnologia ESP32 como alternativa de baixo custo**. 2019. Disponível em <<https://repositorio.bambui.ifmg.edu.br/index.php/ec/article/view/77>>. Acesso em: 08 out. 2021.
- BRASIL. **Lei nº 9.433 de 08 de janeiro de 1997. Institui a Política Nacional de Recursos Hídricos, cria o Sistema Nacional de Gerenciamento de Recursos Hídricos, regulamenta o inciso XIX do art. 21 da Constituição Federal, e altera o art. 1º da Lei nº 8.001, de 13 de março de 1990, que modificou a Lei nº 7.990, de 28 de dezembro de 1989**. Diário Oficial [da] República Federativa do Brasil, Brasília, DF, 09 jan. 1997.
- CARVALHO, Adriana Rosa; SCHLITTLER, Flávio Henrique Mingante; TORNISIELO, Valdemar Luiz. **Relações da atividade agropecuária com parâmetros físicos químicos da água**. São Paulo, 2000. Disponível em: <<https://pdfs.semanticscholar.org/a527/80438e4b1fb07e379484717fe4617190f0e7.pdf>>. Acesso em: 25 set. 2021.
- CORREIA, Aislan; BARROS, Erick; SILVA, Jadiael; RAMALHO, Jamilson. **Análise da Turbidez da Água em Diferentes Estados de Tratamento**. Paulo Afonso, 8o Encontro Regional de Matemática Aplicada e Computacional, 2008. Disponível em: <https://di-map.ufrn.br/~sbmac/ermac2008/Anais/Resumos%20Estendidos/Analise%20da%20turbidez_Aislan%20Correia.pdf>. Acesso em: 10 out. 2021.
- DFROBOT. **Gravity Analog TDS Sensor Meter for Arduino SKU SEN0244**. [S.l.], 2021. Disponível em: <https://wiki.dfrobot.com/Gravity__Analog_TDS_Sensor__Meter_For_Arduino_SKU__SEN0244#target_1>. Acesso em 10 out. 2021.
- ESPRESSIF SYSTEMS. **ESP32 Series Datasheet, 2021**. Disponível em: <https://www.espressif.com/sites/default/files/documentation/esp32_da-tasheet_en.pdf>. Acesso em: Acesso em: 08 out. 2021.
- ESTEVES, Francisco de Assis. **Fundamentos de Limnologia**. Interciência, Rio de Janeiro, n.2ª, 1998. Disponível em: <http://professor.ufop.br/sites/default/files/roberthfagundes/files/fundamentos_de_limnologia_-_francisco_de_assis_esteves.pdf>. Acesso em: 18 de dezembro de 2021.

FERREIRA, Rafael Damiani et al. **Atualização e correção do delineamento de áreas alagáveis da bacia Amazônica**. Simpósio Brasileiro de Sensoriamento Remoto (SBSR), v. 16, p. 5864-5871, 2013.

FISCH, Gilberto; MARENCO, José A.; NOBRE, Carlos A. **Uma revisão geral sobre o clima da Amazônia**. Acta amazônica, v. 28, n. 2, p. 101-101, 1998.

FUNDAÇÃO NACIONAL DE SAÚDE. **Manual de Controle da Qualidade da Água para Técnicos que Trabalham em ETAS**. Brasília: 2014. Disponível em: <<http://www.funasa.gov.br/documents/20182/38937/Manual+de+controle+da+qualidade+da+%C3%A1gua+para+t%C3%A9cnicos+que+traham+em+ETAS+2014.pdf/85bbdcbc-8cd2-4157-940b-90b5c5bcfc87>>. Acesso em: 27 set. 2021.

GHOSH, A. **What Are the Advantages of EPP32 Over Arduino UNO?**. Techa Peek. 2019. Disponível em: <<https://thecustomizewindows.com/2019/05/what-are-the-advantages-of-epp32-over-arduino-uno/>>. Acesso em: 08 out. 2021.

GIATTI, Leandro Luiz; CUTOLO, Silvana Audrá. **Acesso à água para consumo humano e aspectos de saúde pública na Amazônia Legal**. *Ambient. soc.*, São Paulo, v. 15, n. 1, p.93-109, Apr. 2012. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1414-753X2012000100007&lng=en&nrm=iso>. Acesso em: 13 de março de 2021.

