



Automatização do processo de monitoramento da qualidade das águas em cursos d'água

Rafael de Souza Duarte - a42930

Dissertação apresentada à Escola Superior de Tecnologia e de Gestão de Bragança para obtenção do Grau de Mestre em Sistemas de Informação, no âmbito da dupla diplomação com o Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)

Trabalho orientado por:

Prof. Doutor Getúlio Paulo Peixoto Igrejas (IPB)

Prof. Mestre Luciano Nascimento Moreira (CEFET-MG)

Esta dissertação não inclui as críticas e sugestões feitas pelo Júri.

Bragança

2019-2020



Automatização do processo de monitoramento da qualidade das águas em cursos d'água

Rafael de Souza Duarte - a42930

Dissertação apresentada à Escola Superior de Tecnologia e de Gestão de Bragança para obtenção do Grau de Mestre em Sistemas de Informação.

Trabalho orientado por:

Prof. Doutor Getúlio Paulo Peixoto Igrejas (IPB)

Prof. Mestre Luciano Nascimento Moreira (CEFET-MG)

Esta dissertação não inclui as críticas e sugestões feitas pelo Júri.

Bragança

2019-2020

Dedicatória

Agradeço a todas as pessoas que me ajudaram a chegar até aqui, desde o primário até o término do mestrado. É muito gratificante ter tido a oportunidade de realizar um mestrado, e com certeza, isso só foi possível devido o esforço dos diversos professores que tive durante a minha trajetória acadêmica.

Agradeço especialmente aos meus pais, Robson e Maria Aparecida, que sempre estiveram ao meu lado apoiando minhas decisões e dando todo suporte necessário. Agradeço a minha namorada, Izabelly, que esteve ao meu lado a todo momento, mesmo a distância, durante todo o tempo do projeto. Agradeço também a família que fiz em Portugal — especialmente ao Márcio, Sérgio e Rafaela — que sempre me incentivaram. Agradeço aos professores Luciano, Fernando e Getúlio por todo o apoio durante o projeto. Sempre atenciosos e disponíveis, representaram uma das importantes bases de conhecimento durante o curso. Agradeço aos meus amigos do curso de Engenharia da Computação (Brasil) e de Sistemas de Informação (Portugal), por tornar mais agradável a convivência diária no período acadêmico.

Agradeço a Secretaria de Relações Internacionais, por ter disponibilizado todo suporte financeiro e acadêmico no Programa de mobilidade acadêmica e a todos os outros estudantes intercambistas que ingressaram junto comigo nessa jornada e que sempre me apoiaram e proporcionaram momentos inesquecíveis. Agradeço ao CEFET-MG pelo apoio financeiro e ao IPB pela calorosa receptividade. Também agradeço a Deus por ter abençoado toda essa trajetória, permitindo que mesmo a milhares de quilômetros de casa, e diversos momentos difíceis como o da pandemia, permitiu que hoje eu esteja apresentando esse trabalho e trabalhando em uma área que amo.

Resumo

A água é um bem essencial para os ecossistemas terrestres, já que interfere diretamente na formação e na dinâmica dos solos e do clima, além de ser responsável pela manutenção da vida humana e de diversas outras espécies existentes em nosso planeta. Neste contexto, o controle da qualidade dos recursos hídricos torna-se de extrema relevância, principalmente após vários estudos correlacionarem o aumento da população, as variações climáticas e a expansão das atividades agrícolas e industriais com a diminuição da disponibilidade e da qualidade da água potável em todo o mundo.

O controle da qualidade da água pode ser realizado através de diversas técnicas, a mais utilizada é o monitoramento convencional, que geralmente é constituída por três etapas: coleta das amostras no local analisado, armazenamento e análise das mesmas, algo geralmente realizado em laboratório. Esse processo tem certas desvantagens, como a necessidade de deslocamento para os locais de coleta das amostras (geralmente distantes e de difícil acesso) e, muita das vezes, a inviabilidade de se obter uma alta amostragem de dados, algo necessário para a investigação da evolução temporal dos parâmetros analisados.

Devido a esses fatores, este trabalho tem como objetivo principal o desenvolvimento de uma solução que envolva dispositivos de baixo custo, aliado a utilização de um software, possibilitando o monitoramento remoto de cursos d'água. De forma que o seu custo possa incentivar a expansão da rede de monitoramento ambiental e a sua utilização, em ambientes que já monitorados pela forma convencional, possa ser aprimorar a amostragem de informações sobre a água, utilizado-as para um melhor gerenciamento das coletas físicas *in loco* (Para que sejam analisados os parâmetros que o protótipo não seja capaz

de informar remotamente).

A proposta inicial é de que a parte de hardware da solução seja composta por uma placa de controle Arduino, um conjunto de sensores para a medição de parâmetros físico-químicos da água e do ambiente a ser analisado, uma bateria para alimentar o sistema e um módulo de comunicação via rede GSM. Já na parte de software, será desenvolvida uma aplicação Android, que permita a visualização dos dados obtidos. O módulo de comunicação deverá enviar os dados obtidos a uma base de dados, esta que além de armazená-los, será encarregada de disponibilizá-los para a aplicação, permitindo que os usuários realizem a análise dos valores obtidos.

Palavras-chave: monitoramento remoto; microcontrolador; análise da água; aplicação android.

Abstract

Water is an essential asset for terrestrial ecosystems, as it directly interferes with the formation and dynamics of soils and climate, and is responsible for maintaining the life of humans and various other species on our planet. In this context, the control of the quality of water resources becomes extremely important, especially after several studies correlate the population increase, climatic variations and the expansion of agricultural and industrial activities with the decrease of the availability and quality of drinking water in worldwide.

Water quality control can be accomplished through several techniques, the most used is conventional monitoring, which usually consists of three steps: collection of samples at the analyzed site, storage and analysis, something generally performed in laboratory. This process has certain disadvantages, such as the need to travel to the places for collect samples (often distant and difficult to reach) and often the unfeasibility to obtain high data sampling, necessary for the investigation of evolution time of the analyzed parameters.

This work has as main objective the development of a solution that involves low cost devices, allied to the use of software, enabling the remote monitoring of water courses. The low cost of the prototype will be able to encourage the expansion of the environmental monitoring and its use, in environments that are already conventionally monitored, may improve the sampling of water information, because it can be used for better management the collection of samples *in loco* (To analyze parameters that the prototype is not able to inform remotely).

The initial proposal is that the hardware part of the solution should be composed by an Arduino control board, a set of sensors for the measurement of water and environment

physicochemical parameters, a battery to power the system and a communication module via GSM network. In the software part, an Android application will be developed, which allows the visualization of the obtained data. The communication module should send the obtained data to a database, which in addition to storing them, will be in charge of making the data available for the application, allowing users to perform the analysis of the values obtained.

Keywords: remote monitoring; microcontroller; water analysis.

Conteúdo

1	Introdução	1
1.1	Motivação	3
1.2	Problema	3
1.3	Objetivo	4
2	Contexto e Tecnologias/Ferramentas	5
2.1	A escassez de água e os impactos antrópicos	5
2.2	Agências Reguladoras	7
2.3	Comparação dos tipos de monitoramento	12
2.4	Parâmetros Físico-Químicos	14
2.4.1	pH	14
2.4.2	Condutividade	15
2.4.3	Oxigênio Dissolvido	16
2.4.4	Temperatura	17
2.4.5	Turbidez	17
2.5	Princípio de funcionamento dos componentes	18
2.5.1	Sensor de pH	18
2.5.2	Sensor de turbidez	20
2.5.3	Condutímetro	21
2.5.4	Sonda de Oxigênio dissolvido	23
2.5.5	Arduino	24

2.5.6	Redes sem fio	26
2.5.7	Smartphones, Sistema Android e Tecnologias de desenvolvimento	27
2.6	Trabalhos Relacionados	28
2.6.1	Sistema de irrigação com monitoramento remoto	28
2.6.2	Estação automática para monitoramento de compostagem	31
2.6.3	Análise automatizada de ambiente para aquicultura	33
3	Abordagem/Análise/Modelação	35
3.1	Etapas	35
3.2	Materiais	37
3.2.1	Microcontrolador	37
3.2.2	Sensor de pH	38
3.2.3	Sensor de temperatura	39
3.2.4	Sensor de condutividade	40
3.2.5	Sensor de oxigênio dissolvido	41
3.2.6	Sensor de turbidez	41
3.2.7	Módulo para padronização da data/hora	42
3.2.8	Alimentação	42
3.2.9	Transmissão de dados	44
3.3	Ferramentas	44
3.3.1	Firestore	44
3.3.2	Android Studio	45
3.3.3	Flutter	46
4	Arquitetura e Desenvolvimento	47
4.1	Hardware	48
4.1.1	Microntroncontrolador	50
4.1.2	Sensores	53
4.1.3	Demais módulos	61
4.2	Base de Dados	63

4.3	API	67
4.4	Software	73
4.4.1	FirstScreen	74
4.4.2	NewUserScreen	75
4.4.3	Homescreen	76
4.4.4	MapScreen	78
4.4.5	StationsScreen	79
4.4.6	AddStationScreen	81
4.4.7	StationProfileScreen	83
4.4.8	FavoriteScreen	85
4.4.9	AnalyseScreen	86
4.4.10	ProfileScreen	88
5	Conclusões	89

Lista de Tabelas

Lista de Figuras

1.1	Unidade de monitoramento remoto [5]	3
2.1	Mapa sobre as regiões que experimentam escassez de água pelo menos um mês do ano. [11]	6
2.2	Impactos ambientais urbanos e suas consequências as nascentes. [16]	7
2.3	Dados sobre pontos de monitoramento contidos na bacia do Rio Doce. [19]	9
2.4	Limites físico-químicos definidos segundo a deliberação conjunta COPAM/CERH-MG n01/2008. [6]	11
2.5	Exemplo da migração de íons da solução mais concentrada para a menos concentrada. [26]	19
2.6	Eletrodo de Prata/Cloreto de Prata. [26]	20
2.7	Sensor de turbidez. [28]	21
2.8	Princípio de funcionamento de um condutivímetro. [29]	22
2.9	Migração de íons em corrente contínua. [29]	22
2.10	Comportamento de íons em uma corrente alternada. [29]	22
2.11	Imagem ilustrativa do interior da sonda galvânica utilizada para mensurar o teor de Oxigênio Dissolvido. [30]	23
2.12	IDE Arduino. [32]	25
2.13	Arduino Uno Rev3. [33]	25
2.14	Protótipo montado para obtenção de dados e atuação no sistema. [42]	29
2.15	Software para o gerenciamento do equipamento. [42]	31
2.16	Protótipo montado para obtenção de dados e atuação no sistema. [43]	32

2.17	Protótipo do sistema montado e sendo testado em um tanque com peixes.	
	[45]	34
3.1	Arduino MEGA 2560. [51]	38
3.2	Eletrodo E-201c e módulo PH-4502C. [52]	38
3.3	Sensor DS18B20. [53]	39
3.4	Sensor HTU-21d. [54]	40
3.5	Kit DFRobot Analog Eletric Conductivity V2. [55]	40
3.6	Kit DFRobot Analog Dissolved Oxygen Sensor/Meter. [56]	41
3.7	Kit de medição analógica da turbidez. [57]	42
3.8	Módulo DS1307. [58]	43
3.9	Adaptador entre bateria de 9V e o padrão de alimentação do Arduino. [59]	43
3.10	Módulo SIM808. [60]	44
4.1	Diagrama geral da solução. [65]	47
4.2	Diagrama geral da parte de Hardware. [65]	49
4.3	Processo de configuração da Placa Arduino a ser utilizada, na ferramenta Arduino IDE. [65]	51
4.4	Processo de configuração da Placa Arduino a ser utilizada, na ferramenta Arduino IDE. [65]	51
4.5	Exemplos disponibilizados na ferramenta Arduino IDE. [65]	52
4.6	Botões presentes na ferramenta Arduino IDE, para verificação, compilação e execução do código no micontrolador do Arduino.	53
4.7	Tabela a ser considerada na conversão da medição em millivolts (mV) em pH. [66]	54
4.8	Comando 'enter' sendo inserido através do monitor serial. [69]	56
4.9	Comando 'cal' sendo inserido através do monitor serial. [69]	56
4.10	Comando 'exit' sendo inserido através do monitor serial. [69]	57
4.11	Tabela de relação entre a voltagem e o valor em NTU's. [70]	57

