

**Universidade do Estado Amazonas
Escola Superior de Tecnologia - EST**

Benilton Seixas Andrade

**DESENVOLVIMENTO DE UM SOFTWARE EMBARCADO
SUPERVISÓRIO DE RECURSOS HÍDRICOS PARA MONITORAR EM
TEMPO REAL A QUALIDADE DOS RECURSOS HÍDRICOS NA
PISCICULTURA.**

Manaus

2022

BENILTON SEIXAS ANDRADE

**DESENVOLVIMENTO DE UM SOFTWARE EMBARCADO
SUPERVISÓRIO DE RECURSOS HÍDRICOS PARA MONITORAR EM
TEMPO REAL A QUALIDADE DOS RECURSOS
HÍDRICOS NA PISCICULTURA.**

Pesquisa desenvolvida durante a disciplina de Trabalho de Conclusão de Curso II e apresentado 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.

Orientação: Dr. Israel Gondres Torné

Manaus

2022

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

Reitor:

André Luiz Nunes Zogahib

Vice-Reitor:

Kátia do Nascimento Couceiro

Diretora da Escola Superior de Tecnologia:

Ingrid Sammyne Gadelha Figueiredo

Coordenador do Curso de Engenharia Eletrônica:

Bruno da Gama Monteiro

Banca Avaliadora composta por: *Data da defesa: 26/10/2022.*

Prof. Dr. Israel Gondres Torné (Orientador)

Prof. Dr. Luis Cuevas Rodriguez

Prof. Dr. Miguel A. Orellana

CIP – Catalogação na Publicação

Andrade, Benilton Seixas

Desenvolvimento de um software embarcado supervisorio de recursos hídricos para monitorar em tempo real a qualidade dos recursos hídricos na Piscicultura / Benilton Seixas Andrade; [orientado por] Israel Gondres Torné. – Manaus: 2022.

57 p.: il.

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

1. Piscicultura. 2. software embarcado. 3. parâmetro hídrico. I. Torné, Israel Gondres.

BENILTON SEIXAS ANDRADE

**DESENVOLVIMENTO DE UM SOFTWARE EMBARCADO
SUPERVISÓRIO DE RECURSOS HÍDRICOS PARA MONITORAR EM
TEMPO REAL A QUALIDADE DOS RECURSOS
HÍDRICOS NA PISCICULTURA.**

Pesquisa desenvolvida durante a disciplina de Trabalho de Conclusão de Curso II e apresentado 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: **8,61 (Oito vírgula sessenta e um)**

Aprovado em: 26 /10/2022

Área de Concentração: Software Embarcado.

BANCA EXAMINADORA



Orientador: Israel Gondres Torné, Dr.



Avaliador: Luis Cuevas Rodriguez, Dr.



Avaliador: Miguel A. Orellana, Dr.

Manaus

2022

DEDICATÓRIA

A minha avó, que cuidou de mim desde a infância e graças ao incentivo dela eu posso finalizar essa etapa da minha vida. A minha companheira, Thayná Almeida, que sempre esteve comigo nos mais diversos momentos. Aos meus professores que me guiaram nesta jornada. E aos meus amigos que me ajudaram direta ou indiretamente. O meu mais sincero agradecimento.

RESUMO

Este trabalho apresenta o desenvolvimento de um *software* embarcado supervisor de recursos hídricos em tempo real na piscicultura. Com o auxílio de sensores, serão monitorados os seguintes parâmetros hídricos: potencial hidrogeniônico (pH), temperatura e turbidez da água. Os dados aferidos serão disponibilizados em uma interface gráfica para visualização. Para o desenvolvimento da aplicação foi adotado e implementado um estilo de código claro, legível e restrições razoáveis. Além disso, o código fonte será disponibilizado na comunidade do *github*. Por fim, os dados obtidos mostraram que o *software* embarcado desenvolvido é capaz de monitorar a qualidade dos recursos hídricos no contexto da piscicultura.

Palavras Chaves: Piscicultura, software embarcado, parâmetro hídrico

ABSTRACT

This work presents the development of an embedded supervisory software for water resources in real time in fish farming. With the help of sensors, the following water parameters will be monitored: hydrogenion potential (pH), temperature and turbidity of the water. The measured data will be made available in a graphical interface for visualization. For the development of the application, a clear, readable and accessible code style was adopted and implemented. Furthermore, the source code will be made available on the github community. Then, the data acquired from fish farming the embedded software is able to give a context of water resources in the context of pisciculture.

Keywords: Pisciculture, embedded software, water parameter.

SUMÁRIO

INTRODUÇÃO.....	10
1 REFERENCIAL TEÓRICO.....	12
1.1 CONTEXTUALIZAÇÃO.....	12
1.2 PARÂMETROS HÍDRICOS.....	12
1.3 TEMPERATURA.....	13
1.4 POTENCIAL HIDROGENIÔNICO (PH).....	15
1.5 TURBIDEZ.....	16
1.6 INTERFACE HOMEM MÁQUINA.....	17
2 ARQUITETURA DO SISTEMA.....	18
2.1 FLUXOGRAMA DO SOFTWARE EMBARCADO.....	19
3 LEVANTAMENTO DE REQUISITOS.....	20
4 MATERIAIS E MÉTODOS.....	22
4.1 SENSORES.....	22
4.2 IDE.....	25
4.3 VISUAL STUDIO CODE.....	25
4.4 PREPARAÇÃO DO AMBIENTE DE DESENVOLVIMENTO.....	26
4.5 FREERTOS.....	27
4.6 MICROCONTROLADORES.....	29
4.7 MICROCONTROLADOR ESP32.....	29
5 BOAS PRÁTICAS DE DESENVOLVIMENTO.....	30
5.1 CONTROLE DE VERSÃO.....	30
5.2 SISTEMA DE CONTROLE DE VERSÃO DISTRIBUÍDO.....	31
5.3 REPOSITÓRIO REMOTO.....	33
6 TESTES EM LABORATÓRIO.....	34
7 RESULTADOS OBTIDOS.....	37
7.1 ESTRUTURA DE PASTAS DO PROJETO.....	37
7.2 PADRONIZAÇÃO DO CÓDIGO DO PROJETO.....	38
7.3 HISTÓRICO DE ALTERAÇÕES DO PROJETO.....	40
7.4 AFERIÇÃO E COMPARAÇÃO DOS DADOS.....	41
CONCLUSÃO.....	52
REFERÊNCIAS.....	54

Lista de Figuras

Figura 1 - Escala Celsius.....	13
Figura 2 - Escala de Potencial Hidrogeniônico (pH).....	15
Figura 3 - Exemplo de níveis de turbidez da água.....	16
Figura 4 - Exemplos de interfaces gráficas homem máquina.....	17
Figura 5 - Modelagem do sistema.....	18
Figura 6 - Fluxograma do software embarcado.....	19
Figura 7 - Sensor de pH constituído pelo módulo PH4202C e eletrodo.....	22
Figura 8 - Sensor turbidez.....	23
Figura 9 - Sensor de temperatura.....	24
Figura 10 - Plataforma disponível para <i>download</i>	25
Figura 11 - Instalação do PlatformIO.....	26
Figura 12 - Criação de um novo projeto.....	27
Figura 13 - Estados que uma tarefa pode assumir.....	28
Figura 14 - Diagrama de blocos funcionais.....	29
Figura 15 - Diagrama de controle de versão.....	30
Figura 16 - Histórico de alterações do projeto.....	31
Figura 17 - Identificação de um commit de uma funcionalidade.....	32
Figura 18 - Identificação de um “commit” de correção.....	32
Figura 19 - Identificação de um “commit” de refatoração.....	32
Figura 20- Equipamento HI98196.....	34
Figura 21- Equipamento HI98703.....	35
Figura 22 - Etapas de testes em laboratório.....	36
Figura 23 - Estruturas de pastas do projeto.....	37
Figura 24 - Padrão de função.....	38
Figura 25 - Padrão de constantes.....	39
Figura 26 - Único ponto de retorno.....	39
Figura 27 - Histórico de alterações do projeto.....	40
Figura 28 - Aferição da turbidez da água mineral com o equipamento HI98703.....	41
Figura 29 - Aferição da primeira amostra de turbidez da água por meio do software.....	42

Figura 30 - Aferição da turbidez da segunda amostra com o equipamento HI98703.....	43
Figura 31 - Aferição da segunda amostra de turbidez da água por meio do software.....	43
Figura 32 - Aferição da turbidez da água por meio do software.....	44
Figura 33 - Aferição da turbidez da água com leite em pó.....	45
Figura 34 - Medição da temperatura da amostra de água mineral com leite em pó.....	46
Figura 35 - Medição da temperatura da água mineral aquecida no microondas.....	47
Figura 36 - Medição da temperatura da amostra de água mineral.....	48
Figura 37 - Comparação do pH da água mineral.....	49
Figura 38 - Comparação do pH da água mineral do filtro Acqua Bella.....	50

Lista de Tabelas

Tabela 1 - Faixa de temperatura ideal para diferentes espécies de peixes.....	14
Tabela 2 - Requisitos funcionais do <i>software</i> embarcado.....	20
Tabela 3 - Requisitos não funcionais do <i>software</i> embarcado.....	21
Tabela 4 - Comparação dos parâmetros aferidos.....	51

INTRODUÇÃO

O desenvolvimento sustentável é uma tendência no mundo moderno, em que existe o desafio de preservar os recursos naturais e satisfazer as necessidades humanas. Neste contexto, as instituições de ensino superior (IES) desempenham um papel relevante em: pesquisar e desenvolver práticas sustentáveis, reduzir o desperdício de recursos escassos e servir como referência para a sociedade.

O desenvolvimento de novas tecnologias proporcionou o controle gestão de água, por meio da utilização de sistemas sensoriais de parâmetros hídricos e com microcontroladores, como o ESP32. Para Silva; Carvalho; Cardoso (2019) o reuso da água é um ponto fundamental para o controle de água potável e melhor gerenciamento dos recursos hídricos.

O Brasil, no contexto mundial, possui cerca de 12% das reservas de água doce do planeta, além de possuir a maior bacia hidrográfica, em termos de volume, a bacia do rio Amazonas (CANTELLE; LIMA; BOGES, 2018).

Por conta da riqueza de recursos hídricos do país, a piscicultura é uma atividade econômica com forte capacidade de crescimento no Amazonas. No entanto, há muitas barreiras que inviabilizam essa atividade econômica, como: assistência técnica deficiente, burocracia ambiental, dificuldade de acesso ao crédito, pouco investimento em pesquisa, escassez de recursos tecnológicos. (CARVALHO; JUNIOR; OLIVEIRA, 2020).

Este trabalho tem como hipótese a ideia de que é possível monitorar diferentes parâmetros hídricos, por meio de um *software* embarcado para monitorar, em tempo real, a qualidade dos recursos hídricos na piscicultura. Além disso, é possível usar um sistema de controle de versão distribuído para preservar as modificações no *software*, aplicar boas práticas de desenvolvimento e embarcar o *software* no microcontrolador Esp32.

O objetivo deste trabalho é desenvolver um *software* embarcado para supervisionar parâmetros hídricos em tempo real na piscicultura. Tendo como objetivos específicos: (I) realizar análise e levantamento dos requisitos do *software* embarcado; (II) versionar os requisitos do *software* embarcado; (III) implementar um estilo de código para facilitar o desenvolvimento; (IV) embarcar *software* no microcontrolador Esp32; (V) validar dados aferidos com o *software* embarcado.

A justificativa para implementação deste trabalho é que o desenvolvimento do *software* embarcado permitirá o controle de qualidade de água no contexto da piscicultura. Tornando-se viável o monitoramento e gerência dos recursos hídricos para os produtores no estado do Amazonas.

Para implementação deste trabalho, foram utilizados vários conceitos estudados no decorrer acadêmico do curso de Engenharia Eletrônica, principalmente nas disciplinas: Técnicas de Programação em Engenharia Eletrônica, Ciências Ambientais para Engenharia Eletrônica, Sistemas Operacionais Embarcados, Microprocessadores e Microcontroladores.