GORON, Leonardo Padua. **Avaliação de Desempenho e Interferências em Redes Wi-Fi**. Porto Alegre, 2013.

HANNA INSTRUMENTS. **HI98194 Medidor para pH, ORP, EC, salinidade, OD, pressão atmosférica e temperatura**. [S.l], 2021a. Disponível em: <<https://www.hanna.pt/produto/hi98194>>. Acesso em: 21 de dezembro de 2021.

HANNA INSTRUMENTS. **HI98703-2 Medidor de Turvação (EPA)**. [S.l], 2021b. Disponível em: <<https://www.hanna.pt/produto/hi98703-02>>. Acesso em: 21 de dezembro de 2021.

HANNA INSTRUMENTS. **Solução Padrão de TDS 1,382 mg/L (ppm) (Frasco de 500 mL)** [S.l], 2021c. Disponível em: <<https://hannainst.com.br/produto/solucion-estandar-de-tds-de-1382-mgl-ppm-frasco-de-500-ml/>>. Acesso em: 21 de dezembro de 2021.

INPE. **A estimativa da taxa de desmatamento por corte raso para a Amazônia Legal em 2019 é de 9.762 km²**. Portal de Notícias do Instituto Nacional de Pesquisas Espaciais (INPE). 2019. Disponível em: <http://www.inpe.br/noticias/noticia.php?Cod_Noticia=5294>. Acesso em: 13 de março de 2021.

INSTITUTO DE GEOCIÊNCIA. **Qualidade das Águas**. Pará, 2015. Disponível em: <https://aedmoodle.ufpa.br/pluginfile.php/360668/mod_resource/content/1/Notas%20de%20Aula%20Qualidade%20das%20%C3%81guas.pdf>. Acesso em: 5 set. 2021.

JUNIOR, V. P. da S. **Microcontroladores PIC 16F e 18F: Teoria e Prática**. 1. ed. São Paulo: Instituto Newton C Braga, 2013.

LOBATO, Alice. **Mercúrio do garimpo contamina peixes dentro e fora da Amazônia.** Amazônia Real, 2021. Disponível em: <<https://amazoniareal.com.br/mercurio-do-garimpo/11>>. Acesso em: 20 de dezembro de 2021.

LORA ALLIANCE, **O que é a especificação LoRaWAN.** [S.l.], 2021. Disponível em: <<https://lora-alliance.org/about-lorawan/>>. Acesso em 21 de dezembro de 2021.

MAXIM INTEGRATED PRODUCTS. **DS18B20 1-Wire Digital Thermometer.** [S.l.], 2019. Disponível em: <<https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>>. Acesso em: 9 out. 2021.

MUTHANNA, Mohammed Saleh Ali; MUTHANNA, Mohammed Manea Ahmed; KHAKIMOV, Abdukodir; MUTHANNA, Ammar. **Development of intelligent street lighting services model based on LoRa technology**, 2018. IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus). IEEE, 2018.

MYERS, D. N. **Why monitor water quality? United States Geological Survey.** 2017. Disponível em: <https://water.usgs.gov/owq/WhyMonitorWaterQuality.pdf> . Acesso em: 13 de março de 2021.

NEILL, Christopher et al. **Hydrological and biogeochemical processes in a changing Amazon: Results from small watershed studies and the large-scale biosphere-atmosphere experiment** . 2006. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.447.8879&rep=rep1&type=pdf> . Acesso em: 13 de março de 2021.

NOGUEIRA, Fábio Fernandes. COSTA, Isabella Almeida. PEREIRA, Uendel Alves. **Análise de parâmetros físico-químicos da água e do uso e ocupação do solo na sub-bacia do Córrego da Água Branca no município de Nerópolis. Goiás, 2015.** Disponível em: <<https://comunicata.ufpi.br/index.php/equador/article/view/9315/5624>>. Acesso em: 21 set. 2021.

PATSKO, Luís Fernando. **TUTORIAL Aplicações, Funcionamento e Utilização de Sensores.** [S.l.], 2006. Disponível em: <https://www.maxwellbohr.com.br/downloads/robotica/mec1000_kdr5000/tutorial_eletronica_-_aplicacoes_e_funcionamento_de_sensores.pdf>. Acesso em: 03 jul. 2021.