4.12	Ilustração sobre a forma de como a solução deve ser inserida na tampa da sonda. [71]	58
4.13	Gráfico com a relação do valor de tensão do ponto de saturação e a temperatura. [71]	60
4.14	Tabela contendo a relação entre o teor de oxigênio dissolvido e a temperatura do meio. [71]	60
4.15	Sequencia de comandos inseridos no monitor serial para verificar se o módulo de comunicação está funcionando corretamente. [75]	63
4.16	Diagrama geral da Base de Dados. [65]	65
4.17	Módulos da API. [65]	68
4.18	Script Entity. [65]	69
4.19	Script Controller. [65]	70
4.20	Conexão com a base de dados. [65]	71
4.21	Script Service. [65]	72
4.22	Diagrama geral da aplicação. [65]	73
4.23	Primeira tela da aplicação. [65]	74
4.24	Tela para cadastro de um novo usuário. [65]	75
4.25	Tela de exibição da situação das estações criadas/favoritadas pelo usuário. [65]	77
4.26	Tela de exibição das estações próximas ao usuário, através de um mapa. [65]	78
4.27	Tela de exibição da lista de todas as estações registradas. [65]	79
4.28	Tela de exibição da lista de estações criadas pelo usuário. [65]	80
4.29	Tela de adição de uma nova estação. [65]	82
4.30	Tela de visualização geral da estação escolhida. [65]	84
4.31	Tela de exibição das estações favoritadas pelo usuário. [65]	85
4.32	Tela de análise dos dados registrados pela estação escolhida. [65]	87
4.33	Tela de exibição dos dados do usuário. [65]	88

Capítulo 1

Introdução

A qualidade da água é resultado de vários processos naturais e humanos agindo conjuntamente, aliado às interações entre eles no espaço e no tempo. Atualmente, os processos que são responsáveis pelos maiores impactos nos ecossistemas aquáticos são os de eutrofização e assoreamento, estes que, são acelerados por atividades antrópicas, como a exposição de solos para agricultura, agropecuária, mineração ou para urbanização [1]. Esse gradativo aumento na concentração de matéria orgânica traz grandes prejuízos aos ecossistemas aquáticos, incluindo a maioria dos cursos d'água no Brasil, algo que acaba afetando diretamente a vida aquática, bem como a utilização deste recurso pelos seres humanos. Essa crescente deterioração da qualidade das águas, tem levado à busca de estratégias para proteção e manutenção da qualidade dos ecossistemas aquáticos.

Para que o cenário dos ecossistemas aquáticos seja avaliado, um dos procedimentos mais utilizados é o de monitoramento. Artiola et al. [2] definem monitoramento como observação e estudo do meio ambiente. Em termos científicos, os dados são coletados para gerar conhecimento. Os pesquisadores [2] ainda discorrem que as informações geradas por meio de atividades de monitoramento podem ser utilizadas de diversas formas, desde o entendimento sobre a alta na taxa de mortalidade de determinada espécie de peixe em extinção em um pequeno córrego, até a definição de estratégias de longo período para a preservação de grandes áreas que contém muitos recursos naturais. O monitoramento ambiental é um tema cada vez mais relevante devido ao padrão de consumo e o aumento

da população, principalmente nos grandes centros urbanos, algo que promove a utilização cada vez maior dos recursos ambientais [3].

Dentre as tendências mais recentes de monitoramento dos ecossistemas aquáticos, destaca-se o monitoramento remoto em tempo real [4]. Essa técnica reúne várias tecnologias, tais como: uso de sensores de alta qualidade para medidas físicas, químicas e biológicas da água com medições climatológicas, armazenamento de dados e transmissão destes por meio de telefonia celular (GPRS), via satélite ou por via de radiofrequência.

Devido o monitoramento remoto possibilitar um monitoramento intensivo, ou seja, com obtenção de dados contínuos em intervalos reduzidos de tempo, ele se constitui como uma ferramenta importante na gestão dos recursos hídricos. Além de proporcionar a identificação imediata de alterações bruscas na qualidade da água, o monitoramento remoto também contribui com uma amostragem maior de dados, estes que, auxiliam diretamente no estudo e na escolha de medidas de manejo e recuperação ambiental. Essas medidas são fundamentais, já que visam garantir a integridade dos recursos naturais e consequentemente sua exploração sustentada, com o acréscimo de economia de recursos financeiros.

Porém o alto custo de implantação e manutenção são um dos maiores limitadores para a implantação das unidades de monitoramento remoto em massa. Em 2015, por exemplo, o governo do estado brasileiro da Bahia [5] investiu cerca de R\$ 1.741.330,92 (Aprox. \$ 420.000) na aquisição de novas unidades de monitoramento remoto (UMR's) - Uma delas representada pela Figura 1.1 - em seis reservatórios contidos dentro do território do estado. Esse valor certamente inviabiliza a compra desses equipamentos por boa parte dos municípios, institutos governamentais, organizações não governamentais, empresas, entre outros. Algo que acaba prejudicando o início da aplicação desse tipo de monitoramento em diversos cursos d'água contidos em todo o país.

No entanto foi proposto estruturar um protótipo, com um custo menor que as UMR's analisadas, que seja capaz de enviar dados para a realização de um monitoramento remoto da qualidade da água. De forma que seu menor custo possa incentivar a expansão da rede de monitoramento ambiental (principalmente em córregos e nascentes) e que sua capacidade de enviar dados remotos sobre o ambiente analisado, possa ser uma alternativa



Figura 1.1: Unidade de monitoramento remoto [5]

que proporcione um melhor gerenciamento da aplicação do método de monitoramento tradicional.

1.1 Motivação

Grande parte das nascentes, córregos e rios situados no Brasil não possuem um monitoramento intensivo, sendo que boa parte deles sequer é monitorado, algo que deixa esses locais cada vez mais vulneráveis a degradação causada por atividades antrópicas. A motivação deste trabalho é poder contribuir simultaneamente com o meio ambiente e a sociedade que nele o vive, promovendo o desenvolvimento de uma solução que, caso implantada, possibilite o monitoramento remoto em nascentes e córregos - estes que são os mais susceptíveis as atividades antrópicas - possuindo um custo e um porte menor que as UMR's utilizadas em reservatórios brasileiros, como a utilizada pelo INEMA [5] na Bahia.

1.2 Problema

Segundo informações fornecidas pelo site do IGAM - Instituto Mineiro de Gestão das Águas [6], o estado brasileiro de Minas Gerais conta com 600 estações de amostragem

distribuídas nas bacias hidrográficas dos rios São Francisco, Grande, Doce, Paranaíba, Paraíba do Sul, Mucuri, Jequitinhonha, Pardo, Buranhém, Itapemirim, Itabapoana, Itanhém, Itaúnas, Jucuruçu, Peruípe, São Mateus e Piracicaba/Jáguari. Porém, na maior parte dessas estações as campanhas de amostragem são trimestrais, com um total anual de quatro campanhas, um intervalo relativamente grande para detecção de alterações bruscas no meio onde a análise é realizada.

Outro fator importante é que, segundo relatórios sobre a qualidade das águas publicados pelo IGAM [6], o colar metropolitano do Vale do Aço (composto por 24 municípios) possui apenas estações de amostragem em áreas de rios (Rio Doce, Rio Piracicaba, Rio Santa Bárbara, Rio da Prata, Rio Caratinga e Rio Sacramento). Ou seja, outros tipos de corpos d'água, como nascentes ou córregos, não são abrangidos, algo que promove uma possibilidade de menor entendimento dos processos ambientais que ocorrem nas diversas nascentes e córregos que cortam essa extensa região.

1.3 Objetivo

O objetivo principal deste trabalho consiste em estruturar uma solução que na parte de hardware, utilize uma placa de controle Arduino em conjunto com módulos e sensores, possibilitando a obtenção de dados físico-químicos do curso d'água em que o mesmo estiver localizado. Já na parte de software, será desenvolvida uma aplicação, que possibilite que os dados obtidos possam ser analisados. No final do desenvolvimento, a solução deverá fornecer aos usuários, de forma remota, os valores simulados de algumas variáveis (Definidas pela ANA - Agência Nacional de Águas) [7] que interferem diretamente no cálculo do índice de qualidade da água e que são capazes de demonstrar o teor de pureza da água, como: pH, temperatura, turbidez e a condutividade do meio no qual o equipamento estiver presente.

Capítulo 2

Contexto e Tecnologias/Ferramentas

2.1 A escassez de água e os impactos antrópicos

Nas últimas décadas, o problema da escassez de água se agravou cada vez mais ano a ano, ameaçando a saúde humana e a sobrevivência de diversas espécies [8] [9]. Conforme mostrado pela Figura 2.1, grande parcela da população mundial, cerca de 40%, já foi afetada pela escassez ou falta de água, seja por razões políticas, econômicas e/ou climáticas [10] [11]. E, aproximadamente, 25% da população mundial já teve problemas de saúde ou de higiene, devido à falta de acesso a água potável e saneamento básico. Alguns países da África, Ásia e América Latina são os que mais sofrem por esse cenário [12].

Uma das principais causas para a poluição dos mananciais são as atividades antrópicas, como agricultura, agropecuária, mineração, industrial e urbanização [1]. Dentre essas atividades, destacam-se às atividades industriais que, principalmente nos países subdesenvolvidos, geram elevadas quantidades de resíduos e efluentes que, muitas vezes, são despejados diretamente em cursos naturais [12]. Nos últimos cinco anos, o estado de Minas Gerais presenciou dois, dos principais desastres ambientais provocados pelo homem no Brasil. Os rompimentos das barragens de Fundão (Em Mariana) e do Córrego do Feijão (Em Brumadinho), foram responsáveis pelo despejo de 51 milhões de metros cúbicos de rejeitos [13] no meio ambiente, contaminando a maior parte da bacia hidrográfica do

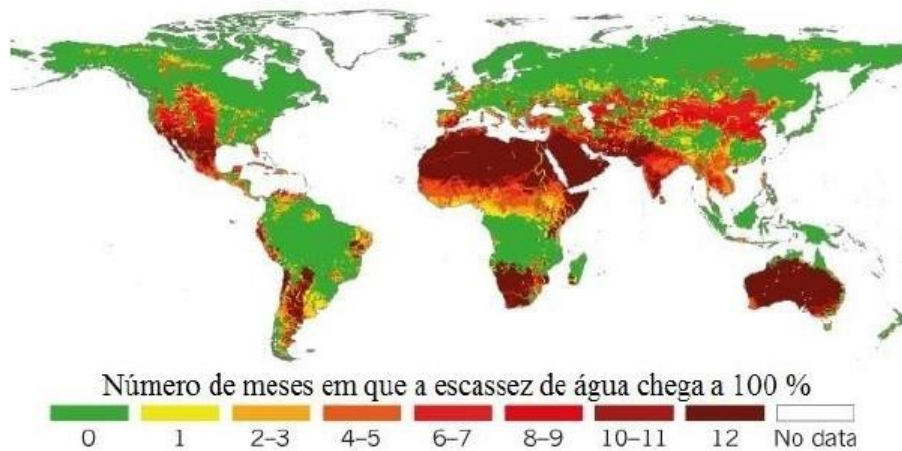


Figura 2.1: Mapa sobre as regiões que experimentam escassez de água pelo menos um mês do ano. [11]

Rio Doce - A maior da região sudeste do país [14], e uma parte significativa da bacia hidrográfica do Rio Paraopeba [15].

Conforme demonstrado na figura 2.2, Felipe [16] cita os impactos das atividades urbanas sobre as nascentes. Em seu trabalho, o autor demonstra como essa situação é alarmante, já que as nascentes são facilmente degradadas pelas atividades humanas mais comuns em ambientes urbanos.