Esta pesquisa tem como objetivo final, o desenvolvimento de um *software* embarcado capaz de aferir qualidade dos recursos hídricos na piscicultura, com uso de sensores de parâmetros hídricos associados ao microcontrolador Esp32. Além de boas práticas de desenvolvimento e técnicas de programação.

Este trabalho está dividido em: referencial teórico, arquitetura do sistema, levantamento de requisitos, materiais e métodos, boas práticas de desenvolvimento, testes em laboratório e resultados obtidos.

Por fim, na conclusão, será retomado o que foi proposto como hipótese em comparação com os resultados obtidos.

1 REFERENCIAL TEÓRICO

Neste capítulo, serão apresentados os aspectos teóricos relativos ao trabalho acadêmico. A princípio, será efetuada uma contextualização quanto à sustentabilidade bem como o uso de recursos hídricos.

1.1 CONTEXTUALIZAÇÃO

A partir de 1950 uma parte da população global iniciou um debate, questionando o modelo de sociedade, baseado no desenvolvimento nocivo ao meio ambiente. Como resultado, em 1972, o meio ambiente foi tema mundial na primeira conferência de Estocolmo da Assembleia Geral da Organização das Nações Unidas (ONU) (BARBOSA; DRACH; CORBELLA, 2014).

Enquanto que no Brasil, a gestão dos recursos hídricos é regulamentada pela Política de Recursos Hídricos, Lei nº 9.433/1997, onde estão presentes conceitos, fundamentos, diretrizes e instrumentos de gestão. Os principais objetivos políticos são: (I) assegurar à atual e às futuras gerações a necessária disponibilidade de água, em padrões de qualidade adequados aos respectivos usos; (II) a utilização racional e integrada dos recursos hídricos, incluindo o transporte aquaviário, com vistas ao desenvolvimento sustentável; (III) a prevenção e a defesa contra eventos hidrológicos críticos de origem natural ou decorrentes do uso inadequado dos recursos naturais.

No Estado do Amazonas estima-se que mais de 200 mil pessoas estão envolvidas com a produção do pescado e seu subproduto, desde a captura, o manuseio, a distribuição nos pontos de venda até o consumidor final (IDAM, 2020). Sendo assim, a produção de peixes é uma atividade econômica bastante promissora no Amazonas e é necessário preservar a riqueza dos recursos hídricos.

1.2 PARÂMETROS HÍDRICOS

O ambiente em que os peixes estão inseridos varia ao longo do dia, podendo prejudicar e até mesmo inviabilizar a produção do pescado. Sendo assim, a gestão de grandezas físico-químicas são fundamentais para manter uma boa qualidade da água.

1.3 TEMPERATURA

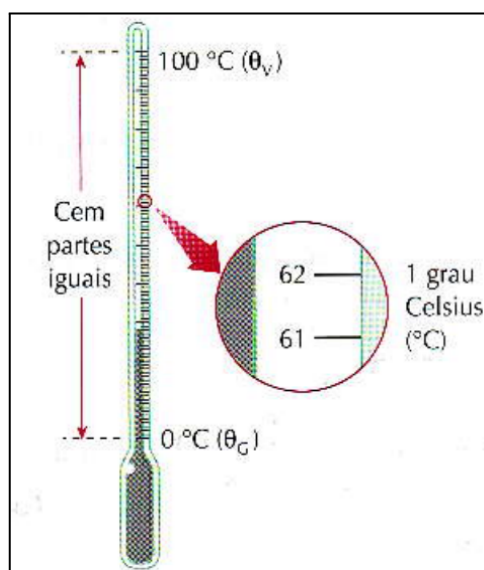
Acerca da temperatura é correto afirmar que:

É a medida da intensidade de calor expresso em uma determinada escala. Uma das escalas mais usadas é grau centígrado ou grau Celsius ($^{\circ}\text{C}$). A temperatura pode ser medida por diferentes dispositivos, como, por exemplo, termômetro ou sensor. (PINTO, 2007, p. 1).

A escala utilizada no Sistema Internacional de Medidas (SI) para quantificar a temperatura é Kelvin (K). Em alguns países é usada a escala Fahrenheit, que adota os valores de 32 para o ponto de fusão da água e 212 para o ponto de ebulição da água, a escala Fahrenheit é dividida em 180 partes, cada uma das quais corresponde ao grau Fahrenheit, cujo símbolo é $^{\circ}\text{F}$ (JUNIOR; FERRARO; SOARES, 2007).

No entanto, a escala mais usada é a Celsius, que adota os valores de 0 e 100 para os pontos de fusão e ebulição da água, respectivamente. O intervalo entre os pontos fixos é dividido em cem partes. Cada uma dessas cem partes é a unidade da escala, o grau Celsius, cujo símbolo é $^{\circ}\text{C}$, como mostra a figura 1 (JUNIOR; FERRARO; SOARES, 2007).

Figura 1 - Escala Celsius.



Fonte: (JUNIOR; FERRARO; SOARES, 2007, p. 11).

Os peixes são animais de sangue frio, chamados pecilotérmicos. Essa peculiaridade significa uma melhor eficiência alimentar, uma vez que a energia dos alimentos ingeridos é utilizada para seu crescimento e não para a preservação da temperatura corporal como é no caso dos mamíferos, por exemplo (LOPES, 2012).

“A temperatura influencia diretamente nas atividades fisiológicas dos peixes. Cada espécie possui uma zona térmica ótima, ou seja, uma variação de temperatura confortável que possibilita seu desenvolvimento” (LOPES, 2007, p. 45).

Os peixes tropicais referem-se àqueles que apresentam o maior potencial de desenvolvimento, quando a temperatura da água se encontra entre 22°C e 30°C, conforme a tabela 1 (PEZZATO; CASTAGNOLLI; ROSSI, 2001, p. 36).

Tabela 1 - Faixa de temperatura ideal para diferentes espécies de peixes.

Espécie	Temperatura (°C)
Pacu	25 a 30
Tambaqui	26 a 28
Tambacu	25 a 30
Carpa Comum	24 a 30
Carpa Capim	24 a 27
Tilápia	26 a 28
Bagre Africano	24 a 28
Pintado	22 a 28
Dourado	20 a 28

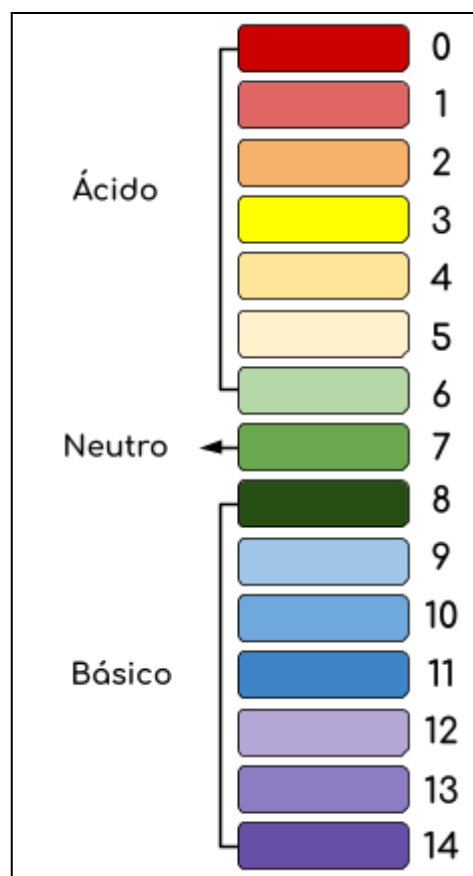
Fonte: (LOPES, 2007, p. 46).

1.4 POTENCIAL HIDROGENIÔNICO (PH)

Sobre o Potencial Hidrogeniônico, como demonstrado na figura 2, é correto afirmar que:

O pH é uma medida da intensidade do caráter ácido de uma solução. É dado pela atividade do íon hidrogênio (H^+), sendo medido potenciométricamente e apresentado em uma escala anti-logarítmica. A escala de pH, compreendida entre 0 e 14, indica se o meio é ácido, básico ou neutro, quando o pH for menor, maior ou igual a 7, respectivamente. O pH é uma propriedade expressa unidimensionalmente, ou seja, sem unidade. (PINTO, 2007, p. 3).

Figura 2 - Escala de Potencial Hidrogeniônico (pH)



Fonte: Autoria Própria

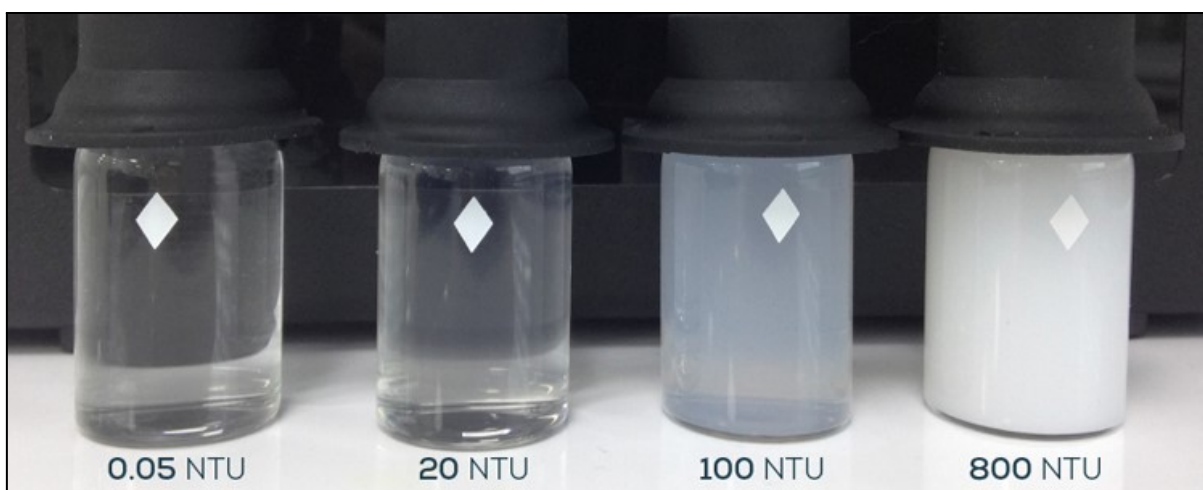
Segundo Lopes (2012) variações abruptas no pH da água podem gerar mortalidade em massa dos peixes e exposições por longo tempo comprometem o desenvolvimento dos mesmos. A faixa de pH para a piscicultura está entre 6,0 e 9,0, sendo ideal entre 7,0 e 8,5.

1.5 TURBIDEZ

Como demonstra a figura 3 e de acordo com Correia et al. (2008, p. 2):

A medida da dificuldade de um feixe de luz atravessar certa quantidade de água é chamada de Turbidez. Que é causada por matérias sólidas em suspensão (silte, argila, colóides, matéria orgânica, etc.). A turbidez é medida através do turbidímetro, comparando-se o espalhamento de um feixe de luz ao passar pela amostra com o espalhamento de um feixe de igual intensidade ao passar por uma suspensão padrão. Quanto maior o espalhamento maior será a turbidez. Os valores 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.

Figura 3 - Exemplo de níveis de turbidez da água.



Fonte: (Envicon química e meio ambiente, 2020)

1.6 INTERFACE HOMEM MÁQUINA

Segundo OLIVEIRA (2015, p. 11):

"A interface é a única maneira que o usuário tem de avaliar o sistema. A ele não interessa a linguagem de programação na qual o sistema foi desenvolvido, o tipo de equipamento no qual a aplicação foi desenvolvida, nem a metodologia empregada na sua concepção e desenvolvimento"

No contexto da pesquisa, a interface tem a função de mostrar para o usuário final as grandezas aferidas, a figura 4 mostra algumas interfaces para sistemas embarcados. A interface escolhida é o *display* Oled, por conta das dimensões reduzidas.