PINTO, Magda Cristina Ferreira. **Manual Medição in loco: Temperatura, pH, Condutividade Elétrica e Oxigênio Dissolvido.** Belo Horizonte, 2007. Disponível em: http://www.cprm.gov.br/publique/media/gestao_territorial/geologia_medica/manual_medicoes_T_%20pH_OD.pdf. Acesso em: 29 set. 2021.

RAZA, U.; KULKARNI, P.; SOORIYABANDARA, M. **Low power wide area networks: Na overview.** IEEE Communications Surveys & Tutorials, IEEE, v. 19, n. 2, p. 855–873, 2017.

RIKA. **Rika Sensor**, 2019. Disponível em: <<https://www.rikasensor.com/watersensor.html>>. Acesso em: 18 jun. 2021.

RODRIGUES, Gerlane de Siqueira. DINIZ, Simone Ferreira. **Avaliação Do Monitoramento E Qualidade Da Água Para O Abastecimento Público Da Zona Urbana Do Município De Forquilha.** Ceará, 2019 Disponível em: <<https://comunicata.ufpi.br/index.php/equador/article/view/9315/5624>>. Acesso em: 06 out. 2021.

ROSE, K. **Redes de computador e a internet**. 5. ed. São Paulo: Pearson Education do Brasil, 2010. Disponível em: <<http://webcache.googleusercontent.com/search?q=cache:K5bmdpgI-euEJ:revista.fatecbt.edu.br/index.php/tl/article/download/488/302+&cd=15&hl=pt-BR&ct=clnk&gl=br>>. Acesso em: 18 jun. 2021.

SAAD, A. R.; SEMENSATTO JR, D. L.; AYRES, F. M.; OLIVEIRA, P. E. **Índice de Qualidade da Água – IQA do reservatório do Tanque Grande**, município de Guarulhos, estado de São Paulo, Brasil: 1990 – 2006. Revista UnG – Geociências, v.6, n.1, p.118-133, 2007.

SOUZA FILHO, Elton Alves de. **Diagnóstico da qualidade das águas do igarapé do Mindu e criação de protótipo de aplicativo para disponibilização de dados em Manaus – Am.** Manaus, 2018. Disponível em:<<http://repositorioinstitucional.uea.edu.br/handle/riuea/2890>>. Acesso em: 30 ago. 2021.

THETHINGSNETWORK, T. T. N. **LoRaWAN Overview**. [S.l.], 2019. Disponível em: <<https://www.thethingsnetwork.org/docs/lorawan/architecture/>>. Acesso em 13 out. 2021.

TRATA BRASIL. **Pará, o estado com maior déficit em tratamento de esgoto. Trata Brasil: Saneamento é Saúde. 2019.** Disponível em: <<http://www.tratabrasil.org.br/blog/2019/09/03/para-o-estado-com-maior-deficit-em-tratamento-de-esgoto/>>. Acesso em: 13 de março de 2021.

USINAINFO. **Sensor de PH Arduino + Módulo de Leitura**. [S.l.], 2021a. Disponível em: <<https://www.usinainfo.com.br/outros-sensores-arduino/sensor-de-ph-arduino-modulo-de-leitura-5316.html>>. Acesso em: 04 out. 2021.

USINAINFO. **Sensor de Turbidez Arduino Para Monitoramento da Água - ST100**. [S.l.], 2021b. Disponível em:<<https://www.usinainfo.com.br/outros-sensores-arduino/sensor-deturbidez-arduino-para-monitoramento-da-agua-st100-4539.html>>. Acesso em: 09 out. 2021.

USINAINFO. **ESP32 LORA WIFI SX1278**. [S.l.], 2019. Disponível em: <<https://www.usinainfo.com.br/blog/esp32-lora-wifi-sx1278/>>. Acesso em 23 dez. 2021.