IMPACTOS	CONSEQUÊNCIAS GERAIS NO SISTEMA HÍDRICO	CONSEQUÊNCIAS PARA AS NASCENTES
Impermeabilização do solo	<ul style="list-style-type: none"> • Aumento da quantidade e da velocidade do escoamento superficial. • Redução da recarga de aquíferos. • Intensificação dos processos erosivos, aumento da carga sedimentar para os cursos d'água, assoreamento e inundações. 	<ul style="list-style-type: none"> • Descaracterização. • Redução da vazão. • Desaparecimento.
Resíduos (combustível, esgoto, lixões, etc.)	<ul style="list-style-type: none"> • Poluição das águas subterrâneas. 	<ul style="list-style-type: none"> • Redução na qualidade da água.
Retirada de água subterrânea	<ul style="list-style-type: none"> • Rebaixamento do nível freático. 	<ul style="list-style-type: none"> • Redução da vazão. • Desaparecimento.
Substituição da cobertura vegetal	<ul style="list-style-type: none"> • Intensificação dos processos erosivos, assoreamento, inundações. • Diminuição da retenção de água. • Aumento da energia dos fluxos superficiais. 	<ul style="list-style-type: none"> • Descaracterização. • Redução da vazão. • Desaparecimento.
Construções	<ul style="list-style-type: none"> • Drenagem de nascentes. • Aterramento. 	<ul style="list-style-type: none"> • Descaracterização. • Desaparecimento.
Canalização de rios	<ul style="list-style-type: none"> • Aumento da velocidade e da energia dos fluxos. • Alteração no padrão de influência/efluência dos rios. 	<ul style="list-style-type: none"> • Descaracterização. • Redução da vazão.
Ilha de calor	<ul style="list-style-type: none"> • Alteração no padrão de chuvas. • Alteração no padrão de recarga. 	<ul style="list-style-type: none"> • Alteração da vazão.

Figura 2.2: Impactos ambientais urbanos e suas consequências as nascentes. [16]

2.2 Agências Reguladoras

Para que os impactos humanos sobre o meio ambiente possam ser minimizados e fiscalizados foram criadas agências reguladoras. No âmbito federal brasileiro, existe a ANA - Agência Nacional das Águas, autarquia vinculada ao Ministério do Meio Ambiente, que é responsável pela implantação da Política Nacional de Recursos Hídricos. Dentre os objetivos dessa agência está o propósito de criar as condições para a implantação das orientações técnicas emanadas pela Lei das Águas (Lei 9.433, de 08 de janeiro de 2001)

[17], nas diferentes regiões do país.

Essa agência também regula o acesso e o uso dos recursos hídricos de domínio da União - os que fazem fronteiras com outros países ou passam por mais de um estado, emite e fiscaliza o cumprimento de normas - em especial as outorgas, além de ser responsável por acompanhar a situação dos recursos hídricos do Brasil. A ANA também coordena a Rede Hidrometeorológica Nacional que capta, com o apoio dos estados e outros parceiros, informações como nível, vazão e sedimentos dos rios ou quantidade de chuvas. Essas informações servem para planejar o uso da água e prevenir eventos críticos, como secas e inundações [7].

No âmbito estadual de Minas Gerais, existe o IGAM - Instituto Mineiro de Gestão das Águas, que é o responsável por planejar e promover ações direcionadas à preservação da quantidade e da qualidade dos recursos hídricos de Minas Gerais. O gerenciamento é feito com base nas diretrizes do Plano Estadual de Recursos Hídricos e dos Planos Diretores de Recursos Hídricos, definidos pela ANA.

Além disso, o IGAM é responsável por implantar metodologias que orientam a concessão de outorga de direito de uso da água, pelo monitoramento da qualidade das águas superficiais e subterrâneas do Estado, por pesquisas, programas e projetos e por disseminar informações consistentes sobre recursos hídricos, bem como pela consolidação de Comitês de Bacias Hidrográficas – CBHs e Agências de Bacias (Decreto 47343, de 23/01/2018) [18].

Conforme informado pelos relatórios de monitoramento [19], o monitoramento da qualidade das águas na bacia hidrográfica do Rio Doce é realizado pelo IGAM, por meio do Programa Águas de Minas, desde o ano de 1997. Este monitoramento contempla atualmente 64 pontos de amostragem de água, onde são realizadas coletas e análises laboratoriais com periodicidade trimestral e avaliação de aproximadamente 50 parâmetros físico-químicos e hidrobiológicos. A figura 2.3 mostra informações sobre alguns desses pontos.

Estações	Município	Descrição	Latitude (S)	Longitude (W)
RD011	Barra Longa	Rio Gualaxo do Norte próximo de sua foz no rio do Carmo, município de Barra Longa	20° 17' 9,81"	43° 3' 57,02"
RD071	Barra Longa	Rio do Carmo em Barra Longa	20° 16' 58,0"	43° 01' 56,00"
RD072	Rio Doce	Rio Doce no município de Rio Doce	19° 14' 51,00"	42° 53' 07,00"
RD019	Rio casa e São Domingos do Prata	Rio Doce entre os municípios de Rio Casca e São Domingos do Prata	20° 0,1' 18,0"	42° 45' 0,80"
RD023	Marliéria e Pingo d'Água	Rio Doce entre os municípios de Marliéria e Pingo D'Água	19° 45' 35"	42° 29' 0,600"
RD033	Belo Oriente	Rio Doce no município de Belo Oriente	19° 19' 38,93"	42° 22' 32,97"
RD035	Ipatinga	Rio Doce no município de Ipatinga	19° 29' 18,99"	42° 29' 38,99"
RD083	Periquito	Rio Doce logo a jusante do município de Periquito	19° 5' 42,19"	42° 9' 17,1"
RD044	Governador Valadares	Rio Doce na cidade de Governador Valadares	18° 53' 0"	41° 57' 10,00"
RD045	Governador Valadares	Rio Doce no município de Conselheiro Pena	18° 51' 36,19"	41° 50' 1,35"
RD053	Tumiritinga	Rio Doce no município de Tumiritinga	18° 58' 10,19"	41° 38' 49,39"
RD058	Conselheiro Pena	Rio Doce no município de Conselheiro Pena	19° 9' 58,84"	41° 27' 35,69"
RD059	Resplendor	Rio Doce no município de Resplendor	19° 20' 45,71"	41° 14' 19,49"
RD067	Aimorés	Rio Doce no município de Aimorés	19° 30' 20,00"	41° 0' 47,00"

Figura 2.3: Dados sobre pontos de monitoramento contidos na bacia do Rio Doce. [19]

Para a avaliação da qualidade águas superficiais, o IGAM realiza medições em cima dos seguintes parâmetros: condutividade elétrica, oxigênio dissolvido, pH, temperatura, sólidos totais, sólidos dissolvidos totais, sólidos em suspensão totais, turbidez e arsênio total, bem como os metais: alumínio dissolvido, ferro dissolvido, cobre dissolvido, cromo total, cádmio total, chumbo total manganês total, mercúrio total e níquel total [19]. Além de definirem uma tabela (seguindo a deliberação conjunta COPAM/CERH-MG n 01/2008) contendo os limites que cada parâmetro pode ter nas amostras analisadas (Conforme a figura 2.4).

Parâmetros	LIMITE DN COPAM/CERH-MG – 01/2008 - Classe 2
Alumínio dissolvido	0,1 mg/L
Arsênio total	0,01 mg/L
Cádmio total	0,001mg/L
Chumbo total	0,01 mg/L
Cobre dissolvido	0,009 mg/L
Cromo total	0,05 mg/L
Ferro dissolvido	0,3 mg/L
Manganês total	0,1 mg/L
Merúrio total	0,2 µg/L
Níquel total	0,025 mg/L
Oxigênio dissolvido	> 5 mg/L
pH <i>in loco</i>	6 a 9
Sólidos em suspensão totais	100 mg/L
Sólidos dissolvidos totais (mg/L)	500 mg/L
Turbidez	100 NTU
Zinco total	0,18 mg/L

Figura 2.4: Limites físico-químicos definidos segundo a deliberação conjunta COPAM/CERH-MG n01/2008. [6]

Na tabela disponibilizada não é informado o limite da condutividade, porém segundo informado pela Funasa - Fundação Nacional da Saúde [20], as águas naturais apresentam teores de condutividade na faixa de 10 a 100 µS/cm e ambientes poluídos por esgotos domésticos ou industriais os valores podem chegar a 1.000 µS/cm. Os ambientes aquáticos brasileiros apresentam, em geral, temperaturas na faixa de 20°C a 30°C. Em relação às águas para consumo humano, temperaturas elevadas aumentam as perspectivas de rejeição ao uso.

2.3 Comparação dos tipos de monitoramento

Glasgow et al. [21] apresenta uma comparação resumida das vantagens e desvantagens do uso das diversas formas de monitoramento, como o monitoramento tradicional, sensoriamento remoto e o monitoramento remoto em tempo real. O monitoramento tradicional compreende o modo de coleta em campo com barco, garrafa de van Dorn, disco de Secchi, redes de zoo e fitoplâncton, dragas para coleta de sedimentos, sondas e outros equipamentos necessários, dependendo do tipo de análise a ser realizada. Abaixo está um resumo feito por Pompêo et al. [4] sobre o levantamento realizado por Glasgow.

- Monitoramento tradicional:
 - Vantagens:
 - * Menor desgaste dos equipamentos, por não permanecerem em ambientes agressivos;
 - * Procedimentos e técnicas consolidados.
 - * Desvantagens:
 - * Monitoramento dificultado em períodos noturnos;
 - * Menor quantidade de dados coletados;
 - * Maior tempo despendido para obtenção do dado e posterior análise.

- Sensoriamento remoto:
 - Vantagens:
 - * Tecnologia pode ser aplicada em diversos ramos, como em mapeamentos geológicos e manejo de recursos florestais e oceanográficos.

- * Desvantagens:

- * Dificuldade em obtenção de informações diretas abaixo da superfície;
- * Dados podem ser afetados de acordo com meteorologia;
- * Não há contato dos sensores com o corpo d'água (as medições são feitas à distância).

- Monitoramento em tempo real:

- Vantagens:

- * Obtenção de dados contínuos em intervalos reduzidos de tempo;
- * Detecção de alterações bruscas em tempo real, possibilitando a manutenção da qualidade da água;
- * Possibilidade de alarmes;
- * Possibilidade de maior entendimento dos processos ambientais.

- * Desvantagens:

- * Dificuldade em obtenção de informações diretas abaixo da superfície;
- * Ambiente agressivo reduz o tempo de vida dos equipamentos;
- * Calibração e manutenção dificultadas;
- * Necessidade de análise paralela da consistência dos dados gerados.

2.4 Parâmetros Físico-Químicos

2.4.1 pH

Conforme apresentado pelo manual de controle da qualidade da água para técnicos [20] o potencial hidrogênio (pH) representa a intensidade das condições ácidas ou alcalinas do meio líquido, por meio da medição da presença de íons hidrogênio (H^+). É calculado em escala antilogarítmica, abrangendo a faixa de 0 a 14 (Valores inferiores a 7 representam condições ácidas e valores superiores a 7 representam condições básicas/alcalinas). O valor do pH influi na distribuição das formas livre e ionizada de diversos compostos químicos, além de contribuir para um maior ou menor grau de solubilidade das substâncias e de definir o potencial de toxicidade de vários elementos. As alterações de pH podem ter origem natural (dissolução de rochas, fotossíntese) ou antropogênica (despejos domésticos e industriais). Para a adequada manutenção da vida aquática, o pH deve situar-se, geralmente, na faixa de 6 a 9.

A acidificação das águas pode ser um fenômeno derivado da poluição atmosférica, mediante complexação de gases poluentes com o vapor d'água, provocando o predomínio de precipitações. A origem da acidez tanto pode ser natural (CO_2 absorvido da atmosfera, ou resultante da decomposição de matéria orgânica, gerando presença de H_2S – gás sulfídrico) como por atividades antrópicas (como despejos industriais, urbanos ou passagem da água por minas abandonadas, por exemplo). A distribuição das formas de acidez também é função do pH da água:

- pH > 8.2: CO_2 ausente;
- pH entre 4.5 e 8.2: Acidez carbônica;
- pH < 4.5: Acidez por ácidos minerais fortes, geralmente resultantes de despejos industriais.

Já ambientes aquáticos com altos valores de alcalinidade podem manter aproximadamente os mesmos teores de pH, mesmo com o recebimento de contribuições fortemente ácidas

ou alcalinas. Os principais constituintes da alcalinidade são os bicarbonatos (HCO_3^-), carbonatos (CO_3^{2+}) e hidróxidos (OH^-). Outros ânions, como cloretos, nitratos e sulfatos, não contribuem para a alcalinidade. Valores elevados de alcalinidade estão associados a processos de decomposição da matéria orgânica e à alta taxa respiratória de microorganismos, com liberação e dissolução do gás carbônico (CO_2) na água. A distribuição entre as três formas de alcalinidade na água (bicarbonatos, carbonatos e hidróxidos) é função do seu pH:

- pH > 9.4: Hidróxidos e carbonatos;
- pH entre 8.3 e 9.4: Carbonatos e bicarbonatos;
- pH < 4.5: Apenas bicarbonatos.