Figura 4 - Exemplos de interfaces gráficas homem máquina

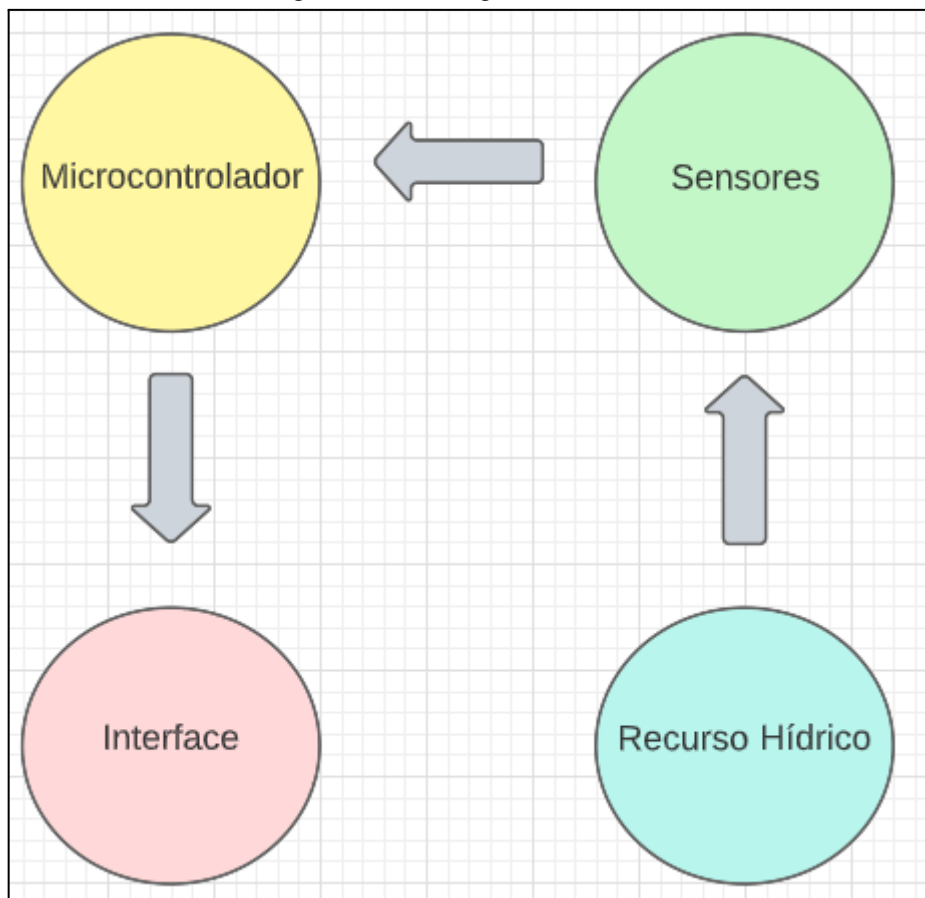


Fonte: Autoria Própria

2 ARQUITETURA DO SISTEMA

Nesta etapa, serão realizadas as tarefas de modelagem do sistema, que consiste no processo de elaboração de um modelo que represente de forma abstrata o sistema a ser desenvolvido, a figura 5. A modelagem tem como função: (I) identificar as principais partes do sistema; (II) auxiliar na visualização do sistema; (III) entender como as diferentes partes do sistema podem interagir entre si.

Figura 5 - Modelagem do sistema.

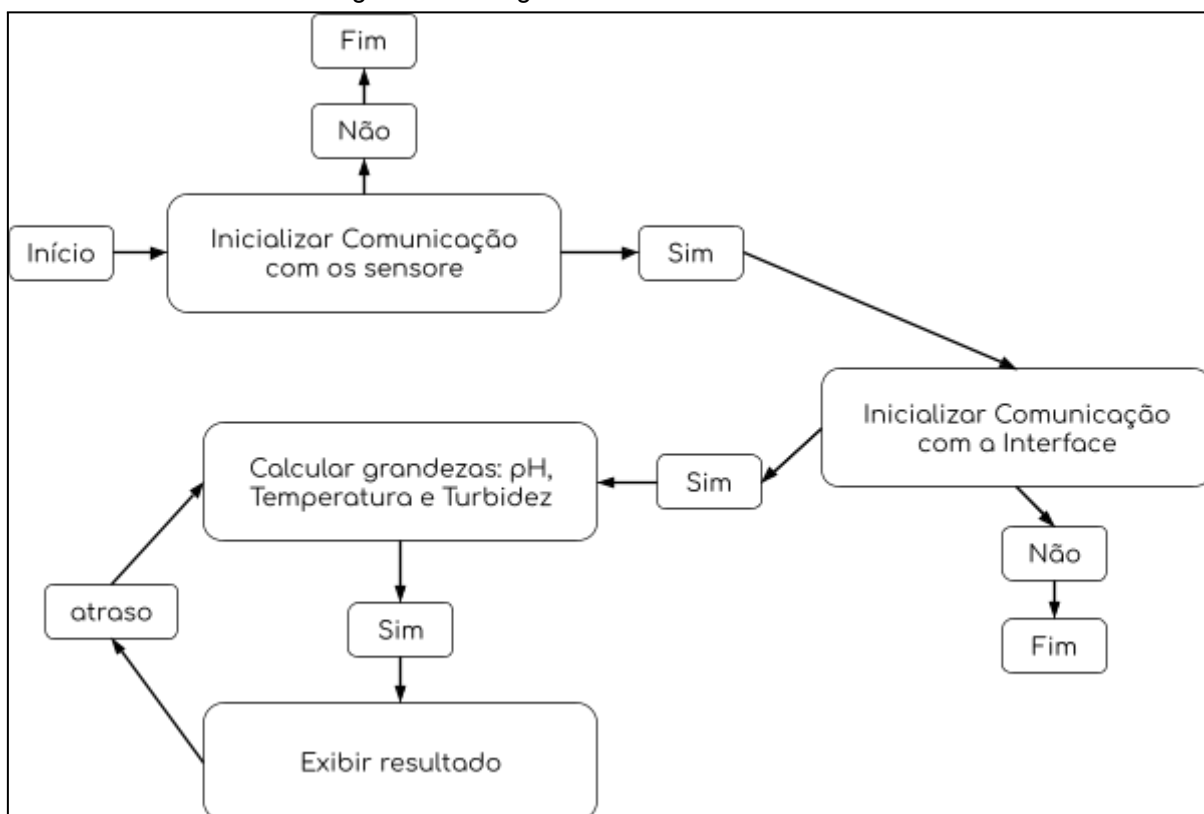


Fonte: Autoria Própria

2.1 FLUXOGRAMA DO SOFTWARE EMBARCADO

A figura 6 descreve, em alto nível, o fluxo de funcionamento do *software* embarcado. O objetivo deste fluxograma é descrever as principais etapas da aplicação desenvolvida.

Figura 6 - Fluxograma do *software* embarcado.



Fonte: Autoria Própria

Portanto, assim que a aplicação é colocada em execução, ela inicia a comunicação com os sensores. Em seguida, inicializa a comunicação com uma interface gráfica. Caso ocorra algum erro nas etapas supracitadas, a aplicação é encerrada.

Se a comunicação inicial dos sensores e da IHM forem bem sucedidos, o programa inicia o *loop* principal. Este *loop* terá as funções responsáveis por calcular os parâmetros hídricos de: ph, temperatura e turbidez Além disso, irá exibir os dados calculados na IHM por período de tempo limitado.

3 LEVANTAMENTO DE REQUISITOS

É a primeira fase no ciclo de desenvolvimento, onde são definidas as funcionalidades e escopo do projeto. Sendo assim, esta etapa serve para levantar as necessidades antes do desenvolvimento (CARVALHO, 2009).

Essa etapa deve ser analisada cuidadosamente, uma vez que impacta diretamente o desenvolvimento do *software* embarcado, podendo até mesmo inviabilizá-lo.

Esta fase serve para delimitar não só quais parâmetros hídricos podem ser monitorados, mas também as restrições do software e tarefas aplicáveis ao desenvolvimento do *software* embarcado.

Requisitos funcionais: define a funcionalidade de um sistema, conforme mostra a tabela 2.

Tabela 2 - Requisitos funcionais do *software* embarcado.

Identificador	Descrição
RF01	O <i>software</i> embarcado deve ler os sinais elétricos do sensor de temperatura.
RF02	O <i>software</i> embarcado deve ler os sinais elétricos do sensor pH.
RF03	O <i>software</i> embarcado deve ler os sinais elétricos do sensor de turbidez
RF05	O software embarcado deve inicializar a comunicação com os sensores.
RF06	O <i>software</i> embarcado deve inicializar a comunicação com uma IHM.
RF07	O <i>software</i> embarcado deve ser capaz de calcular a temperatura da água.
RF08	O <i>software</i> embarcado deve ser capaz de calcular o nível de turbidez da água.
RF09	O <i>software</i> embarcado deve ser capaz de calcular o nível de pH da água.
RF10	O <i>software</i> embarcado deve exibir a temperatura na IHM
RF11	O <i>software</i> embarcado deve exibir o nível de turbidez na IHM.
RF12	O <i>software</i> embarcado deve ser capaz de exibir o nível pH na IHM.

Fonte: Autoria Própria

Requisitos não funcionais: são aqueles que estão relacionados ao desempenho, usabilidade, confiabilidade, segurança, disponibilidade, manutenção e tecnologias envolvidas, conforme a tabela 3.

Tabela 3 - Requisitos não funcionais do *software* embarcado.

Identificador	Descrição
RNF01	O <i>software</i> embarcado deve seguir uma padronização.
RNF02	O <i>software</i> embarcado deve estar em um único idioma.
RNF03	O <i>software</i> embarcado deve ser resiliente a erros.
RNF04	O <i>software</i> embarcado deve estar preparado para adição de novas funcionalidades.
RNF05	O <i>software</i> embarcado deve ser de fácil compartilhamento.
RNF06	O <i>software</i> embarcado deve ser desenvolvido com tecnologias <i>open-source</i>
RNF07	O <i>software</i> embarcado deve exibir as unidades de medidas de cada grandeza na interface
RNF08	O <i>software</i> embarcado deve atingir um nível satisfatoriamente alto de desempenho, a fim de gerenciar a qualidade do recurso hídrico em tempo real.
RNF09	O <i>software</i> embarcado deve atualizar as informações de temperatura, turbidez e pH em um determinado período de tempo.
RNF10	O <i>software</i> embarcado deve ter uma estrutura de pastas de fácil entendimento.

Fonte: Autoria Própria

4 MATERIAIS E MÉTODOS

4.1 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 divididos em sensores analógicos e sensores digitais. Os sensores digitais são baseados em níveis de tensão bem definidos, que por sua vez podem ser caracterizados em nível lógico alto ou nível lógico baixo. Isto significa, que os sensores digitais operam com lógica binária.

“Ao contrário de um sensor analógico, onde os valores possíveis são teoricamente infinitos, um sensor digital poderá apenas alternar entre certos estados bem definidos, não sendo possível haver um valor intermediário entre eles” (PATSKO, 2006, p. 2). Portanto, os sensores selecionados para o trabalho são:

- 1) Sensor digital de pH, figura 7:
 - 1.1) Modelo: PH4502C;
 - 1.2) Alimentação: 5.00V DC;
 - 1.3) Faixa de medição: 0~14pH;
 - 1.4) Acurácia: ± 0.1 pH;
 - 1.5) Tempo de resposta: ≤ 1 min;
 - 1.6) Sinal: analógico;

Figura 7 - Sensor de pH constituído pelo módulo PH4202C e eletrodo.



Fonte: (SARAVATI, 2021c, p. 01).