APÊNDICE A - CÓDIGO USADO NO MICROCONTROLADOR MASTER PARA RECEBER A LEITURA DOS SENSORES E ENVIAR PARA A PLATAFORMA WEB.

```

#include "arduino_secrets.h"
//Bibliotecas importadas
#include "thingProperties.h"
#include <heltec.h>
#include <ArduinoJson.h>
//Definição da frequência a ser utilizada (US915)
#define FREQUENCIA 915E6
//Variáveis globais
String mensagem;           // mensagem
byte  enderecoLocal       = 0xFD; // endereço deste dispositivo
byte  enderecoDestino     = 0xBB; // endereço do dispositivo destino
byte  contadorMensagens   = 0;    // contador de mensagens
long  ultimaAtualizacao   = 0;    // última atualização
int   intervaloAtualizacao = 500; // intervalo de envio de mensagens

/**
 * Objetivo: Função responsável por configurar na inicialização
 */
void setup() {
  Heltec.begin(true, true, true, true, FREQUENCIA);
  // Defined in thingProperties.h
  initProperties();
  // Connect to Arduino IoT Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);
  /*
   * The following function allows you to obtain more information
   * related to the state of network and IoT Cloud connection and errors
   * the higher number the more granular information you'll get.
   * The default is 0 (only errors).
   * Maximum is 4
  */

```

```

*/
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();
}

void enviarMensagem(String mensagem) {
  LoRa.beginPacket();      // Inicia o pacote
  LoRa.write(enderecoLocal); // Configura quem está enviando
  LoRa.write(enderecoDestino); // Configura o endereço de destino
  LoRa.write(contadorMensagens); // Configura o id da mensagem
  LoRa.write(mensagem.length()); // Configura o tamanho do payload
  LoRa.print(mensagem);    // Configura o payload
  LoRa.endPacket();        // Finaliza o pacote e envia
  contadorMensagens++;     // Incrementa o id da mensagem
}

void onReceive(int packetSize) {
  //Se o pacote tiver tamanho zero, termina a execução
  if (packetSize == 0) return;      // não teve resposta
  Serial.println("Recebendo dados...");

  //Lê os cabeçalhos dos pacotes
  byte remetente = LoRa.read();      // endereço de quem enviou
  int destinatario = LoRa.read();    // endereço do destinatário
  byte idMensagemChegando = LoRa.read(); // identificador da mensagem
  byte tamanhoMensagemChegando = LoRa.read(); // tamanho da mensagem
  String mensagemResposta = "";

  while (LoRa.available())
    mensagemResposta += (char)LoRa.read();

  if (tamanhoMensagemChegando != mensagemResposta.length()) {
    Serial.println("Erro: O tamanho da mensagem não bate com a mensagem que chegou!");
    return;
  }

  Serial.println("Remetente   : 0x" + String(remetente, HEX));
  Serial.println("Destinatario : 0x" + String(destinatario, HEX));
}

```

```

Serial.println("EnderecoLocal : 0x" + String(enderecoLocal, HEX));
if(destinatario == enderecoLocal){
  //Se a mensagem for para este dispositivo... imprima os detalhes:
  Serial.println("Id da Mensagem: " + String(idMensagemChegando));
  Serial.println("Tamanho da Mensagem: " + String(tamanhoMensagemChegando));
  Serial.println("Mensagem: " + mensagemResposta);
  Serial.println("RSSI: " + String(LoRa.packetRssi()));
  Serial.println("Snr: " + String(LoRa.packetSnr()));
  Serial.println();
  tratarDadosRecebidos(mensagemResposta);
}
}

void tratarDadosRecebidos(String dados) {
  StaticJsonDocument<128> doc;
  DeserializationError error = deserializeJson(doc, dados);
  if (error) {
    Serial.print(F("deserializeJson() falhou: "));
    Serial.println(error.f_str());
    return;
  }
  temperatura = doc["temperatura"];
  tds      = doc["tds"];
  ph      = doc["ph"];
  turbidez = doc["turbidez"];

  localizacao = Location(-3.01591217925088, -60.09362403378129); //Definição de geolocalização do escravo (Praia Dourada)

  Serial.printf("Temperatura: %.2f | TDS: %.2f | PH: %.2f | Turbidez: %.2f\n", temperatura, tds, ph, turbidez);
}

void loop() {
  ArduinoCloud.update();
  if (millis() - ultimaAtualizacao > intervaloAtualizacao) {
    String comando = "lerdados"; //comando para solicitar dados