2.4.2 Condutividade

A condutividade elétrica da água indica a sua capacidade de transmitir a corrente elétrica em função da presença de substâncias dissolvidas, que se dissociam em ânions e cátions. Essa propriedade varia com a concentração total de substâncias ionizadas dissolvidas na água, com a temperatura, com a mobilidade dos íons, com a valência dos íons e com as concentrações real e relativa de cada íon [22]. Muito embora não se possa esperar uma relação direta entre condutividade e concentração de sólidos totais dissolvidos, já que as águas naturais não são soluções simples, tal correlação é possível para águas de determinadas regiões onde exista a predominância bem definida de um determinado íon em solução. A condutividade elétrica da água deve ser expressa em unidades de resistência (mho ou S) por unidade de comprimento (geralmente cm ou m). Até algum tempo atrás, a unidade mais usual para expressão da resistência elétrica da água era o mho (inverso de ohm), mas atualmente é recomendável a utilização da unidade S (Siemens). Enquanto que as águas naturais apresentam teores de condutividade na faixa de 10 a 100 $\mu S/cm$, em ambientes poluídos por esgotos domésticos ou industriais os valores podem chegar a 1.000 $\mu S/cm$ [20].

2.4.3 Oxigênio Dissolvido

O oxigênio dissolvido é o nível de oxigênio livre não composto presente na água ou em outros líquidos. O oxigênio não composto, ou oxigênio livre (O_2), é o oxigênio que não está ligado a nenhum outro elemento. Ele é um parâmetro importante na avaliação da qualidade da água, devido à sua influência nos organismos que vivem dentro de um corpo d'água e também pela possibilidade de indicar a existência de alta decomposição no meio (Fator diretamente relacionado a impactos causados pelo ser humano).

O oxigênio dissolvido entra na água pelo ar ou como um subproduto da planta. Do ar, o oxigênio pode se difundir através da superfície da água a partir da atmosfera, ou ser misturado rapidamente por meio de aeração, seja natural ou artificial, como por exemplo em cachoeiras ou represas. Das plantas o oxigênio pode surgir da fotossíntese do fitoplâncton, algas, algas marinhas e outras plantas aquáticas.

Em equilíbrio, a porcentagem de cada gás na água seria equivalente à porcentagem daquele gás na atmosfera, pois a água irá absorver lentamente o oxigênio e outros gases da atmosfera até atingir o equilíbrio na saturação completa. Este processo pode ser acelerado pelas ondas do vento e outras fontes de aeração.

A solubilidade do oxigênio diminui com o aumento da temperatura, pois as águas superficiais mais quentes requerem menos oxigênio dissolvido para atingir 100% de saturação de ar do que as águas mais profundas e frias. O oxigênio dissolvido também diminui exponencialmente conforme os níveis de salinidade da água aumentam. É por isso que, à mesma pressão e temperatura, a água salgada retém cerca de 20% menos oxigênio dissolvido do que a água doce. A água em altitudes mais baixas pode conter mais oxigênio dissolvido do que a água em altitudes mais altas. Essa relação ocorre pois em pressões hidrostáticas maiores, a água pode reter mais oxigênio dissolvido sem que ele escape [23].

Outro fator importante é relacionado a decomposição bacteriana no ambiente aquático, quando ela aumenta ocorre o consumo da maior parte ou todo o oxigênio dissolvido disponível no meio. Isso cria um ambiente anóxico, ou pobre em oxigênio, onde peixes e outros organismos não podem sobreviver. Esses níveis de nutrientes podem ocorrer

naturalmente, mas são mais frequentemente causados pela poluição do escoamento de fertilizantes ou águas residuais mal tratadas.

2.4.4 Temperatura

A temperatura é uma medida da energia interna de um sistema, que tem como objetivo mensurar o quão quente ou frio um determinado objeto está. Uma das escalas mais usadas é o grau Celsius ($^{\circ}\text{C}$) [22]. Elevações da temperatura aumentam a taxa das reações físicas, químicas e biológicas (na faixa usual de temperatura), diminuem a solubilidade (Como o oxigênio dissolvido) e também aumentam a taxa de transferência de gases (O que pode gerar mau cheiro, no caso da libertação de gases com odores desagradáveis) [24].

Variações de temperatura são parte do regime climático normal, onde corpos de água naturais apresentam variações sazonais e diurnas, bem como estratificação vertical. A temperatura superficial é influenciada por diversos fatores, tais como: latitude, altitude, estação do ano, período do dia, vazão e profundidade. A elevação da temperatura em um corpo d'água geralmente é provocada por despejos industriais (indústrias canavieiras, por exemplo) e usinas termoeletricas [25].

2.4.5 Turbidez

A turbidez pode ser definida como uma medida do grau de fótons refletidos através das partículas suspensas, essa medida é expressa por meio de unidades de turbidez (também denominadas unidades nefelométricas). A turbidez dos corpos d'água é particularmente alta em regiões com solos erosivos, onde a precipitação pluviométrica pode carrear partículas de argila, silte, areia, fragmentos de rocha e óxidos metálicos do solo. Grande parte das águas de rios brasileiros é naturalmente turva em decorrência das características geológicas das bacias de drenagem, ocorrência de altos índices pluviométricos e uso de práticas agrícolas, muitas vezes inadequadas. Ao contrário da cor, que é causada por substâncias dissolvidas, a turbidez é provocada por partículas em suspensão, sendo, portanto, reduzida por sedimentação. Em lagos e represas, onde a velocidade de escoamento da

água é menor, a turbidez pode ser bastante baixa. Além da ocorrência de origem natural, a turbidez da água pode, também, ser causada por lançamentos de esgotos domésticos ou industriais [20].

2.5 Princípio de funcionamento dos componentes

2.5.1 Sensor de pH

O pH é determinado por potenciometria, ou seja, através da diferença de potencial entre dois eletrodos - Que serão imersos na amostra a ser analisada (Equações 1 e 2). A célula galvânica utilizada para medição de pH, emprega basicamente dois elementos que geram um potencial elétrico proporcional à concentração de íons de hidrogênio livres na amostra (Equação 3).

$$\Delta E = E_{Ref} + E_{Amostra} \text{ (Equação 1)}$$

$$E = E^\circ - \frac{RT}{nF} \log \frac{[produto]}{[reagente]} \text{ (Equação 2)}$$

$$\Delta E \propto pH \text{ (Equação 3)}$$

O eletrodo de medição do pH, é composto por uma fina membrana de vidro sensível ao pH selada na ponta de um tubo de vidro ou de plástico. Nessa membrana os íons de H^+ vão se acumulando, formando uma diferença de concentração iônica entre a solução da amostra e a solução de referência, de concentração constante. Essa diferença iônica provoca a geração de uma tensão no eletrodo (Conforme mostrado pela Figura 2.5).

Quando o eletrodo de medição é imerso na solução contendo o analito, ele desenvolve um potencial, que depende da atividade do analito. A maioria dos eletrodos de medição empregados na potenciometria é seletiva em sua resposta. O eletrodo de referência é o eletrodo em relação ao qual se mede o potencial do eletrodo de medição. O seu potencial é conhecido, constante e completamente independente da composição da solução de analito. O potencial de um eletrodo de referência deve ser independente das propriedades da solução e, se possível, deve ser fisicamente isolado a partir da solução de interesse [27].

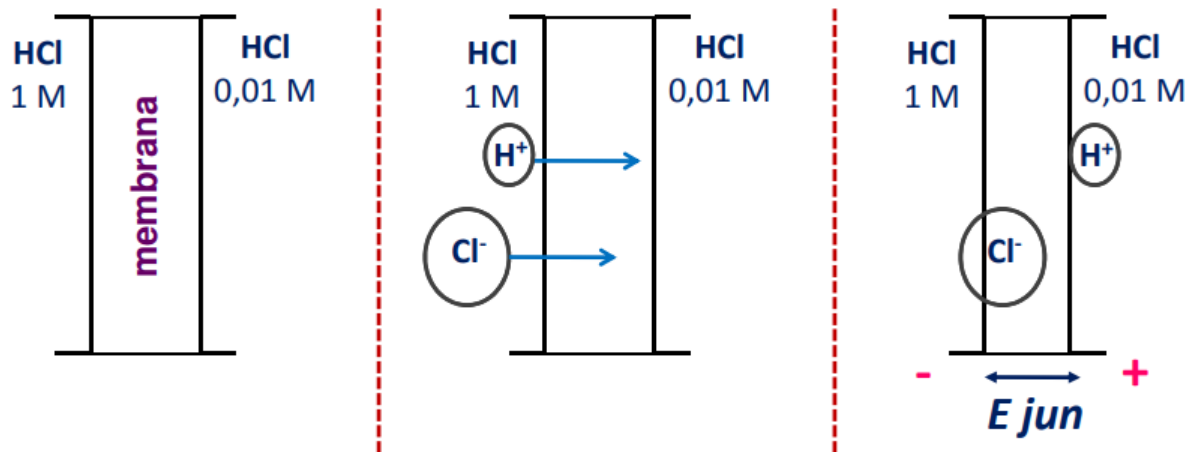
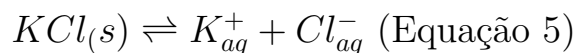
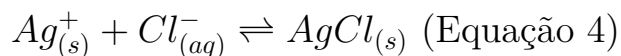


Figura 2.5: Exemplo da migração de íons da solução mais concentrada para a menos concentrada. [26]

Um dos eletrodos mais utilizados é o de Prata/Cloreto de Prata ($AgCl$), que contém além da membrana de vidro e do eletrólito, um fio de Prata (Ag), um fio de Prata/Cloreto de Prata ($Ag/AgCl$) e a solução de KCl (Figura 2.6). Dentro do eletrodo ocorrem reações de forma que ocorram apenas reações de equilíbrio, (conforme mostrado pelas equações 4 e 5) e assim a concentração de Cl^- seja constante.



Com a concentração constante de Cl^- , o potencial do eletrodo de referência 22 ($E_{(Ref)}$) assume um valor constante (A temperatura de $25^\circ C$) e permite com que o valor do potencial da amostra seja encontrado (Conforme equação 1 e 2).

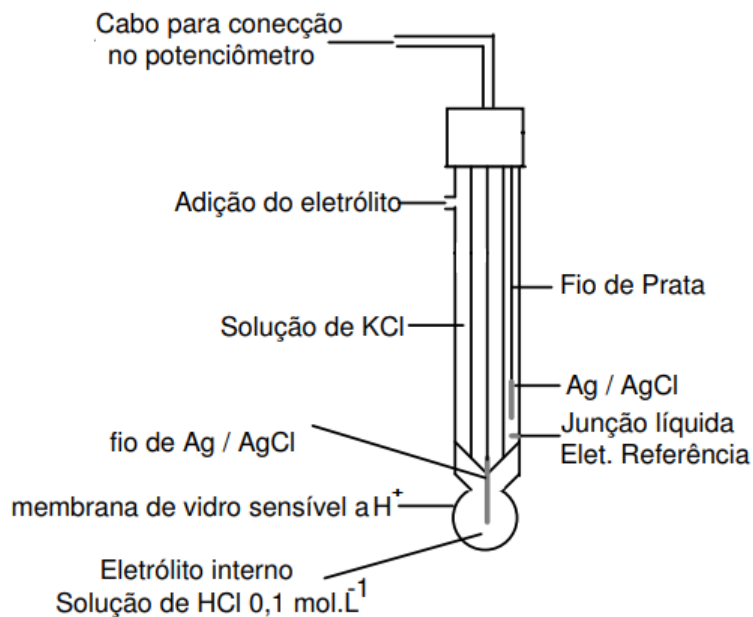


Figura 2.6: Eletrodo de Prata/Cloreto de Prata. [26]

2.5.2 Sensor de turbidez

O sensor de turbidez possui um emissor e um receptor de luz infravermelha, um em cada lado, conforme ilustra a Figura 2.7. Ao se emitir luz infravermelha, o sensor mede qual foi o nível de recepção da mesma no receptor, localizado logo à frente do emissor (em um ângulo de 180° com o mesmo). Quanto maior a quantidade de partículas em suspensão na água, maior é a absorção e reflexão da luz infravermelha por parte do líquido e, portanto, mais turva a água tende a ser, elevando-se assim a turbidez da água. No sensor de turbidez, tal nível de luz infravermelha capturado é manifestado na forma de uma tensão de saída, variante de 0 a 5 V, sendo esta tensão de saída lida na placa de interface e condicionamento de sinal do sensor [28].