2) Sensor digital de Turbidez, figura 8:

2.1) Alimentação: 5V DC;

2.2) Corrente de trabalho: máximo de 40 mA;

2.3) Sinal: analógico;

2.4) Tempo de resposta: < 500 ms;

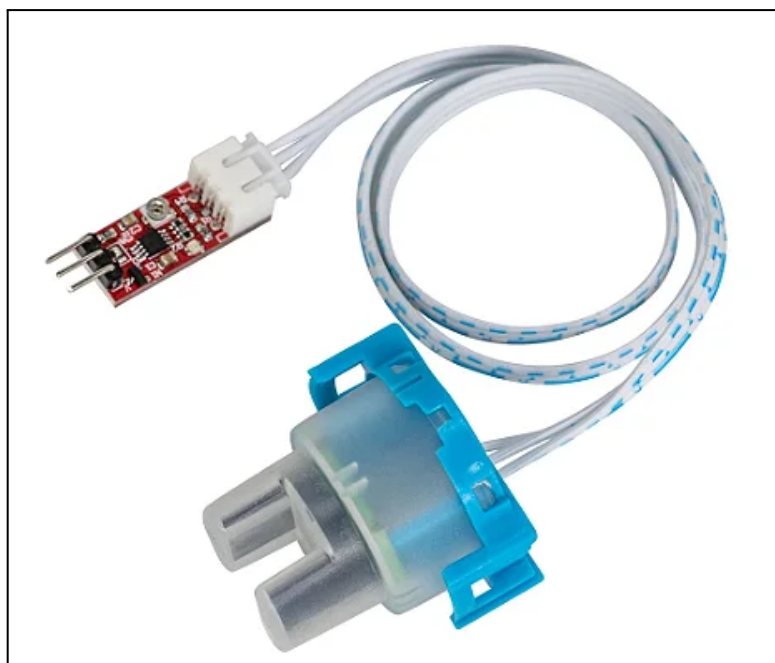
2.5) Resistência de isolamento: mínimo de 100 mΩ;

2.6) Temperatura de operação: -20~90 °C;

2.7) Faixa de medição: 0 ~ 1000 NTU;

2.8) Marca: OEM.

Figura 8 - Sensor turbidez.



Fonte: (SARAVATI, 2021a, p. 1)

3) Sensor digital de Temperatura, figura 9:

3.1) Modelo: DS18B20;

3.2) Alimentação: 3,0 ~ 5V DC;

3.3) Acurácia: ± 0.5 °C;

3.4) Temperatura de medição: -55°C até 125 °C;

3.5) Acurácia: ± 0.5 °C;

3.6) Sinal: digital;

3.7) Resolução: 9 a 12 bits.

Figura 9 - Sensor de temperatura.



Fonte: (ALLDATASHEET, 2022)

4.2 IDE

De acordo com Ribeiro e Silva (2008, p. 2) “um IDE pode ser definido como sendo a união de várias ferramentas que possibilitam o aumento de produtividades dos desenvolvedores em uma determinada linguagem. “

Além disso um bom IDE deve oferecer aos desenvolvedores de *software* um maior aproveitamento e um desenvolvimento com maior agilidade, tendo três importantes elementos: editor, compilador e debugador (RIBEIRO; SILVA, 2008)

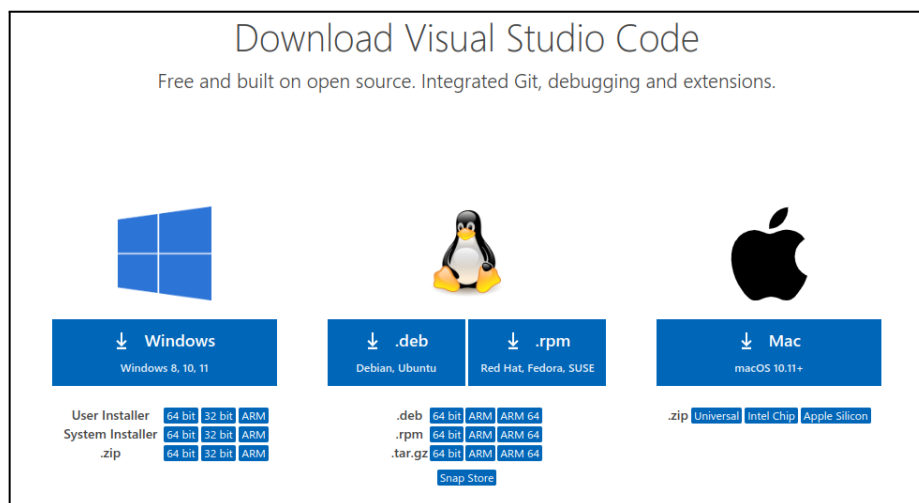
4.3 VISUAL STUDIO CODE

O *Visual Studio Code* (VSCode) é um editor de código fonte leve e aberto, desenvolvido pela *Microsoft* e está disponível para as plataformas: Windows, macOS e Linux (CODE, 2022). Portanto a IDE escolhida é o VSCode.

De acordo com site oficial, o VSCode conta com suporte integrado para JavaScript, TypeScript e Node.js. Além disso, possui um abundante ecossistema de extensões para outras linguagens e ambientes de execução, tais como: C++, C#, Java, Python, PHP, Go, .NET e outras linguagens (CODE, 2022).

Para realizar a instalação do VSCode é preciso entrar no seguinte endereço: <https://code.visualstudio.com/Download>. Uma vez que estiver na página, basta escolher a versão compatível com o sistema operacional desejado, conforme mostra a figura 10.

Figura 10 - Plataforma disponível para *download*.



Fonte: Autoria Própria.

4.4 PREPARAÇÃO DO AMBIENTE DE DESENVOLVIMENTO

Para dar início a implementação do *software* é necessário configurar o ambiente de desenvolvimento. Sendo assim, foi escolhida uma extensão chamada PlatformIO. Tal extensão é uma ferramenta profissional multiplataforma, que pode ser integrada ao VSCode, para que desenvolvedores possam escrever aplicações embarcadas com maior facilidade (PLATFORMIO, 2022).

Além disso, o PlatformIO é uma extensão com mais de 2.000 avaliações de cinco estrelas em todo ecossistema da *Microsoft*. Este recurso também foi instalado por mais de 2.000.000 de usuários em todo o mundo (PLATFORMIO, 2022).

Para realizar a integração deste recurso com o VSCode é preciso realizar as seguintes etapas, figura 11:

- Abrir o VSCode;
- Na barra lateral, clicar em “Extensões”;
- Digitar a extensão “platformio”;
- Em seguida, clicar em “Instalar”;
- Aguardar a instalação e reiniciar a IDE.

Figura 11 - Instalação do PlatformIO.



Fonte: Autoria Própria.

Para certificar se a extensão foi realmente instalada, na barra lateral irá aparecer o ícone correspondente a ela. Com a integração do PlatformIO ao VSCode, a criação de um novo projeto dá-se de forma bem simples, conforme mostra a figura 12.

Figura 12 - criação de um novo projeto.



Fonte: Autoria Própria.

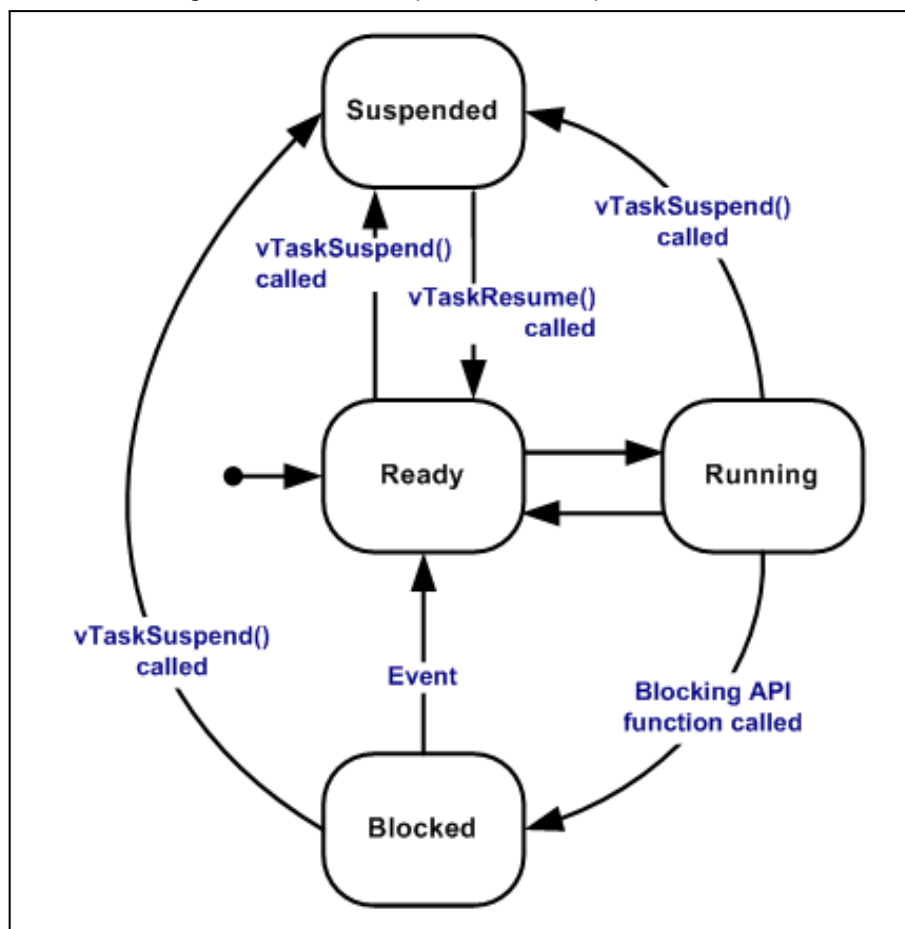
4.5 FREERTOS

Compartilhado de maneira gratuita sob a licença *open source*, o FreeRTOS inclui um kernel e um ecossistema crescente de bibliotecas Internet of Things (IoT), que são adequadas para os mais diversos usos na indústria (FREERTOS, 2022).

De acordo com Justi (2020, p.19) “o FreeRTOS é um kernel de tempo real (ou escalonador de tempo real) que permite que a aplicação desenvolvida seja organizada como um conjunto de threads de execução independentes, aqui chamadas de tarefas.”

Segundo o site oficial do FREERTOS, uma Tarefa (Task) pode existir em um dos seguintes estados, figura 13:

Figura 13 - Estados que uma tarefa pode assumir.



Fonte: (FREERTOS, 2022).

- Em execução (*Running*): atualmente está utilizando o processador.
- Pronto (*Ready*): são aquelas que podem ser executadas (não estão no estado Bloqueado ou Suspenso), mas não estão em execução no momento porque uma tarefa diferente de prioridade igual ou superior já está no estado em execução.
- Bloqueada (*Blocked*): uma Tarefa está no estado Bloqueado se estiver aguardando um evento temporal ou externo.
- Suspensa (*Suspended*): não podem ser selecionadas para entrar no estado de execução, mas as tarefas no estado Suspenso não têm tempo limite.

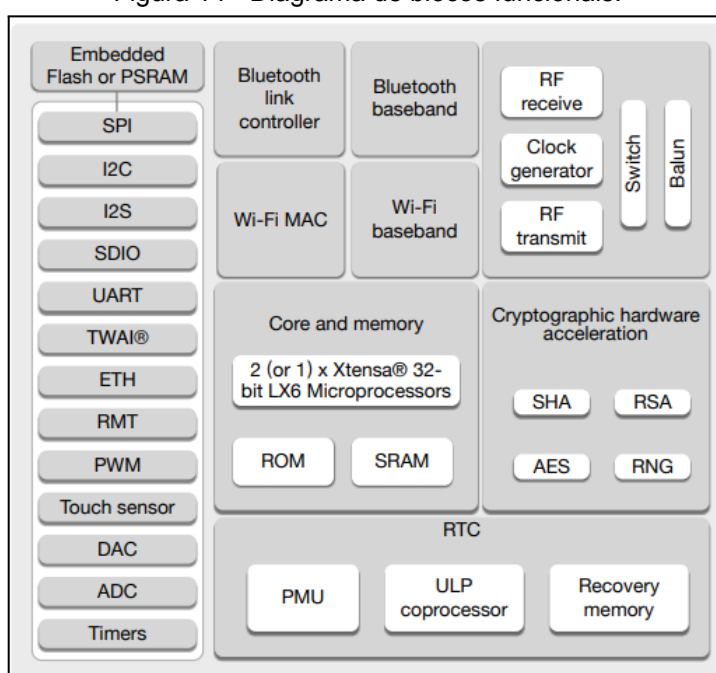
4.6 MICROCONTROLADORES

Para Medeiros (2018) um microcontrolador é um circuito integrado, em outras palavras, trata-se de um computador em um único chip. No mesmo chip são integrados um núcleo (CPU) e circuitos periféricos, como: *clock*, interface de comunicação serial, barramento de entrada e saída, conversores analógicos e digitais. Sendo assim, os microcontroladores são uma ótima solução para aplicações de baixo custo.