```

```
enviarMensagem(comando);  
Serial.println("Enviando: " + comando);  
ultimaAtualizacao = millis(); //última vez que atualizou  
intervaloAtualizacao = random(2000) + 1000; // 2-3 seconds  
}  
//Traduz em pacotes e registra o callback:  
onReceive(LoRa.parsePacket());  
}
```

APÊNDICE B - CÓDIGO USADO NO MICROCONTROLADOR SLAVE PARA REALIZAR A LEITURA DOS SENSORES DE TEMPERATURA, PH, TDS E TURBIDEZ.

```

//Bibliotecas importadas
#include <OneWire.h>
#include <DallasTemperature.h>
#include <ArduinoJson.h>
#include "heltec.h"

#define FREQUENCIA      915E6      //Definição da banda/frequência utilizada
(868E6,915E6)

//Definições das constantes
const float  ADC_RES      = 4096.0; // Resolução máxima do pino analógico
const float  VREF         = 3.3;    // Voltagem de referência
const int    TDS_SENSOR   = 12;     // pino do Sensor de tds
//const int   TEMP_SENSOR  = 13;     // pino do Sensor de temperatura
const int    PH_SENSOR    = 34;     // pino do Sensor de pH
const int    TURBIDEZ_SENSOR = 36;  // pino do sensor de turbidez

//Definições das variáveis
String mensagem;                // mensagem
byte  enderecoLocal   = 0xBB;    // endereço deste dispositivo
byte  enderecoDestino = 0xFD;    // endereço do dispositivo destino
byte  contadorMensagens = 0;     // contador de mensagens
long  ultimaAtualizacao = 0;     // última atualização
int   intervaloAtualizacao = 500; // intervalo de envio de mensagens
float temperatura     = 0.0;     // temperatura lida pelo sensor;
float tds              = 0.0;     // tds lida pelo sensor;
float ph              = 0.0;     // ph lido pelo sensor;
float turbidez        = 0.0;     // turbidez lida pelo sensor;
String comando        = "lerdados"; // comando para solicitar dados

//Instanciações das classes
//OneWire oneWire(TEMP_SENSOR);    // Instância do objeto oneWire
//DallasTemperature sensores(&oneWire); // Instância do DallasTemperature

```

```
/**
```

Objetivo: Função responsável por iniciar a configuração dos pinos e outros

Recebe: N/A

Retorna: N/A

```
*/
```

```
void setup() {
```

```
  //Configurando o modo de operação do pino
```

```
  pinMode(TDS_SENSOR, INPUT);    //Configurando o pino do sensor TDS para modo
  entrada
```

```
  pinMode(PH_SENSOR, INPUT);     //Configurando o pino do sensor PH para modo en-
  trada
```

```
  pinMode(TURBIDEZ_SENSOR, INPUT); //Configurando o pino do sensor PH para
  modo entrada
```

```
  // sensores.begin();           //Inicializando os sensores de temperatura
```

```
  Heltec.begin(true, true, true, true, FREQUENCIA);
```

```
  Serial.println("Projeto Agua Rio inicializado...");
```

```
}
```

```
/**
```

Objetivo: Função responsável por disparar a leitura de temperatura

Recebe: N/A

Retorna: Temperatura em graus celsius

```
*/
```

```
float lerTemperatura() {
```

```
  float temp = 0.0;
```

```
  // sensores.requestTemperatures();    // Envia o comando para ler a temperatura
```

```
  // temp = sensores.getTempCByIndex(0); // Envia o comando para ler em graus celsius
```

```
  Serial.printf("Temperatura: %.2f\n", temp);
```

```
  delay(1000);
```

```
  return temp;
```

```
}
```

```
/**
```