Figura 2.7: Sensor de turbidez. [28]

2.5.3 Condutivímetro

Um condutivímetro consiste de um par de eletrodos, nos quais uma voltagem é aplicada. De acordo com a Lei de Ohm, a voltagem (V) aplicada através de uma solução é proporcional a corrente (I). Essa proporcionalidade, definida como resistência (R), pode ser calculada com o fluxo de corrente (I) medido, quando uma tensão conhecida é aplicada (Equações 6 e 7).

$$V = R I \text{ (Equação 6)}$$

$$R = \frac{V}{I} \text{ (Equação 7)}$$

A condutância (G) é definida como o inverso da resistência (R) (Equação 8).

$$G = \frac{1}{R} \text{ (Equação 8)}$$

Com a voltagem aplicada, o condutivímetro mede a corrente de fluxo e calcula a condutância (Figura 2.8).

Um detalhe importante é que quando uma corrente contínua (DC) é aplicada nos eletrodos, os íons positivamente carregados (cátions) migram para o eletrodo negativamente carregado (cátodo) e os íons carregados negativamente (ânions) se movem em direção ao

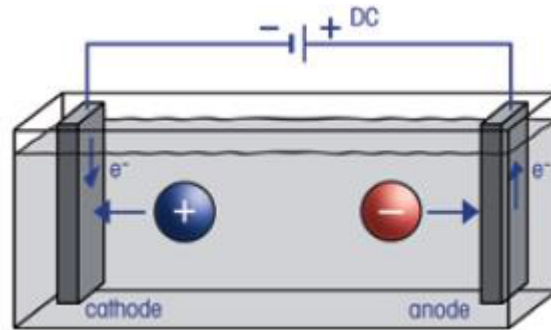


Figura 2.8: Princípio de funcionamento de um condutímetro. [29]

eletrodo carregado positivamente (ânodo) (Figura 2.9). Isso pode levar a um acúmulo de íons perto das superfícies do eletrodo, fazendo com que se ocorra uma polarização que é responsável por causar algumas reações químicas. Esta eletrólise afeta a composição da solução e, portanto, também a condutividade.

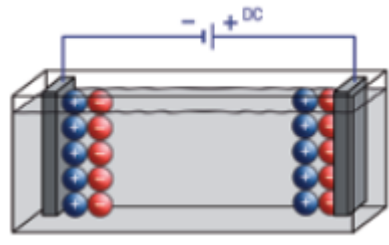


Figura 2.9: Migração de íons em corrente contínua. [29]

Para evitar as indesejáveis reações de polarização e eletrólise, é utilizada a corrente alternada (AC) para a medição de condutividade. Com a corrente alternada os íons não migram em uma direção e sim oscilam sobre suas posições com o ritmo da frequência aplicada (conforme a Figura 2.11) [29].

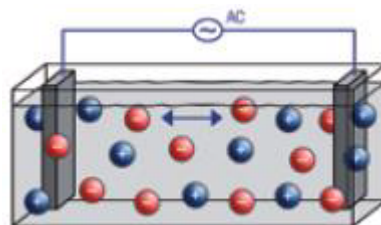


Figura 2.10: Comportamento de íons em uma corrente alternada. [29]

2.5.4 Sonda de Oxigênio dissolvido

Conforme mostrado na figura 2.11, a sonda de oxigênio dissolvido inclui uma membrana de polietileno, um cátodo (Ag) e um ânodo (Zn) imerso em um eletrólito. Para reduzir o oxigênio sem um potencial externo aplicado, a diferença de potencial entre o ânodo e o cátodo deve ser de pelo menos 0,5V. O cátodo aceita elétrons do ânodo por meio de um circuito interno e os passa para as moléculas de oxigênio, dessa forma, o ânodo é oxidado e o oxigênio é reduzido na superfície do cátodo [30].

Quando colocados em uma solução eletrolítica, o potencial entre metais diferentes faz com que eles se autopolarizem, com os elétrons viajando internamente do ânodo para o cátodo. As moléculas de oxigênio que se difundem através da membrana a uma taxa constante são reduzidas no cátodo e uma voltagem é produzida. Se não houver moléculas de oxigênio, o valor medido pela sonda será 0 mV. A medida que o oxigênio dissolvido aumenta, a medição de saída da sonda também aumentará [31].

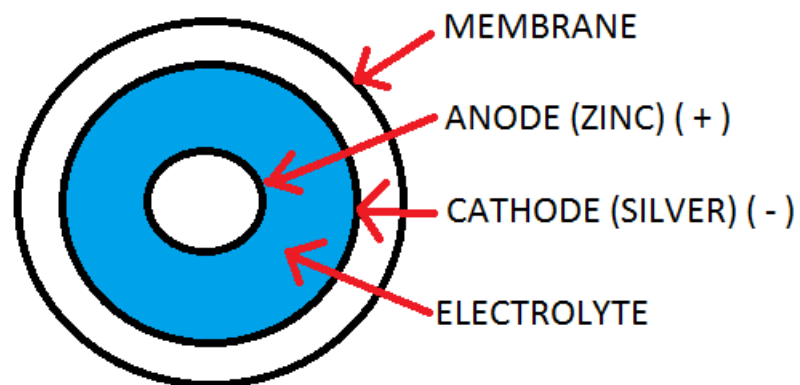


Figura 2.11: Imagem ilustrativa do interior da sonda galvânica utilizada para mensurar o teor de Oxigênio Dissolvido. [30]

2.5.5 Arduino

O Arduino é uma plataforma de prototipagem eletrônica de hardware e softwares livres concebida no Interaction Design Institute Ivrea, no norte da Itália, por um grupo formado por Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis. Foi baseado em outro projeto criado para a dissertação de mestrado de Hernando Barragan, auxiliado por Massimo Banzi e Casey Reas no Interaction-Ivrea, o Wiring, este, por sua vez, estava alicerçado no projeto de prototipagem eletrônica de Banzi, o Programma2003, e a plataforma Processing, idealizada por Reas e Ben Fry.

As primeiras placas Arduino foram projetadas por Massimo Banzi e David Cuartielles, nas quais Cuartielles ficou encarregado de desenvolver o software, baseado em Wiring, junto à colaboração de Nicholas Zambetti. Já o hardware e sua manufatura foi elaborado por Gianluca Martino. O projeto recebeu um grande apoio do Programa de Tecnologia Interativa, da Faculdade de Artes Tisch, na Universidade de Nova Iorque, um dos fatores que indicam que, diferentemente de outros projetos de prototipagem eletrônica, o Arduino não era destinado à indústria ou ao especialista, mas sim a iniciantes ou pessoas com pouca afinidade na área da eletrônica [32].

A linguagem de programação do Arduino é muito semelhante à linguagem C, tornando-a acessível para quem conhece minimamente as principais linguagens de programação, e é escrita dentro de uma interface chamada Arduino IDE (Figura 2.12). A versão mais popular do Arduino, denominada Arduino UNO (Figura 2.13), conta com um microcontrolador ATmega328P, 16 pinos digitais de entrada ou saída, 6 pinos analógicos de entrada, um cristal de quartzo de 16 MHz, um conector USB, um conector de alimentação e um cabeçote ICSP [33]. A intenção de construção do Arduino é de que, como seu público-alvo não necessariamente tem conhecimentos mais aprofundados em eletrônica, a placa possua dispositivos de segurança suficientes para que nenhum tipo de dano à placa possa ocorrer devido à má utilização, por exemplo, um curto circuito.

```
list_i2c_dispositives | Arduino 1.8.10
Arquivo Editar Sketch Ferramentas Ajuda
list_i2c_dispositives
#include <Wire.h>

void setup()
{
  Wire.begin();

  Serial.begin(9600);
  while (!Serial); // Leonardo: wait for serial monitor
  Serial.println("\nI2C Scanner");
}

void loop()
{
  byte error, address;
  int nDevices;

  Serial.println("Scanning...");
```

Figura 2.12: IDE Arduino. [32]

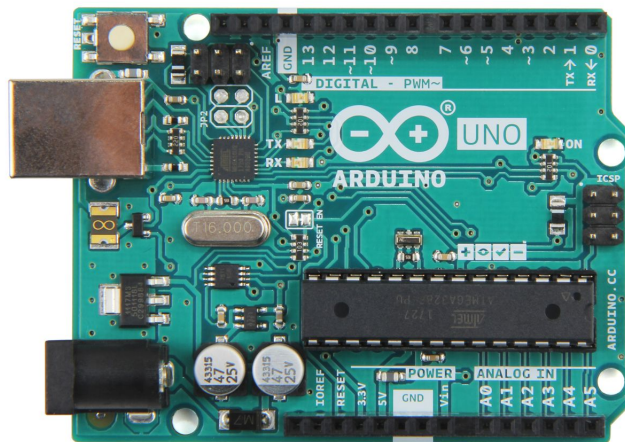


Figura 2.13: Arduino Uno Rev3. [33]

2.5.6 Redes sem fio

As redes sem fio são utilizadas como uma alternativa às redes com fio, considerando a facilidade de instalação de uma infraestrutura, os custos elevados gerados pelo cabeamento e as áreas a serem cobertas nem sempre possibilitarem a presença de cabos. Além disso, as redes sem fio permitem que os usuários se locomovam permanecendo conectados, o que difundiu os serviços de voz e dados, exemplificados pelas redes de celulares [34].

Historicamente na telefonia, o serviço básico tem sido a aplicação da voz. Com o advento dos computadores, a partir da última década, este cenário vem mudando. Desde a criação dos primeiros roteadores, que permitem a comunicação entre computadores, criou-se uma nova necessidade ao homem, comunicar-se através do transporte de pacotes de dados [35].

A evolução dos sistemas celulares está dividida em gerações. As redes da Primeira Geração (1G) são redes móveis analógicas ou semi-analógicas, que oferecem serviços básicos para usuários e a ênfase está nos serviços de voz. A especificação da Segunda Geração (2G), feita por grupos internacionais, se deu pela necessidade de um sistema de comunicação móvel globalizado e a melhoria das condições da geração anterior. O destaque da rede 2G está na compatibilidade e na transparência internacional, além da mudança de protocolos de telefonia móvel analógica para digital.

Seu desenvolvimento deriva da necessidade de poder ter um maior número de ligações simultâneas. Para isso foram introduzidos protocolos de telefonia digital que além de permitir mais conexões simultâneas com a mesma largura de banda, também permitiam integrar outros serviços, que anteriormente eram independentes, no mesmo sinal, como o envio de mensagens de texto (Short Message Service - SMS) e a transmissão de dados entre dispositivos de fax e modem. O sucesso comercial dessa geração é o Global System for Mobile communications (GSM), que possui uma taxa de transmissão de dados de 9.6 kbps. Já as redes 3G permitem telefonia móvel de longo alcance e evoluíram para incorporar redes de acesso à Internet em alta velocidade. Essa tecnologia é capaz de suportar um número maior de clientes de voz e dados, (especialmente em áreas urbanas)

e também proporciona maiores taxas de dados, permitindo a transmissão de 384 kbits/s para sistemas móveis e 7 Megabits/s para sistemas estacionários. Para se fazer a transição entre os sistemas 2G para o 3G é necessária a troca de equipamentos dos usuários e a introdução de novos equipamentos de rede. Tais alterações dificultaram a mudança direta, surgindo então os sistemas denominados 2,5G, tais como o General Packet Radio Service (GPRS) [36].

2.5.7 Smartphones, Sistema Android e Tecnologias de desenvolvimento

Os smartphones são uma evolução dos telefones móveis, geralmente compostos por um Sistema Operativo, acesso a internet, GPS e possibilidade de acesso a diversos aplicativos. Em 2019, segundo [37] existiram mais de 3.2 bilhões de usuários de smartphone pelo mundo. Se tornando o tipo de equipamento mais utilizado para acesso a informações, devido sua mobilidade, baixo custo e intuitividade.

O principal Sistema Operativo utilizado atualmente é o Android, pertencente a Google, e totalizando mais de 2.5 bilhões de dispositivos [38]. O Android teve seu desenvolvimento iniciado em 2003 pela startup Android Inc e em 2005 foi adquirido pela Google, que foi o responsável por gerir grande parte do desenvolvimento da plataforma.

O primeiro aparelho Android, o HTC Dream G1, foi lançado no dia 22 de outubro de 2008. Ele possuía funcionalidades bastante avançadas para a época, como a janela de notificações, a integração com Gmail e uma central de aplicativos, denominada ‘Android Market’ [39]. Em 2011, o Android se torna o líder mundial no segmento de SO’s para smartphones [40].