4.7 MICROCONTROLADOR ESP32

Segundo a Espressif (2022), fabricante do Esp32, ele é uma solução de *System On Chip (SoC)*, ou seja, sistemas eletrônicos em um circuito integrado, sendo capaz de funcionar em um ambiente de -40°C a $+125^{\circ}\text{C}$. Além disso, ele é altamente integrado, possuindo antena *Wi-Fi* embutida, interface *Bluetooth*, entradas analógicas e digitais, módulo de gerenciamento de energia dentre outras funcionalidades, como mostra o diagrama em bloco da figura 14. Tais características, garantem a confiabilidade em diversas aplicações e diferentes contextos de consumo de energia.

Figura 14 - Diagrama de blocos funcionais.



Fonte: (Espressif, 2022)

5 BOAS PRÁTICAS DE DESENVOLVIMENTO

O *software* não pode apenas funcionar, ele precisa de um padrão de conceitos e boas práticas para manter uma boa legibilidade e organização (MAGALHÃES; TIOSSO, 2019).

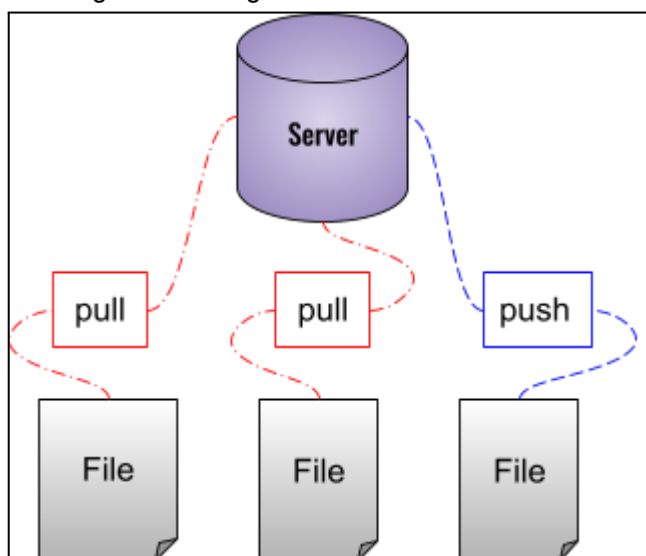
Para Magalhães; Tiosso (2019) na condição do código estar limpo e organizado, isso garante o bom entendimento entre os desenvolvedores da equipe e, conseqüentemente, aumenta a velocidade de produção do *software*. Portanto, boas práticas são, de fato, indispensáveis.

É crucial que as boas práticas sejam implementadas, pois os desenvolvedores de *software*, independentemente de tecnologia, gastam muito mais tempo lendo o código de uma aplicação do que programando em si.

5.1 CONTROLE DE VERSÃO

Para Luz (2018), o controle de versão é um sistema que preserva as alterações de um ou mais arquivos ao longo do tempo, isso permite acessar uma versão específica a qualquer momento. As modificações podem ser locais, como salvar os arquivos no disco rígido do computador ou de maneira remota, preservando as modificações em um servidor remoto, figura 15.

Figura 15 - Diagrama de controle de versão.

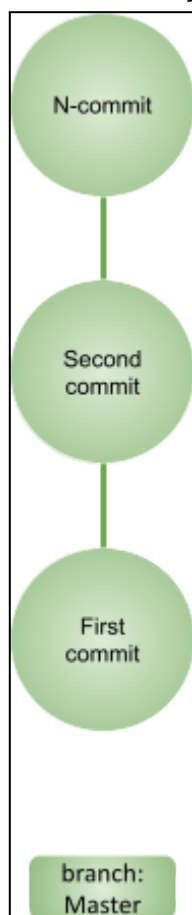


Fonte: Autoria própria.

5.2 SISTEMA DE CONTROLE DE VERSÃO DISTRIBUÍDO

O sistema de controle de versão mais utilizado entre os desenvolvedores de *software* é o Git, permitindo salvar o histórico do projeto, como mostra a figura 16 (LUZ, 2018).

Figura 16 - Histórico de alterações do projeto.



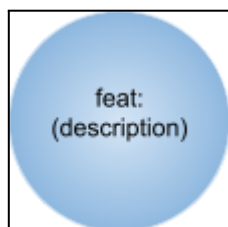
Fonte: Autoria própria.

O git é um *software open-source* desenvolvido por Linus Torvalds, em 2005 e segundo Silverman (2013) com o Git é possível:

- I. muitos desenvolvedores trabalhem em um projeto,
- II. averiguar o estado do projeto em qualquer ponto no passado,
- III. exibir alterações entre as múltiplas etapas do projeto,
- IV. partilhar o desenvolvimento do projeto em ramos independentes.

Cada alteração do *software* embarcado equivale a um “*commit*”, identificado com o prefixo “*feat*”, juntamente com um identificador e uma breve descrição do que foi implementado, como ilustra a figura 17.

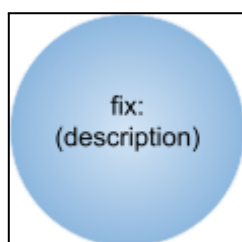
Figura 17 - Identificação de um *commit* de uma funcionalidade.



Fonte: Autoria própria.

Caso algum erro de funcionalidade seja encontrado, ele será corrigido por meio de um novo “*commit*”, identificado com o prefixo “*fix*” e uma descrição, conforme a figura 18.

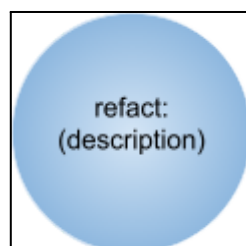
Figura 18 - Identificação de um “*commit*” de correção.



Fonte: Autoria própria.

No caso de refatoração, o “*commit*” será identificado com o prefixo “*refact*”, juntamente com a descrição da refatoração, figura 19.

Figura 19 - Identificação de um “*commit*” de refatoração.



Fonte: Autoria própria

5.3 REPOSITÓRIO REMOTO

Segundo ROCHA; SILVA; MORO (2016, p. 103):

A facilidade de extrair dados de várias fontes e conectá-los tem contribuído para o crescimento de Redes Sociais. Uma rede possível é a de desenvolvedores de código a partir de dados de plataformas como o GitHub, onde participantes são desenvolvedores que podem criar/contribuir/compartilhar/buscar por repositórios de projetos de acordo com assuntos e Linguagens de Programação (LPs) em que foram desenvolvidos.

O GitHub é a maior e mais popular plataforma de código aberto no mundo, com mais de 15 milhões de usuários e mais de 38 milhões de projetos cadastrados. Portanto o Github é o local onde será armazenado o código fonte.

6 TESTES EM LABORATÓRIO

De acordo com OLIVEIRA (2003), “o teste de software é uma atividade crítica na garantia da qualidade. Os seres humanos são passíveis de erros e a atividade de testes é essencial para diminuir os possíveis erros que possam ser encontrados”.

Segundo Cérboli (2017, p. 16) “o principal objetivo dos testes é buscar o maior número de falhas possíveis, no menor tempo e esforço, demonstrando aos que desenvolvem que os resultados não estão de acordo com o especificado.”

A partir disso, foram realizados testes funcionais, tendo como propósito:(I) verificar se o *software* embarcado executa corretamente suas funções programadas; (II) se a implementação dos requisitos foram apropriadas; (III) se o sistema é capaz de sustentar sua correta execução por um período contínuo de execução.

Portanto, os testes em laboratório serviram para investigar o comportamento do *software* em um ambiente controlado. Para a realização dos testes em laboratório foram utilizados os seguintes equipamentos:

- 1) HI98196: de acordo com a fabricante, Hanna Instruments, é um medidor multiparâmetro portátil que mede até 7 diferentes parâmetros de qualidade de água. Este equipamento, figura 20, consegue realizar a leitura de grandezas como o pH, ORP, o oxigênio dissolvido e a temperatura.

Figura 20 - Equipamento HI98196.



Fonte: (HANNA INSTRUMENTS, HI98196, 2022).

- 2) HI98703: de acordo com a fabricante, Hanna Instruments, este equipamento, figura 21, é um medidor de turvação de elevada precisão ($\pm 2\%$ da leitura mais 0,02 NTU). O medidor é fornecido com os padrões de turvação primários AMCO-AEPA-1 utilizados para a verificação da calibração e do desempenho. O HI98703 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.

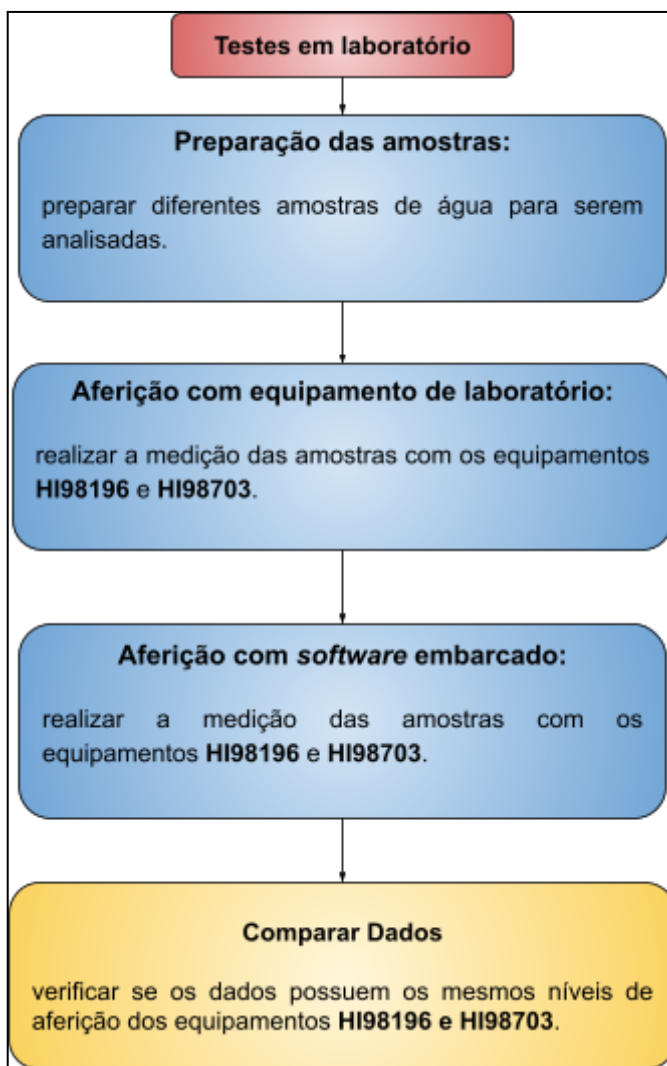
Figura 21 - Equipamento HI98703.



Fonte: (HANNA INSTRUMENTS, HI98703-02, 2022).

Dessa maneira, foi realizado a preparação de diferentes amostras de água para serem analisadas. Tais amostras foram aferidas tanto pelo *software* embarcado quanto por sondas comerciais robustas, HI98196 e HI98703. A comparação entre os equipamentos supracitados e o *software* embarcado serve para assegurar que a aplicação terá os mesmos níveis de precisão dos produtos disponíveis no mercado. As etapas de testes em laboratório estão na figura 22.

Figura 22 - Etapas de testes em laboratório.



Fonte: Autoria própria.

O *hardware* pode ser visualizado no trabalho Desenvolvimento de um protótipo para o monitoramento físico/químico das águas na piscicultura (TEIXEIRA, 2022).