Objetivo: Função responsável por disparar a leitura da tds

Recebe: N/A


```

Retorna: O valor da tds
*/
float lerTds() {
    int SCOUNT = 30;           // Contador de amostras
    int analogBuffer[SCOUNT]; // Armazena o valor analógico lido do ADC em um array
    int analogBufferTemp[SCOUNT];
    int analogBufferIndex = 0;
    int copyIndex = 0;
    float averageVoltage = 0;
    float tdsValue = 0;
    float temperature = 25;
    static unsigned long analogSampleTimepoint = millis();
    if (millis() - analogSampleTimepoint > 40U) { //A cada 40 milisegundos, leia o valor analógico vindo do ADC
        analogSampleTimepoint = millis();
        analogBuffer[analogBufferIndex] = analogRead(TDS_SENSOR); //Lendo o valor analógico e armazenando no buffer
        analogBufferIndex++;
        if (analogBufferIndex == SCOUNT)
            analogBufferIndex = 0;
    }
    static unsigned long printTimepoint = millis();
    if (millis() - printTimepoint > 800U) {
        printTimepoint = millis();
        for (copyIndex = 0; copyIndex < SCOUNT; copyIndex++)
            analogBufferTemp[copyIndex] = analogBuffer[copyIndex];
        averageVoltage = getMedianNum(analogBufferTemp, SCOUNT) * (float)VREF / ADC_RES; // read the analog value more stable by the median filtering algorithm, and convert to voltage value
        float compensationCoefficient = 1.0 + 0.02 * (temperature - 25.0); //temperature compensation formula: fFinalResult(25 ^ C) = fFinalResult(current) / (1.0 + 0.02 * (fTP - 25.0));
        float compensationVoltage = averageVoltage / compensationCoefficient; // temperature compensation
        tdsValue = (133.42 * compensationVoltage * compensationVoltage *

```

```

        compensationVoltage - 255.86 * compensationVoltage * compensationVoltage +
        857.39 * compensationVoltage) * 0.5; //convert voltage value to tds value
    Serial.printf("TDS: %.2fppm\n", tdsValue);
}
return tdsValue;
}
int getMedianNum(int bArray[], int iFilterLen)
{
    int bTab[iFilterLen];
    for (byte 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;
}

```

```
/**
```

Objetivo: Função responsável por facilitar a calibração do sensor de pH

Recebe: N/A

Retorna: N/A

```
*/
```

```
void calibrarSensorPH() {
    float pH = analogRead(PH_SENSOR);
    float voltagem = pH * (VREF / ADC_RES);
    Serial.printf("ph: %.2f | Voltagem: %.2f\n", pH, voltagem);
    delay(500);
}
```

```
/**
```

Objetivo: Função responsável por calcular o pH

Recebe: a voltagem lida

Retorna: o valor calculado do pH

```
*/
```

```
float calcularPH (float voltagem) {
    return 7 + ((VREF - voltagem) / 0.18);
}
```

```
/**
```

Objetivo: Função responsável por disparar a leitura do sensor de pH

Recebe: N/A

Retorna: o valor do pH

```
*/
```

```
float lerPH() {
    int amostras = 40;
    float leituras = 0.0;
    float leitura = 0.0;
    for (int i = 0; i < amostras; i++) {
        leitura = analogRead(PH_SENSOR);
        leituras += leitura;
        delay(10);
    }
}
```

```

}
float voltagem = VREF / ADC_RES * leituras / amostras;
float pH = calcularPH(voltagem);
Serial.printf("Voltagem: %.2f | pH: %.2f\n", voltagem, pH);
delay(1000);
return pH;
}
/**
  Objetivo: Função responsável por disparar a leitura do sensor de turbidez
  Recebe: N/A
  Retorna: o valor da turbidez em NTU
*/
float lerTurbidez(){
  int valorLido = analogRead(TURBIDEZ_SENSOR);
  float voltagem = map(valorLido, 0, 4095, 0, 3.3);
  //Para 5V utilizar:
  //float NTU = -1120.4 * sq(voltagem) + 5742.3 * voltagem - 4352.9;
  //Para 3.3V utilizar:
  float NTU = -2572.2 * sq(voltagem) + 8700.5 * voltagem - 4352.9;
  if(NTU < 0)
    NTU = 0;
  Serial.printf("ADC: %d | Voltagem: %2.1f | NTU: %3f\n", valorLido, voltagem, NTU);
  delay(500);
  return NTU;
}
/**
  Objetivo: Função responsável por disparar a leitura de todos os sensores e
  armazenar os resultados em variáveis
  Recebe: N/A
  Retorna: N/A
*/
String lerSensores() {