Já com relação aos aplicativos utilizados em smartphones, existem diversas tecnologias para o desenvolvimento dos mesmos, e elas são geralmente categorizadas em: WebApp, Nativa e Cross-platform.

As tecnologias WebApp, permitem que os smartphones acessem aplicações desenvolvidas para ambiente Web, estas que possuem código-fonte geralmente escrito em HTML,

CSS e JavaScript. Os WebApps são executados em um ambiente Web, de forma que em boa parte dos casos é preciso ter conexão com a Internet para acesso [41].

Já as tecnologias Nativas criam o aplicativo para uma plataforma específica, fazendo que a aplicação tenha uma compatibilidade maior com a plataforma na qual ela for destinada. Isso permite que além dos recursos já existentes, ela também seja compatível com os novos recursos disponibilizados na plataforma, algo que pode influenciar diretamente no desempenho, pelo fato dos recursos serem utilizados de forma mais otimizada.

A tecnologia cross-platform é uma mistura dos principais pontos positivos das tecnologias citadas acima. Com ela é possível criar aplicativos para diferentes plataformas, permitindo que o código-fonte seja reutilizável, e os recursos de cada plataforma sejam usufruídos de forma otimizada. Para o desenvolvimento de aplicações com esse tipo de tecnologia, geralmente são utilizados Frameworks. Um exemplo é o Flutter, uma ferramenta desenvolvida pela Google e que utiliza a linguagem Dart para o desenvolvimento de aplicações compatíveis com os SO's Android e iOS.

2.6 Trabalhos Relacionados

Nesse capítulo, serão apresentados alguns trabalhos que tem correlação com o tema abordado por este trabalho, estes que, contribuíram diretamente com o desenvolvimento do projeto que está sendo estruturado.

2.6.1 Sistema de irrigação com monitoramento remoto

Correia et al. [42] desenvolveu um trabalho no qual foi apresentado uma solução em monitoramento e controle automático de sistemas de irrigação. O trabalho envolveu tanto o desenvolvimento de um software, para que o usuário conseguisse ter o acesso a variáveis importantes nos processos e pudesse atuar (de forma remota) sobre o sistema, quanto o de um equipamento que através de um microcontrolador Arduíno e um dispositivo HLK-RM04, pudesse possibilitar a integração de informações, via comunicação WEB, entre o equipamento e o software.

Para obter as informações necessárias para o monitoramento do local, foi necessário utilizar três sensores: O sensor DHT11, responsável por informar a temperatura e a umidade relativa do ar do meio, o sensor YL-83 que quantifica o volume de água proveniente da chuva e o módulo RTC-DS1307 que sincroniza a data e o horário do sistema com a data e o horário do servidor Web, para medições mais precisas em relação ao tempo. Também para completar o trabalho e possibilitar o controle remoto do sistema de irrigação através do usuário, foi utilizado como atuador uma válvula solenoide, que controlava a ejeção ou não da água utilizada para a irrigação (Figura 2.14).

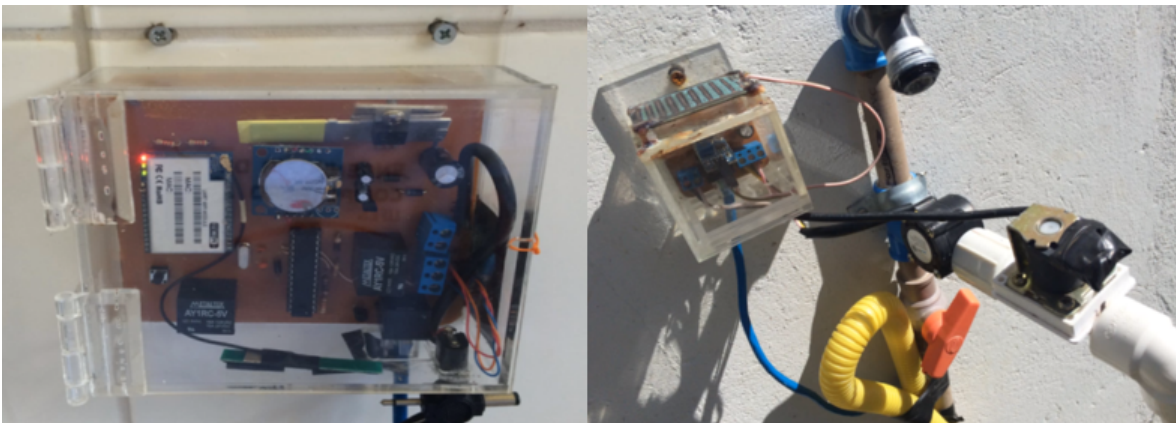


Figura 2.14: Protótipo montado para obtenção de dados e atuação no sistema. [42]

Após a obtenção dos dados através dos sensores, o equipamento pode atuar sobre o sistema tanto de forma automática quanto de forma manual, controlado pelo usuário através do software. Para que o funcionamento automático fosse utilizado, o sistema, através de algoritmos, realizava um cálculo que, através de valores de referência definidos, ejetava uma quantidade de água conforme as condições do ambiente. Caso a forma manual fosse utilizada, o software mostrava ao usuário um conjunto de dados, como: temperatura, umidade relativa do ar, intensidade da chuva naquele momento (fraca, moderada ou forte), data e horário do sistema, volume de água ejetado durante a irrigação daquele dia, volume de água acumulado da chuva (no modo automático) e o status de conexão do dispositivo físico com a Internet. O software também permite o usuário tanto controlar a válvula momentaneamente como pré-definir uma programação de irrigação (Conforme mostra a

Figura 2.15, que contem algumas das telas contidas no software).

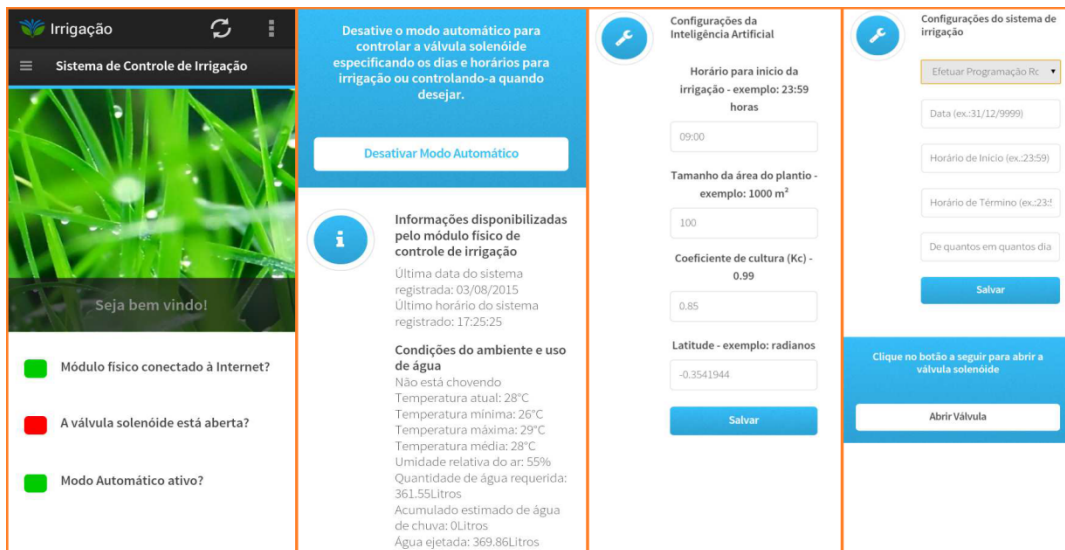


Figura 2.15: Software para o gerenciamento do equipamento. [42]

2.6.2 Estação automática para monitoramento de compostagem

Jordão [43] buscou desenvolver uma estação automática de baixo custo para o monitoramento da temperatura de compostagem. A estação desenvolvida tinha como objetivo ser capaz de monitorar, dentro um intervalo específico, a temperatura do ar em dois pontos diferentes em uma leira de compostagem, suportando condições de temperaturas superiores a 70°C e utilizando equipamentos de baixo custo. A medição de temperatura no processo de compostagem é fundamental, pois conforme afirma Epstein [44], esse parâmetro é um dos principais indicadores da qualidade e segurança sanitária do composto.

Para mensurar os parâmetros, foi desenvolvido um CMS (Compost Monitoring System). Esse sistema (conforme demonstrado pela Figura 2.16) é composto por 4 unidades: uma unidade de controle, uma unidade de contagem de tempo e armazenamento de dados, uma unidade de medição e uma unidade de abastecimento.

Cada unidade é composta pelos seguintes componentes:

- Unidade de Controle:
 - Plataforma Arduino, para atuar como controlador.

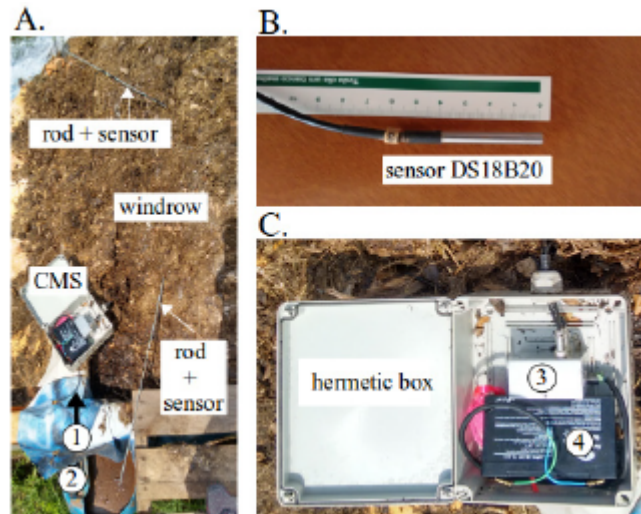


Figura 2.16: Protótipo montado para obtenção de dados e atuação no sistema. [43]

- Unidade de contagem de tempo e armazenamento de dados:
 - Módulo DS1307, para sincronizar a data e hora;
 - SD-Card com capacidade armazenar 2 GB de dados colhidos pelos sensores.
- Unidade de medição:
 - Três sensores de temperatura DS18B20 que possuíam uma taxa de amostragem máxima de 750 milissegundos (MAXIM INTEGRATED, 2015). Estes que foram ligados a uma haste de metal, que permitiram a permanência de cada um em diferentes profundidades no interior da pilha de compostagem.
- Unidade de abastecimento:
 - Bateria recarregável de 6V-12A.

Após a calibração dos sensores e o desenvolvimento de um algoritmo que, através do microcontrolador, fosse o responsável pelo controle do registo de data, hora e temperatura dos três sensores de temperatura e também por armazenar os dados na unidade de armazenamento, foi feita a aplicação do equipamento em campo. O teste de campo durou sete

dias e foi operado com todos os sensores sem interrupções ou geração de dados espúrios, demonstrou a eficácia operacional do CMS em comparação com as opções convencionais. A utilização do CMS reduziu o número de visitas do operador, visto que as visitas foram limitadas à substituição da bateria e do cartão de memória da estação, além de permitir o aumento da resolução espacial e temporal de medição da temperatura.

2.6.3 Análise automatizada de ambiente para aquicultura

Dos Santos et al. [45], desenvolveu um trabalho para analisar, de forma automática, o pH e a temperatura da água para aquicultura. Segundo Silveira et al. [46], a utilização de tecnologias no processo de produção animal influencia diretamente na diminuição dos custos de produção e no melhor aproveitamento das condições químicas, físicas e biológicas do ambiente.

Conforme Braga et al. [47] descreve em seu trabalho, há diversos parâmetros físico-químicos determinantes para a caracterização da qualidade da água. Dentre esses parâmetros destacam-se, o pH (potencial hidrogeniônico) e a temperatura [48]. Conforme descrevem Arana [49] e Moro et al. [50] valores de pH inferiores a 6,5 ou superiores a 8,5 causam problemas fisiológicos diversos, tanto no crescimento, na reprodução, bem como são causas de mortalidade expressiva do plantel. Em seus trabalhos Arana [49] e Moro et al. [50] também mostram que, temperaturas acima ou abaixo da faixa ideal podem inibir o crescimento, além de favorecer a incidência de doenças.