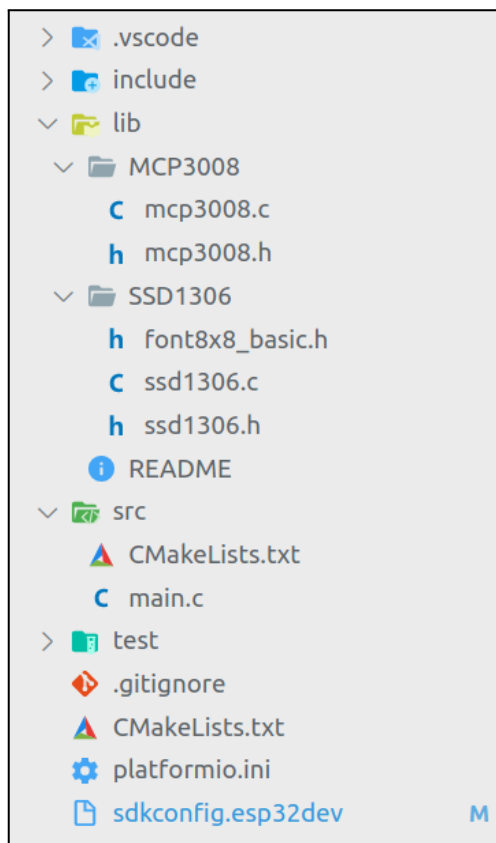
7 RESULTADOS OBTIDOS

Este capítulo apresenta quatro seções, em que mostra os resultados atingidos após a implementação de todas as etapas do desenvolvimento do *software* embarcado: estrutura de pastas do projeto, padronização do código, histórico de *commits*, aferição e comparação dos dados.

7.1 ESTRUTURA DE PASTAS DO PROJETO

A implementação da estrutura de pasta do projeto é mostrada na figura 23. Vale ressaltar que a pasta “*src*” contém o arquivo principal, neste arquivo é definido as inicializações com os sensores, inicialização do display I2C, cálculo dos parâmetros hídricos e a criação de uma única “*Task*”. Enquanto a pasta “*lib*” será usada para adicionar as dependências necessárias: display I2C e conversor MCP3008.

Figura 23 - Estrutura de pastas do projeto.



Fonte: Autoria Própria.

7.2 PADRONIZAÇÃO DO CÓDIGO DO PROJETO

Para o desenvolvimento do *software* embarcado foi adotado e adaptado algumas regras do *Google C++ Style Guide*. Sendo assim, as seguintes regras foram aplicadas:

- 1) O código está em um único idioma, o inglês;
- 2) O código está indentado com apenas espaços e recuo dois espaços de cada vez;
- 3) Não utilização de muitos parâmetros em funções;
- 4) Os nomes de variáveis, incluindo parâmetros de função, e membros de dados são todos em minúsculas, separados com caracteres "underscore" entre as palavras (*snake_case*).
- 5) Os nomes de funções é iniciado com letra minúscula e unidas com "underscore" (*snake_case*), figura 24:

Figura 24 - Padrão de função

```
double calc_temperature(DeviceAddress *temp_sensors) { ...  
double calc_ph(MCP3008 *device){ ...  
double calc_turbity(MCP3008 *device){ ...  
void show_data(double turbity, double ph, double temperature){ ...  
void set_display(){ ...  
void app_main(void){ ...
```

Fonte: Autoria própria.

6) Constantes são declaradas com letras maiúsculas, figura 25:

Figura 25 - Padrão de constantes

```
14 #define SDA_PIN GPIO_NUM_21
15 #define SCL_PIN GPIO_NUM_22
16 #define I2C_FREQUENCY 100000
17 #define VCC_ADC 5000.0
```

Fonte: Autoria própria.

7) Não se aplica desnecessariamente a expressão de retorno entre parênteses;

8) Se a linha for muito extensa, é feita uma “quebra” de linha;

9) As funções possuem apenas um único ponto de retorno, figura 26:

Figura 26 - Único ponto de retorno.

```
32 double calc_turbidez(MCP3008 *device){
33     double voltage, ntu;
34
35     mcp3008_read(device, 0);
36     voltage = ((float) device->data / (float)(1 << device->bits)) * VCC_ADC;
37
38     if ( voltage < 2500 ) {
39         ntu = 3000;
40     }
41     else if (voltage > 4200) {
42         ntu = 0;
43     }
44     else {
45         ntu = -1120.4 * pow(voltage/1000, 2) + 5742.3 * (voltage/1000) - 4352.8;
46     }
47
48     ESP_LOGI("TURBIDEZ", "NTU: %.2f, Voltage: %.2fmV, data: %d", ntu, voltage, device->data);
49     return ntu;
50 }
```

Fonte: Autoria própria

7.3 HISTÓRICO DE ALTERAÇÕES DO PROJETO

A medida que cada requisito foi concluído, ele foi integrado ao ramo principal do projeto. Sendo assim, é possível visualizar o histórico de alterações do projeto na figura 27.

Figura 27 - Histórico de alterações do projeto.

```
(base) benilton.andrade@smamb10263:~/Documentos/github/waterfall
• $ git log
commit e522cfd72946059d2ef1e1df2b9ecb85396fb56b (HEAD -> master, origin/master, origin/HEAD)
Author: benilton02 <bsa.ele@uea.edu.br>
Date:   Fri Oct 7 23:45:07 2022 -0400

    feat: show data on display

commit 9661da88555800b223ec44ee26c22f9ed1b2f5f3
Author: benilton02 <bsa.ele@uea.edu.br>
Date:   Fri Oct 7 02:13:04 2022 -0400

    feat: adding temperature read DS18B20

commit e2f671200af6d2656257ba7d82201fc5aeff2555
Author: benilton02 <bsa.ele@uea.edu.br>
Date:   Fri Oct 7 02:13:04 2022 -0400

    feat: add read sensor PH

commit 8e6c01e3044f6bc13afaf9e63f73103c32325be4
Author: benilton02 <bsa.ele@uea.edu.br>
Date:   Fri Oct 7 02:13:04 2022 -0400

    feat: read turbidez sensor

commit e7154328af3050f948ff1bde087f345a11073127
Author: benilton02 <bsa.ele@uea.edu.br>
Date:   Wed Sep 28 02:39:48 2022 -0400

    feat: first commit
```

Fonte: Autoria própria

Para ter acesso ao código fonte, basta ir até o endereço: <https://github.com/benilton02/waterfall>.

7.4 AFERIÇÃO E COMPARAÇÃO DOS DADOS

Para realizar a medição dos parâmetros hídricos, foi realizada uma comparação com algumas amostras de água e verificando o funcionamento do *software* embarcado. A primeira amostra, figura 28, é uma água mineral.

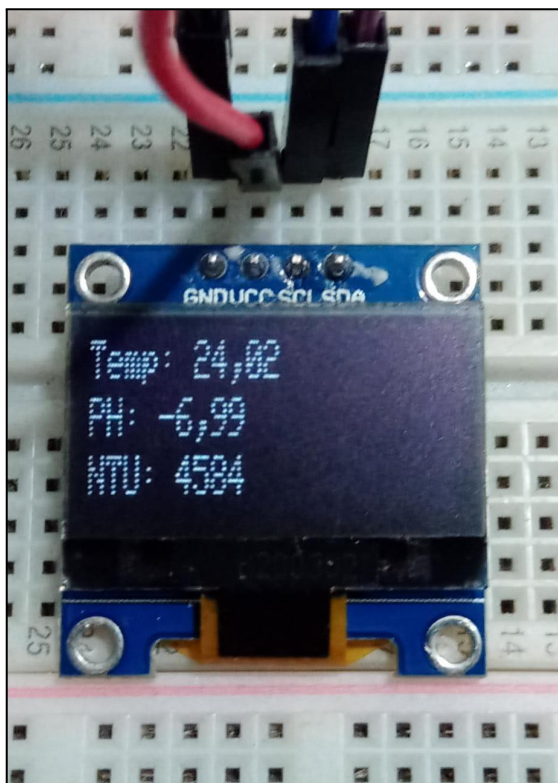
Figura 28 - Aferição da turbidez da água mineral com o equipamento HI98703.



Fonte: Autoria própria.

Por meio do *software* embarcado não foi possível aferir a amostra supracitada, isso ocorreu por conta do baixo nível de turbidez da água mineral. Como mostra a figura 29, a leitura de turbidez é incompatível com a da figura 28.

Figura 29 - Aferição da primeira amostra de turbidez da água por meio do software.



Fonte: Autoria própria.

A segunda amostra é uma água mineral misturada com terra. A figura 30 mostra o nível de turbidez aferido da segunda amostra.

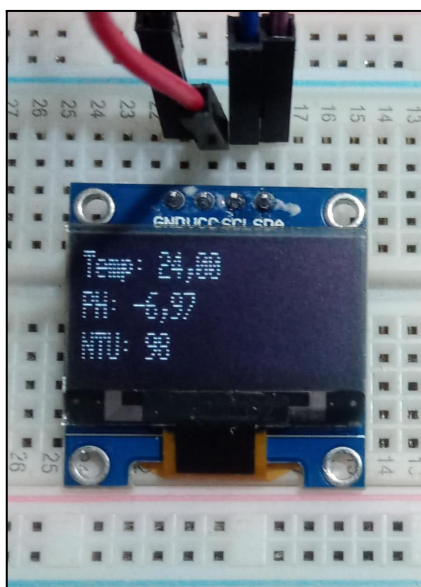
Figura 30 - Aferição da turbidez da segunda amostra com o equipamento HI98703.



Fonte: Autoria própria.

Por meio do *software* embarcado foi possível aferir a segunda amostra de água. Como mostra a figura 31, o *software* foi capaz de realizar a medição e exibir o resultado, no entanto, os valores do *software* e do equipamento combinaram.

Figura 31 - Aferição da segunda amostra de turbidez da água por meio do software.



Fonte: Autoria própria.

A terceira amostra é uma água mineral misturada com leite em pó. A figura 32 mostra o nível de turbidez aferido da terceira amostra.