```

```

//Chama a rotina de leitura de temperatura
temperatura = lerTemperatura();
//Chama a rotina de leitura de tds
tds = lerTds();
//Chama a rotina de leitura de pH
ph = lerPH();
//Chama a rotina de leitura da turbidez
turbidez = lerTurbidez();
//Instanciando um objeto json
StaticJsonDocument<64> doc;
//Populando os dados dentro do json
doc["temperatura"] = temperatura;
doc["tds"] = tds;
doc["ph"] = ph;
doc["turbidez"] = turbidez;
String dados = "";
serializeJson(doc, dados);
return dados;
}
/**
  Objetivo: Função responsável por enviar as mensagens através do LoRa
  Recebe: a mensagem a ser enviada
  Retorna: N/A
*/
void enviarMensagem(String mensagem) {
  Serial.println("Enviando dados...");
  Serial.println("Dados: " + mensagem);
  LoRa.beginPacket(); // Inicia o pacote
  LoRa.write(enderecoLocal); // Configura quem está enviando
  LoRa.write(enderecoDestino); // Configura o endereço de destino
  LoRa.write(contadorMensagens); // Configura o id da mensagem
  LoRa.write(mensagem.length()); // Configura o tamanho do payload

```

```

LoRa.print(mensagem);    // Configura o payload
LoRa.endPacket();        // Finaliza o pacote e envia
contadorMensagens++;     // Incrementa o id da mensagem
}
/**
  Objetivo: Função principal
  Recebe: N/A
  Retorna: N/A
*/
void loop() {
  if (millis() - ultimaAtualizacao > intervaloAtualizacao) {
    enviarMensagem(comando);
    Serial.println("Enviando: " + comando);
    ultimaAtualizacao = millis(); //última vez que atualizou
    intervaloAtualizacao = random(2000) + 1000; // 2-3 seconds
  }
  // parse for a packet, and call onReceive with the result:
  onReceive(LoRa.parsePacket());
}
/**
  Objetivo: Função responsável por ler os eventos de respostas recebidas do LoRa
  Recebe: o tamanho do pacote recebido
  Retorna: N/A
*/
void onReceive(int packetSize) {
  //Se o pacote tiver tamanho zero, termina a execução
  if (packetSize == 0) return; // não teve resposta
  Serial.println("Recebendo dados...");
  //Lê os cabeçalhos dos pacotes
  byte remetente = LoRa.read(); // endereço de quem enviou
  int destinatario = LoRa.read(); // endereço do destinatário
  byte idMensagemChegando = LoRa.read(); // identificador da mensagem

```

```

byte tamanhoMensagemChegando = LoRa.read();// tamanho da mensagem
String mensagemResposta = "";
while (LoRa.available())
    mensagemResposta += (char)LoRa.read();
if (tamanhoMensagemChegando != mensagemResposta.length()) {
    Serial.println("Erro: O tamanho da mensagem não bate com a mensagem que chegou!");
    return;
}
Serial.println("Remetente   : 0x" + String(remetente, HEX));
Serial.println("Destinatario  : 0x" + String(destinatario, HEX));
Serial.println("EnderecoLocal : 0x" + String(enderecoLocal, HEX));
if (destinatario == enderecoLocal){
    //Se a mensagem for para este dispositivo... imprima os detalhes:
    Serial.println("Id da Mensagem: " + String(idMensagemChegando));
    Serial.println("Tamanho da Mensagem: " + String(tamanhoMensagemChegando));
    Serial.println("Mensagem: " + mensagemResposta);
    Serial.println("RSSI: " + String(LoRa.packetRssi()));
    Serial.println("Snr: " + String(LoRa.packetSnr()));
    Serial.println();
    tratarResposta(mensagemResposta);
}
}
/**
Objetivo: Função responsável por tratar as respostas recebidas
Recebe: a resposta
Retorna: N/A
*/
void tratarResposta(String mensagemResposta) {
    Serial.println("Tratando resposta...");
    if (mensagemResposta.equals("lerdados")) {
        Serial.println("Recebi um comando solicitando atualização da leitura de dados...");
        comando = lerSensores();}
}

```