Devido a esses fatos, foi estruturado um equipamento que através de um microcontrolador e de sensores, possibilitasse o registro e visualização instantânea dos valores de pH e temperatura do meio analisado. Para que isso fosse possível, foi utilizada uma placa com o microcontrolador Atmega2560, um sensor DS18B20 para aferir a temperatura, um kit SEN0161 que contém um sensor de pH e um módulo para repassar as informações ao microcontrolador, um módulo de cartão SD, para armazenamento de dados, uma fonte de alimentação de e um visor de LCD, para proporcionar um melhor acompanhamento das leituras aferidas pelo sistema. A montagem do sistema foi feita conforme mostrado

na figura 2.17.

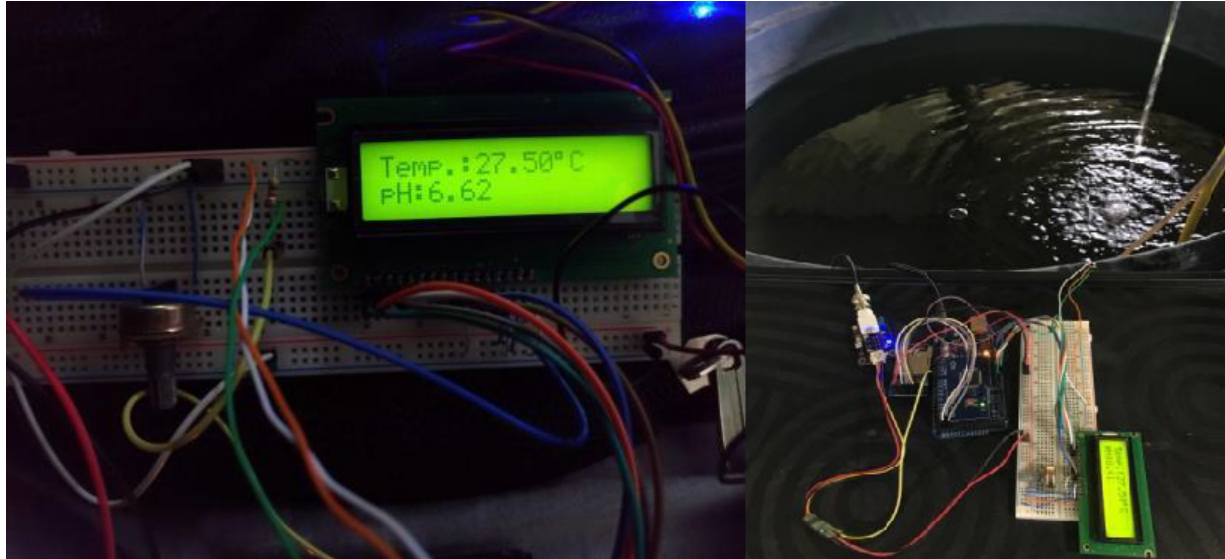


Figura 2.17: Protótipo do sistema montado e sendo testado em um tanque com peixes. [45]

Após a montagem do protótipo do sistema, foram realizados dois testes: O de precisão dos sensores e o da atuação do sistema em um tanque com peixes. O primeiro teste demonstrou que o kit pH SEN0161 estava descalibrado, pois apresentou valores muito discrepantes dos valores reais de pH das soluções tampão. Após serem feitas regulagens no potenciômetro (regulador de tensão) do módulo e no código, o kit conseguiu obter informações precisas. Já o sensor de temperatura DS18B20 obteve variação de $0,5^{\circ}\text{C}$ em relação à mesma medição utilizando um termômetro de álcool, demonstrando a boa eficácia de uso deste sensor no sistema.

Já no segundo teste, o protótipo do sistema foi introduzido em um tanque de peixes, com o propósito de executar, a cada hora, as leituras de temperatura e pH, mostrar os resultados das análises no visor de LCD, salvar estes resultados em formato 'txt' no cartão de memória. Através da análise dos dados obtidos, a autora pode comprovar que, os resultados obtidos nos testes demonstraram boa acurácia e precisão na aferição, demonstrando o potencial de uso da plataforma Arduino e seus sensores na utilização em atividades aquícolas.

Capítulo 3

Abordagem/Análise/Modelação

Para que a solução seja estruturada de forma que possa atender os objetivos propostos, será realizada uma série de etapas. Elas variam desde a escolha de cada um dos componentes a serem utilizados em conjunto com simulações sobre como cada tipo de informação obtida seria enviada para a aplicação, como também o processo de desenvolvimento da aplicação e visualização dos dados na mesma.

3.1 Etapas

As etapas que já foram, ou serão realizadas para a elaboração do protótipo, são as seguintes:

1. Estudo de cada um dos componentes: Antes da escolha de cada um dos componentes a serem utilizados, foi realizado um estudo em cima de quatro aspectos: As especificações, a disponibilidade de documentação, o preço e a aplicação dos mesmos em trabalhos, práticas e/ou projetos. Essa etapa foi realizada para diminuir a chance de ocorrer incompatibilidade entre os componentes utilizados pela solução, além de permitir com que cada um deles possa ser montado e operado conforme indicado pelo fabricante ou conforme foi utilizado em outros trabalhos que obtiveram êxito na utilização dos componentes.

2. Definição da arquitetura dos componentes: Após os estudos iniciais dos componentes utilizados, foi realizada a definição da arquitetura de conexão dos componentes, definindo assim a parte de hardware da solução. Para que a arquitetura fosse definida, foram analisados tanto os manuais e guias de suporte de cada um dos componentes, como também as soluções utilizadas em trabalhos relacionados com o tema. Essa etapa é importante para que o processo de montagem siga um padrão, permitindo que esse processo seja facilitado e menos susceptível a erros.
3. Desenvolvimento da base de dados para hospedar os dados obtidos: Após a definição da arquitetura na parte de hardware, foi definida a arquitetura na base de dados da solução. Ela foi estruturada de forma que todos os dados obtidos pelo hardware, sejam tratados e armazenados na nuvem. Dessa forma, além de serem guardados em segurança, os dados obtidos poderão ser acessados pela aplicação, que permitirá que seus usuários os visualizem, analisem e/ou modifiquem.
4. Simulação do comportamento da base de dados: Após o banco de dados ser implantado e a parte de hardware ser arquitetada, foram iniciadas as simulações do funcionamento desse ambiente. Essas simulações foram baseadas em situações nas quais todos os dados capazes de serem lidos pelos sensores, fossem tratados pelo módulo arduino e enviados para a base de dados. Essa etapa foi fundamental para verificar o funcionamento da parte de comunicação entre as partes, verificando a integridade e confiabilidade no envio das informações.
5. Arquitetura e desenvolvimento da aplicação: Após a parte de hardware estar definida, o envio de informações estar funcional e o banco de dados estar apto a armazenar os dados, as telas da aplicação foram arquitetadas. A aplicação teve como principal objetivo, a capacidade de informar aos usuários todos os dados obtidos pelos sensores (na parte de hardware) que integram a solução, como também os dados da localização na qual os equipamentos com os sensores estejam presentes.

6. Testes da aplicação: Com a aplicação desenvolvida, foram realizados testes que verificaram se os dados presentes no banco, foram obtidos de forma íntegra pela aplicação. Dessa forma, garantiu-se a possibilidade de que os usuários, com a posse das informações registradas pelos sensores, possam realizar as análises da qualidade da água no local analisado de forma remota, além de que sejam informados sobre qualquer alteração brusca em algum parâmetro físico-químico ocorra.
7. Conclusão: Após a realização de todas as etapas acima, os resultados obtidos serão agrupados e analisados, de forma a mostrar os pontos positivos e negativos em relação ao funcionamento e implantação da solução, bem como as possíveis extensões para o trabalho.

3.2 Materiais

Para este trabalho serão utilizados os seguintes componentes:

3.2.1 Microcontrolador

Para o papel de controlador do sistema, será utilizado a plataforma de prototipagem Arduino MEGA 2560 (Demonstrado pela Figura 3.1), esta que é equipada com um microcontrolador de 8 bits Atmel Mega16u2, possui 54 pinos de entrada/saída digitais, 16 entradas analógicas, 4 portas seriais UART, uma conexão USB e um conector de alimentação. Ele possui uma tensão de operação de 5 V e pode ser alimentado por fontes que fornecem de 7 a 12 V, sua velocidade de clock pode operar a 16 MHz e possui 256 KB de memória Flash [33].

Três motivos foram predominantes para a escolha dessa plataforma, o primeiro foi o fato de que as plataformas Arduino são as mais utilizadas em projetos com objetivos semelhantes a esse trabalho, proporcionando uma maior facilidade em encontrar componentes compatíveis, artigos, documentações e guias que auxiliam no desenvolvimento do trabalho. O segundo motivo foi seu baixo custo e o terceiro a grande quantidade de portas

digitais, analógicas e seriais contidas na plataforma, algo necessário para receber os dados provenientes dos sensores que compõem o protótipo.



Figura 3.1: Arduino MEGA 2560. [51]

3.2.2 Sensor de pH

Para mensurar o valor de pH do meio, foi utilizado o eletrodo de referência cloreto de prata/prata H-101. Ele é capaz de indicar a neutralidade, acidez ou até mesmo a alcalinidade de uma solução líquida e transmite os dados obtidos via conector BNC. Para intermediar a conexão entre o eletrodo e o Arduino é utilizado o módulo DFRobot pH meter v1.1, que também possui um potenciômetro que permite ajustes na tensão fornecida ao eletrodo, permitindo a calibração do mesmo. Esse conjunto deve ser alimentado por uma tensão de 5V (Figura 3.2).



Figura 3.2: Eletrodo E-201c e módulo PH-4502C. [52]

Ambos os componentes compõem o kit, que é destinado para medições a nível industrial. Esse kit foi escolhido devido a sua durabilidade (cerca de 1 ano), sua precisão em ambientes com condições adversas e sua membrana em anel, que dificulta o entupimento. Todas essas características fazem com que o kit seja adequado para um monitoramento remoto a longo prazo, algo que se alinha com os objetivos do trabalho.

3.2.3 Sensor de temperatura

Para medir a temperatura do meio aquoso foi utilizado o sensor DS18B20 (Figura 3.3). Segundo sua documentação, esse sensor é a prova d'água, alcança a precisão de até 0.5° C e também proporciona leituras de até 12-bits através de sua comunicação com o microcontrolador. Ele consegue ser operado com uma tensão de entrada entre 3 e 5,5V e possui uma faixa de medição entre -55° C a +125°C.



Figura 3.3: Sensor DS18B20. [53]

Já para medir a temperatura externa do meio foi utilizado o sensor HTU-21d (Figura 3.4). Além de conseguir registrar dados sobre a temperatura, ele também é capaz de registrar a umidade relativa do ar do meio externo. Ele opera com uma tensão de entrada de 3,3V.



Figura 3.4: Sensor HTU-21d. [54]

Ambos os componentes foram escolhidos devido seu baixo custo, a existência de documentação e sua ampla utilização em projetos com objetivos semelhantes, algo que motiva o seu uso.

3.2.4 Sensor de condutividade

Para mensurar o valor da condutividade do meio aquoso será utilizado o Kit DFRobot Analog Electric Conductivity Sensor V2 ($K = 1$) (Figura 3.5). Os principais componentes do kit são um eletrodo para a medição da condutividade e um módulo para intermediar a comunicação entre o eletrodo e o Arduino, além de possibilitar o ajuste da tensão fornecida ao eletrodo, de forma a calibrar o mesmo.

O kit foi escolhido devido a sua escala de mensuração valores de condutividade, que está entre 0 a 20 ms/cm, ser capaz de detectar o valor máximo recomendável de 1000 μ /cm. [20] Outro fator importante é a alimentação do kit, que deve ser entre 3 e 5 V.



Figura 3.5: Kit DFRobot Analog Electric Conductivity V2. [55]

3.2.5 Sensor de oxigênio dissolvido

Para medir o teor de oxigênio dissolvido do meio, será utilizado o kit DFRobot Analog Dissolved Oxygen Sensor/Meter (Figura 3.6). Os principais componentes do kit são uma sonda galvânica para mensurar o valor de oxigênio dissolvido e um módulo para intermediar a comunicação entre o eletrodo e o Arduino.

O kit foi escolhido por possuir sonda uma sonda galvânica, que não necessita de ser polarizada, e também por possuir uma solução de enchimento e a tampa de membrana substituíveis, resultando em baixo custo de manutenção. Ele possui uma faixa de detecção entre 0 e 20 mg/L, e pode ser alimentado com uma tensão de 3.3 a 5.5V [56].