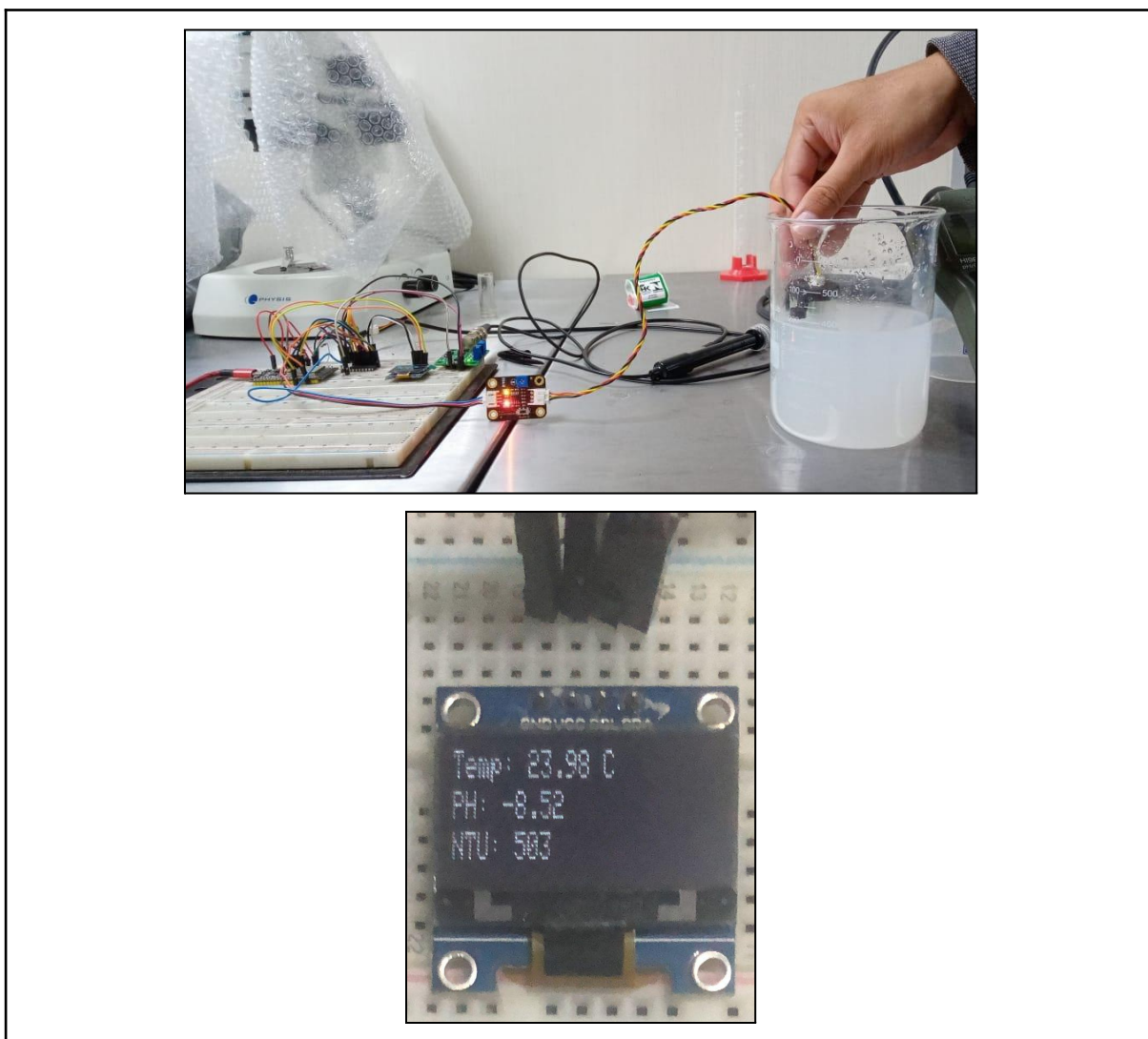
Figura 32 - Aferição da turbidez da água por meio do software.



Fonte: Autoria própria.

Por meio do *software* embarcado foi possível aferir a terceira amostra de turbidez, água com leite em pó. Como mostra a figura 33, o *software* foi capaz de realizar a leitura e exibi-la no *display*, porém não teve a mesma exatidão do equipamento de comparação.

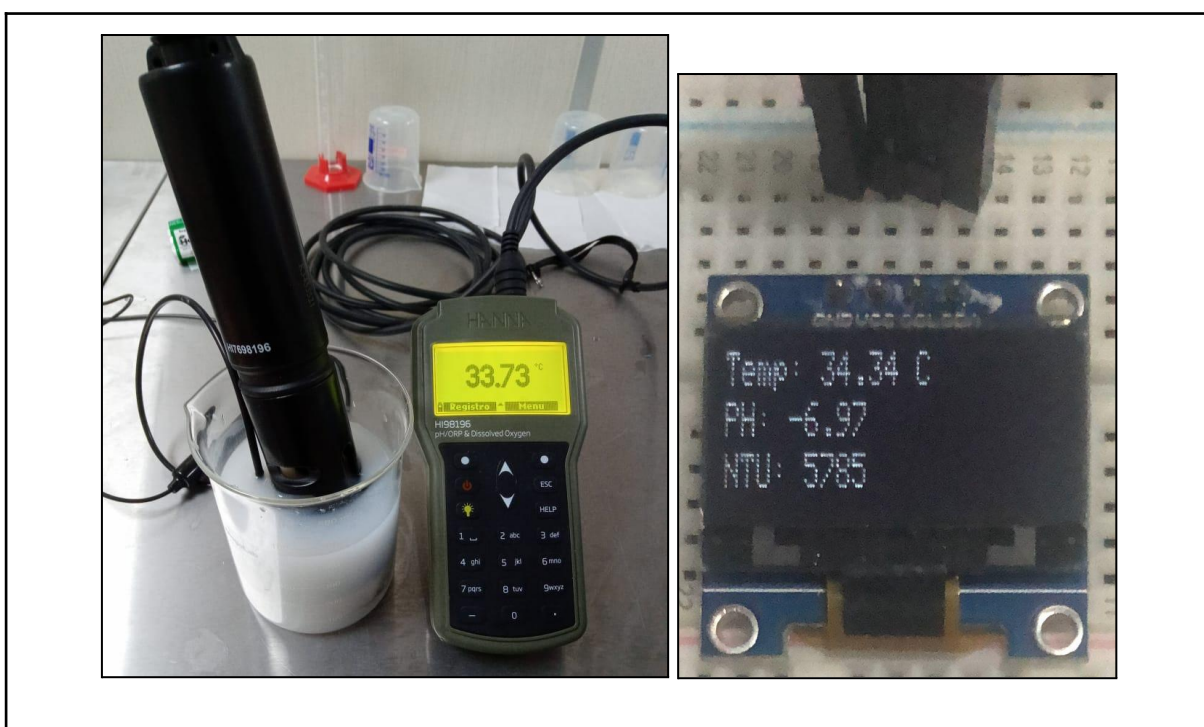
Figura 33 - Aferição da turbidez da água com leite em pó.



Fonte: Autoria própria.

Para realizar a medição da temperatura da água, foi utilizada uma amostra de água mineral misturada com leite em pó. Para tal, a amostra foi aquecida em um forno micro-ondas. Então, utilizando o equipamento HI98196 foi aferida a temperatura. Ao mesmo tempo, foi imerso na amostra o sensor digital de temperatura para determinar a aferição por meio do *software*. Os valores aferidos tanto pelo *software* embarcado quanto pela sonda de teste estão na figura 34.

Figura 34 - Medição da temperatura da amostra de água mineral com leite em pó.



Fonte: Autoria própria.

A amostra de água mineral também foi aquecida em um forno micro-ondas, porém por mais tempo. Os valores aferidos tanto pelo *software* embarcado quanto pela sonda de teste estão na figura 35.

Figura 35 - Medição da temperatura da água mineral aquecida no microondas.



Fonte: Autoria própria.

Após um determinado período de tempo, a amostra de água mineral foi novamente aferida. Os valores aferidos tanto pelo software embarcado quanto pela sonda de teste estão na figura 36.

Figura 36 - Medição da temperatura da amostra de água mineral.



Fonte: Autoria própria.

Para realizar a medição do pH da água, foi utilizado uma amostra de água mineral, cujo pH é neutro. Para tal, foi utilizado o equipamento HI98196 e comparado com o valor medido pelo *software*, figura 37.

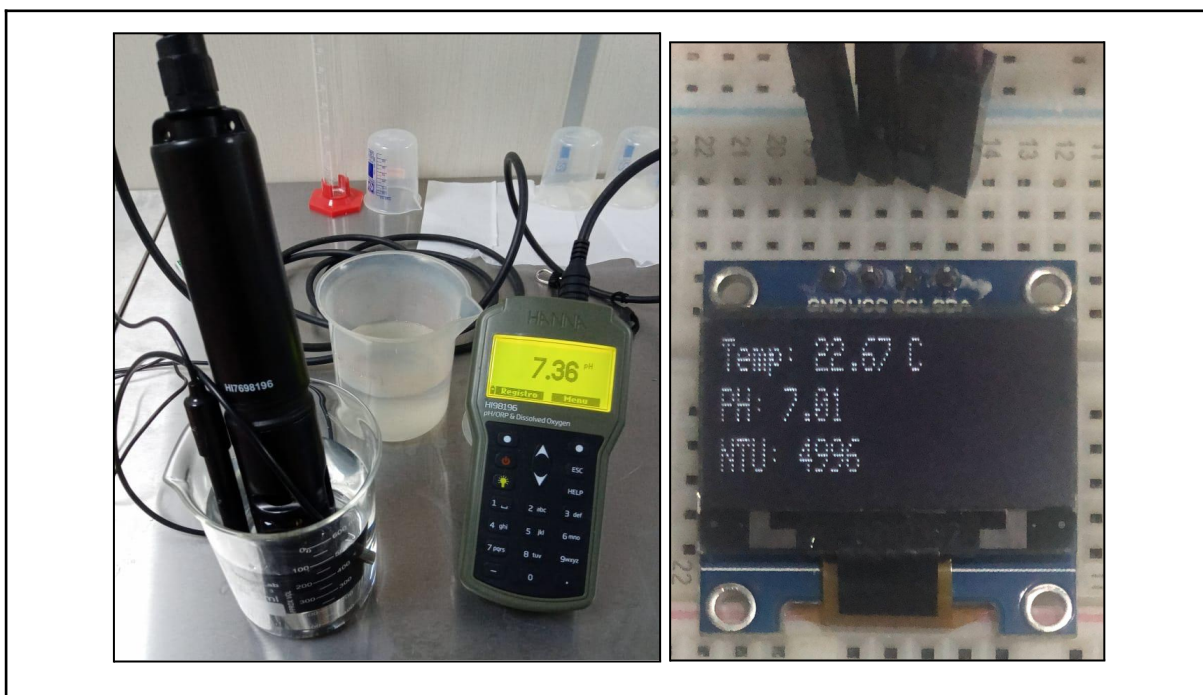
Figura 37 - Comparação do pH da água mineral.



Fonte: Autoria própria.

Para realizar outra medição do pH da água, foi utilizado uma amostra de um filtro da marca *Acqua Bella*. Para tal, foi utilizado o equipamento HI98196. Ao mesmo tempo, foi inserido o eletrodo do sensor de pH para determinar a aferição por meio do *software*, figura 38.

Figura 38 - Comparação do pH da água mineral do filtro *Acqua Bella*.



Fonte: Autoria própria.

Os valores obtidos com o *software* embarcado e com o equipamento de teste estão apresentados na Tabela 4.

Tabela 4 - Comparação dos parâmetros aferidos.

Parâmetro	Aferição com os equipamentos HI98196 e HI98703-02.	Aferição com o software embarcado	Erro Relativo (%)
Temperatura	água com leite: 33,73 °C	água com leite: 34,34 °C	+ 1,80
	água mineral: 38,48 °C	água mineral: 39,25 °C	+ 2,00
	água mineral: 21,96 °C	água mineral: 22,01 °C	+ 0,22
PH	água mineral 1: 7,00	água mineral: 6,82	- 2,57
	água filtrada: 7,36	água filtrada: 7,01	- 4,75
Turbidez	água mineral: 0,91 NTU	água mineral: 4584 NTU	+ 503636,26
	água com terra: 78,8 NTU	água com terra: 98 NTU	+ 24,36
	água com leite: 418 NTU	água com leite: 503 NTU	+ 20,33

Fonte: Autoria própria.

CONCLUSÃO

No decorrer do processo de desenvolvimento dessa pesquisa foi realizado um estudo sobre os principais parâmetros hídricos. Por conseguinte, foi feita uma análise e levantamento dos requisitos funcionais e não funcionais para o desenvolvimento do *software* embarcado, chegando a três parâmetros hídricos: temperatura, ph e turbidez.

Além disso, foi apresentada a configuração para o ambiente de desenvolvimento (IDE). Por meio da IDE e da extensão PlatformIO, foi criada uma estrutura de pastas do projeto para mantê-lo com baixo acoplamento e alta coesão.

Para registrar o histórico de alterações do *software* embarcado foi realizado um estudo sobre o funcionamento e as principais características do sistema de controle de versão distribuído, o *Git*. Com a ferramenta supracitada, cada alteração no código corresponde a um requisito funcional do *software* embarcado, tornando o *software* facilmente rastreável. Ademais, um repositório remoto foi criado no *GitHub* para fácil distribuição e consulta do projeto desenvolvido.

Da mesma forma, foi realizada uma pesquisa sobre boas práticas de desenvolvimento de *software*. Consequentemente, foi adotado e adaptado um paradigma para auxiliar no desenvolvimento. A intenção de padronizar o código do projeto foi desenvolver de forma clara, legível e com restrições razoáveis.

Também, foi feita uma análise sobre o funcionamento e as principais características do microcontrolador ESP32. Foram apresentadas, de forma detalhada, a aferição e comparação dos resultados com produtos comerciais para garantir a validação do *software* embarcado.

No desenvolvimento do projeto notou-se que para desenvolver um *software* embarcado, foi necessário o conhecimento de diversas áreas cursadas ao longo do curso de Engenharia Eletrônica.

Acerca da hipótese citada no início deste trabalho e com base nos dados coletados, concluiu-se que o *software* embarcado é capaz de monitorar os diferentes parâmetros hídricos em tempo real, utilizando tanto sensor digital como sensor analógico associados ao microcontrolador ESP32. Além disso, foi verificado que é possível utilizar o FreeRTOS para gerenciar e exibir os parâmetros hídricos.

Para a realização de trabalhos futuros, recomenda-se: (I) realizar a comparação com um maior número de amostras, a fim de manter a precisão no gerenciamento de informações em tempo real; (II) utilização da metodologia *gitflow* para adição de novas funcionalidades, refatoração ou correções no código; (III) realizar um teste de campo em um viveiro de peixes; (IV) escolha de um sensor mais robusto para medir o parâmetro de turbidez.

REFERÊNCIAS

ALLDATASHEET. **Electronic Components Datasheet Search**, 2022. Disponível em:

<https://www.alldatasheet.com/view.jsp?Searchword=Ds18b20%20datasheet&gclid=Cj0KCQjwnP-ZBhDiARIsAH3FSReoooYm-n2VEEkUluhGitiZPs9vkCJb1mvinhT3ZrNepI9tuh89MT0aAu9rEALw_wcB>. Acesso em 7 de out. de 2022.

BARBOSA, G. S.; DRACH, P. R.; CORBELLA, O. D. **A Conceptual Review of the Terms Sustainable Development and Sustainability**. International Journal of Social Sciences, [S.l.] 2014. Disponível em:

<https://www.iises.net/download/Soubory/soubory-puvodni/pp-01-15_ijossV3N2.pdf>. Acesso em: 24 fev. 2022.

Brasil. Lei nº 9.433, de 8 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 09/01/1997, P. 470. Disponível em:

<http://www.planalto.gov.br/ccivil_03/leis/L9433.htm>. Acesso em: 26 fev. 2022.

CANTELE, Tatiana Dias; LIMA, Eudes de Castro; BORGES, Luís Antônio Coimbra. **Panorama dos Recursos Hídricos no Mundo e no Brasil**. Revista em Agronegócio e Meio Ambiente, Maringá, 2018 Disponível em:

<<https://periodicos.unicesumar.edu.br/index.php/rama/article/view/5425/3350>>. Acesso em: 24 fev. 2022.

CARVALHO, Deyse Maria Portela; JUNIOR, Aristides da Rocha Oliveira; OLIVEIRA, Fabiana Lucena. **Análise do desempenho econômico recente da piscicultura no Estado do Amazonas**. Brazilian Journal of Development, 2020. Disponível em:

<<https://www.brazilianjournals.com/index.php/BRJD/article/view/17623/14302>>. Acesso em: 24 fev. 2022.

CARVALHO, Pedro F. **Técnicas de Levantamento de Requisitos**. Rio Preto, 2009 Disponível em:

<http://pedrofcarvalho.com.br/PDF/ENGENHARIA_ANALISE_LEVANTAMENTO_REQUISITOS_2.pdf>. Acesso em: 25 fev. 2022.

CÉRGOLI, Robson; **Uma Análise Da Aplicação De Testes No Desenvolvimento De Um Sistema Erp**. São Paulo, 2017. Disponível

em:<https://spo.ifsp.edu.br/images/phocadownload/DOCUMENTOS_MENU_LATERAL_FIXO/POS_GRADUA%C3%87%C3%83O/ESPECIALIZA%C3%87%C3%83O/Gest%C3%A3o_da_Tecnologia_da_Infoma%C3%A7%C3%A3o_____/PRODUCAO/2017/Uma_An%C3%A1lise_da_Aplica%C3%A7%C3%A3o_de_Testes_no_Developimento_de_um_Sistema_ERP.pdf>. Acesso em: 26 de out. 2022.

CODE. **Getting Started**, 2022. Disponível em:<<https://code.visualstudio.com/docs>>. Acesso em: 5 out. 2022.

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://www.dimap.ufrn.br/~sbmac/ermac2008/Anais/Resumos%20Estendidos/Analise%20da%20turbidez_Aislan%20Correia.pdf>. Acesso em: 28 ago. 2021.

ESPRESSIF SYSTEMS. **ESP32 Series Datasheet**, 2022. Disponível em:<[esp32_datasheet_en.pdf](#)>. Acesso em: 25 fev. 2022.

FREERTOS. **FreeRTOS Kernel Quick Start Guide**, 2022. Disponível em: <<https://www.freertos.org/FreeRTOS-quick-start-guide.html>>. Acesso em: 5 out. 2022.

GOOGLE. **Google C++ Style Guide**. Disponível em: <<https://google.github.io/styleguide/cppguide.html>>. Acesso em: 23 ago. 2022.

HANNA INSTRUMENTS. **HI98703-02 Medidor de turvação**. 2022. Disponível em:<<https://www.hanna.pt/produto/hi98703-02>>. Acesso em 26 out. 2022

HANNA INSTRUMENTS. **HI98196 Medidor Portátil Profissional À Prova d'Água Multiparâmetro pH/ORP/OD/Pressão/Temperatura**. Disponível em:<<https://hannainst.com.br/produto/hi98196-medidor-portatil-a-prova-dagua-multiparametro-ph-orp-od/>>. Acesso em 26 out. 2022

IDAM, Instituto de Desenvolvimento Agropecuário e Florestal Sustentável do Estado do Amazonas. **Relatório de atividade IDAM 2020**. Manaus, 2020. Disponível em: <http://www.idam.am.gov.br/wp-content/uploads/2021/09/RAIDAM2020_web_vfinal.pdf>. Acesso em: 25 fev. 2022.

LOPES, Jackelline Cristina Ost. **Técnico em agropecuária: piscicultura**. Florianópolis: EDUFPI, 2012.

JUNIOR, Francisco Ramalho; FERRARO, Nicolau Gilberto; SOARES, Paulo Antônio de Toledo. **Os Fundamentos da Física 2 Termologia Óptica Ondas**. 9 ed. São Paulo: Editora Moderna, 2007

JUSTI, Milena Cherubini. **Análise Do Uso De Rtos Na Implementação De Controle Supervisório**. Rio Grande do Sul, Porto Alegre, 2020. Disponível em: <<https://www.lume.ufrgs.br/bitstream/handle/10183/217369/001120673.pdf?sequence=1>>. Acesso em: 1 out. 2022.

LUZ, Daniel Almeida. **Roger-that flow: um uso aprimorado do git-flow**. Brasília, 2018. Disponível em: <https://bdm.unb.br/bitstream/10483/21938/1/2018_DanielAlmeidaLuz_tcc.pdf>. Acesso em: 26 fev. 2022.

MAGALHÃES, P. A.; TIOSSO, F. **Código Limpo: Padrões E Técnicas No Desenvolvimento De Software**. Interface Tecnológica, 2019. Disponível em: <<https://revista.fatectq.edu.br/index.php/interfacetecnologica/article/download/597/348/2581>>. Acesso em: 26 fev. 2022.

MEDEIROS, Savio Gomes Fernandes de. **Sistema De Medição De Variáveis Atmosféricas Usando Microcontrolador**. Caraúbas, 2018. Disponível em: <https://repositorio.ufersa.edu.br/bitstream/prefix/2906/2/SavioGFM_MONO.pdf>. Acesso em: 27 fev. 2022.

ENVICON QUÍMICA E MEIO AMBIENTE. **Monitoramento on-line de turbidez no tratamento de água – Mais do que uma melhoria: uma necessidade**. dia 2 ago. 2020. Disponível em: <<https://www.envicon.com.br/noticias/2020/08/02/monitoramento-on-line-de-turbidez-no-tratamento-de-agua-mais-do-que-uma-melhoria-uma-necessidade/>>. Acesso em: 23 jun. 2022

OLIVEIRA, A. C. de. **ETIERP – Estratégia de Testes para Implantação de Sistemas ERP**. Universidade de Campinas – Mestre em Ciência de Computação, Dezembro, 2003.

OLIVEIRA, Francisco Carlos de Mattos Brito. **Interação Humano Computador**. – 2. ed. Fortaleza, EdUECE, 2015. Disponível em <https://educapes.capes.gov.br/bitstream/capes/432049/2/Livro_Interac%CC%A7a%CC%83o%20Humano%20Computador.pdf> Acesso em: 15 set. 2022.

PATSKO, Luís Fernando. TUTORIAL Aplicações, Funcionamento e Utilização de Sensores. [S.l.], 2006. Disponível em: <https://maxwellbohr.com.br/downloads/robotica/mec1000_kdr5000/tutorial_eletronica_-_aplicacoes_e_funcionamento_de_sensores.pdf> . Acesso em: 30 jun. 2022.

PEZZATO, L. E.; CASTAGNOLLI, N.; ROSSI, F. **Nutrição e alimentação de peixes**. Viçosa, MG: CPT, 2001, 72 p.

PINTO, Magda Cristina Ferreira. **Manual Medição in loco: Temperatura, pH, Condutividade Elétrica e Oxigênio Dissolvido**. Belo Horizonte, 2007. Disponível em: <https://cprm.gov.br/publique/media/gestao_territorial/geologia_medica/manual_medicoes_T_%20pH_OD.pdf>. Acesso em: 21 jun. 2022.

PLATFORMIO. **PlatformIO IDE for VSCode**, 2022. Disponível em: <<https://docs.platformio.org/en/stable/integration/ide/vscode.html>>. Acesso em: 6 out. 2022.

RIBEIRO, Alisson; SILVA, Gabriel da. **Extensão Das Funcionalidades De Um Ambiente Integrado De Desenvolvimento Para Linguagem Php-Gtk: Modelagem E Implementação**. I Jornada Científica e VI FIPA do CEFET, Bambuí, MG, 2008. Acesso em: 7 out 2022

ROCHA, L. M.; SILVA, T. H. P.; MORO, M. M. **Análise da Contribuição para Código entre Repositórios do GitHub**. Salvador, 2016. Disponível em:

<<https://homepages.dcc.ufmg.br/~mirella/pdf/2016.SBBDsROCHAhort.Rocha.pdf>>. Acesso em: 17 ago. 2022.

SARAVATI. **Módulo Sensor de Turbidez Partículas Suspensas na Água**. São Paulo, 2021a. Disponível em: <https://www.saravati.com.br/modulo-sensor-de-turbidez-particulas-suspensas-na-agua?utm_source=Site&utm_medium=GoogleMerchant&utm_campaign=GoogleMerchant&gclid=Cj0KCQjwt-6LBhDIARIsAIPRQclsrBlfGi6nzgWA3k7qukLdCYCbbgulEoYLnNoJOvpJc63z93QA72caAvbKEALw_wcB>. Acesso em: 7 out. 2022.

SARAVATI. **Sensor PH Módulo PH4502C + PH Eletrodo**. São Paulo, 2021c. Disponível em: <<https://www.saravati.com.br/Sensor-PH-Modulo-PH4502C-PH-Eletrodo>>. Acesso em: 7 out. 2022.

SILVA, F. P.; CARVALHO, C. V. A.; CARDOSO, A. M. **A Importância De Políticas Públicas Para A Implementação Do Reuso Da Água No Brasil**. 2019. Episteme Transversalis. Disponível: <<http://revista.ugb.edu.br/ojs302/index.php/episteme/article/view/1347/1068>>. Acesso em: 27 fev. 2022.

SILVERMAN, Richard E. **Git: guia prático**. São Paulo: Novatec, 2013.

TEIXEIRA, T. A. **Desenvolvimento de um protótipo para monitoramento físico/químico das águas na piscicultura**. Universidade do Estado do Amazonas, 2022. Acesso em: 13 out. 2013.