Figura 3.6: Kit DFRobot Analog Dissolved Oxygen Sensor/Meter. [56]

3.2.6 Sensor de turbidez

Para medir a turbidez do meio, será utilizado o sensor SEN0189 (Figura 3.7). Ele é capaz de detectar partículas suspensas na água medindo a transmitância da luz e a taxa de espalhamento que variam com a quantidade de sólidos suspensos totais (TSS) na água. O conjunto deve ser operado com uma tensão de 5V e em um ambiente com temperatura

entre 5 e 90°C. A motivação de sua escolha foi a possibilidade do mesmo poder ser utilizado em rios e córregos.

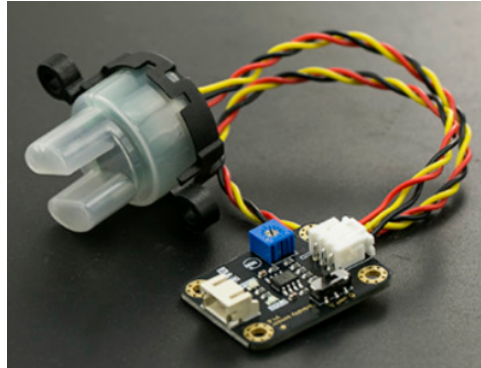


Figura 3.7: Kit de medição analógica da turbidez. [57]

3.2.7 Módulo para padronização da data/hora

Para obter precisão de data e hora no momento das medições pelos sensores, será utilizado o módulo DS1307 (Figura 3.8). Ele possui uma bateria própria e fornece ao microcontrolador as informações de data e hora do sistema. O ajuste com o fuso e data padrão deve ser realizado apenas durante a primeira utilização ou durante a troca de sua bateria. Ele pode ser operado com uma tensão de entrada de 3,3V, além de ser compatível aos formatos de hora e data utilizados mundialmente, sendo que o padrão pode ser alterado pelo usuário em seu código de execução.

3.2.8 Alimentação

Para garantir a alimentação do protótipo quando o mesmo estiver funcionando remotamente, serão utilizados adaptadores entre o padrão da bateria utilizada e a entrada de energia do componente. Para alimentar o Arduino será utilizado o adaptador mostrado pela figura 3.9, já para alimentação direta no protoboard será utilizado o adaptador mostrado pela figura 3.9 sem o conector na ponta.

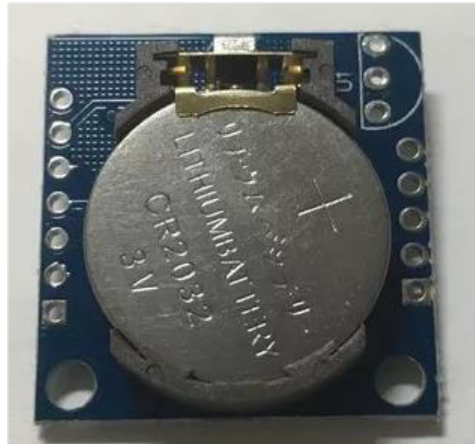


Figura 3.8: Módulo DS1307. [58]



Figura 3.9: Adaptador entre bateria de 9V e o padrão de alimentação do Arduino. [59]

3.2.9 Transmissão de dados

Para transmitir dados do Arduino a um endereço IP, foi utilizado o módulo SIM808 (Figura 3.10). Esse módulo utiliza o tipo de comunicação de dados via GPRS, aplicando o protocolo TCP/IP, permitindo que o Arduino envie ou receba dados de um endereço IP remoto. Ele opera com uma tensão de entrada entre 7 e 23V e é capaz de transmitir dados na frequência de 850 a 1900 MHz, sendo assim compatível com a rede 2G GSM.

Outro fator determinante para a escolha desse módulo, foi a possibilidade do mesmo enviar dados de localização através do GPS. Essa informação será importante tanto para um melhor gerenciamento, caso a rede de monitoramento seja ampla, como também para localizar o aparelho caso ele se desloque involuntariamente a partir do local de instalação inicial.



Figura 3.10: Módulo SIM808. [60]

3.3 Ferramentas

3.3.1 Firebase

O Firebase é um Baas (Backend as a Service) para aplicações Web e Mobile do Google, foi lançado em 2004 e atualmente é um dos serviços mais utilizados para banco de dados do tipo NoSQL. Esse serviço disponibiliza de forma simples toda a estrutura do backend, como por exemplo: configuração de servidor, integração com banco de dados, sistema de push notification e outros serviços. Essa disponibilidade permite que toda essa camada já

esteja completamente pronta para se integrar com a aplicação, restando ao desenvolvedor definir as regras para salvamento dos dados [61].

A plataforma oferece os seguintes serviços para o desenvolvimento de aplicações:

- **Firestore Analytics:** Fornece uma visão aprofundada sobre o uso da aplicação pelos usuários.
- **Firestore Cloud Messaging:** Suporte a uma plataforma de mensagens e notificações para Android, plataformas IOS e aplicativos da web.
- **Firestore Auth:** Permite autenticar usuários usando apenas um id para o cliente, autenticando-os com e-mail e senha a ser armazenada no Firestore e com integração a diversas plataformas (Facebook, Twiter, Google e Github).
- **Firestore Realtime Database:** Fornece o backend e um banco de dados que pode ser analisado em tempo real. Através de uma API, este serviço permite sincronizar as informações e armazená-las no Firestore.
- **Firestore Storage:** Serviço de upload e download de arquivos para aplicativos firebase.

3.3.2 Android Studio

O Android Studio é o Integrated Development Environment (IDE) oficial para o desenvolvimento de aplicativos Android e é baseado no IntelliJ IDEA. Esta plataforma foi anunciada em 2013 pela Google e foi criada com o principal objetivo de substituir o Eclipse, outra plataforma usada para construir aplicativos. Desta forma, com o Android Studio, a Google obteve seu próprio IDE para desenvolvimento de aplicativos, podendo instalar todo o SDK para desenvolver aplicativos específicos adaptados à maioria das versões.

Uma das principais características desta ferramenta é forma dela ser como um escritório do desenvolvedor, visto que ela contém de forma organizada os projetos, pastas, arquivos e tudo aquilo que é necessário para a criação do aplicativo. Outro ponto destaque é a presença de um emulador de Android, que permite visualizar o funcionamento

em tempo real das aplicações em desenvolvimento, verificando se as mesmas podem ser executadas correctamente e sem problemas. [62]

Esta ferramenta pode ser utilizada no Windows, Linux e Mac OS e devido a compatibilidade com diversas linguagens e frameworks, como o Dart e o Flutter, a ferramenta também possibilita o desenvolvimento de aplicações para a plataforma iOS.

3.3.3 Flutter

O Flutter é um cross-platform framework, lançado publicamente em 2016 pela Google, que visa desenvolver aplicativos móveis de alto desempenho. Esse framework permite o desenvolvimento de aplicativos que podem ser executados no Android, iOS e o Fuschia, o sistema operacional de próxima geração do Google.

Além da sua compatibilidade, o Flutter também oferece suporte ao hot-reload durante o desenvolvimento, algo muito importante para o impulsionamento do ciclo de desenvolvimento. O hot-reload permite a inserção de código-fonte atualizado no Dart VM em execução, sem alterar a estrutura interna do aplicativo, preservando todas as transições e ações do aplicativo após o hot-reloading [63].

Em vez de utilizar visualizações da web ou depender dos OEM widgets do dispositivo, o Flutter renderiza todos os componentes de visualização usando seu próprio mecanismo de renderização de alto desempenho, permitindo a construção de aplicativos com performance semelhante ao de aplicativos nativos. Em termos de arquitetura, o código C/C++ do mecanismo é compilado com NDK do Android e LLVM no iOS, e qualquer código em Dart primeiramente passa por uma compilação AOT, que transforma o código desenvolvido em um código nativo [64].

Capítulo 4

Arquitetura e Desenvolvimento

Devido a solução ser capaz de obter os dados, armazená-los no ambiente de nuvem e possibilitar que os usuários a acessem, ela é subdividida em três partes: Hardware, Base de dados e Software (Figura 4.1)

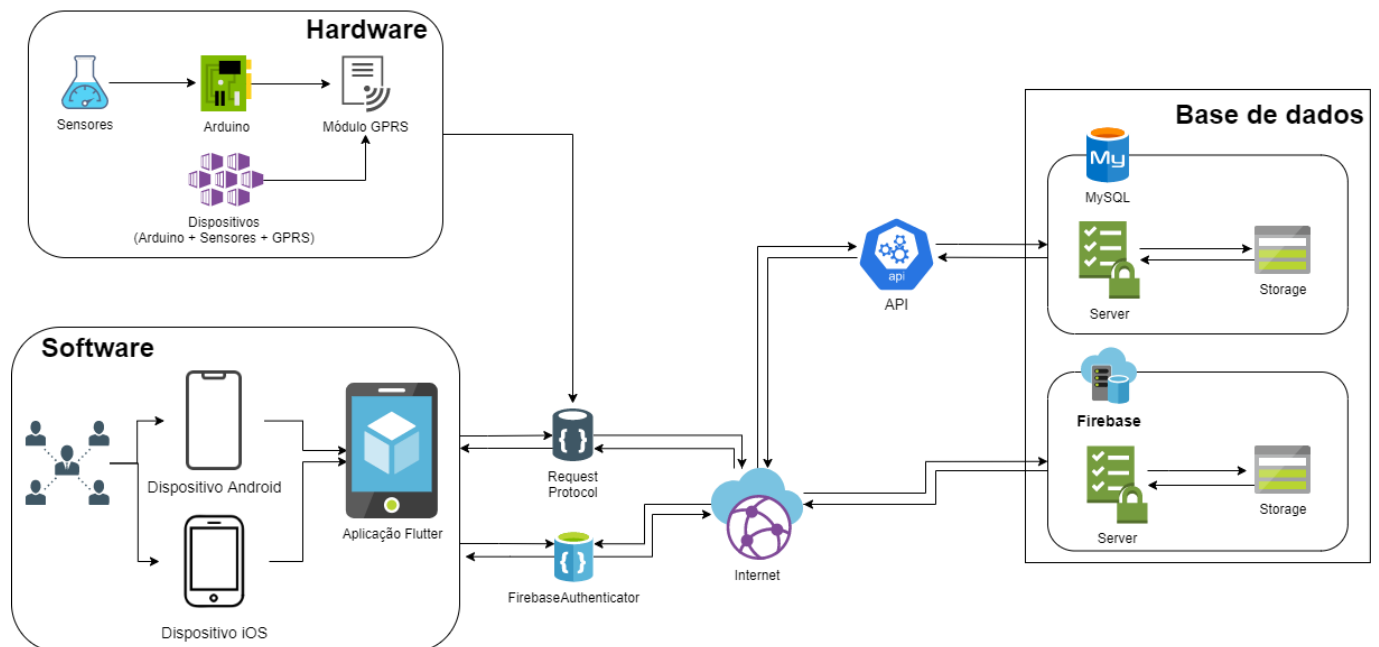


Figura 4.1: Diagrama geral da solução. [65]

A parte de Hardware será responsável por captar os dados, provenientes do meio no qual está sendo realizado a análise, e enviá-los para a base de dados. Para isso, os sensores

serão responsáveis por mensurar os dados físico-químicos do ambiente, passando-os para o microcontrolador arduino, que irá estruturar os dados obtidos em JSON, e enviá-los (através do módulo GPRS) a base de dados a ser utilizada.

A base de dados irá receber os dados, armazená-los de forma estruturada e disponibilizá-los a aplicação. Também ficará a cargo da base de dados a autenticação e o gerenciamento tanto de usuários como de operações realizadas no banco.

Já a parte de Software permitirá que os usuários possam acessar e analisar os dados obtidos pelos sensores. Além de poder cadastrar novos dispositivos, os usuários também poderão pesquisar dados sobre os dispositivos cadastrados, favoritar dispositivos e visualizar alertas, caso os parâmetros físico-químicos ultrapassem os valores pré-definidos.

4.1 Hardware

Para a seleção dos componentes de Hardware foram utilizados os seguintes critérios: especificações, a disponibilidade de documentação, o preço e a aplicação em outros trabalhos/práticas/projetos. No total foram utilizados dez componentes, de forma que cada componente tivesse compatibilidade com o microcontrolador, e que o protótipo também pudesse obter, salvar e enviar os dados.

Após a seleção dos componentes e o estudo da documentação dos mesmos, foi definida a seguinte arquitetura para a parte de Hardware (Figura 4.2):

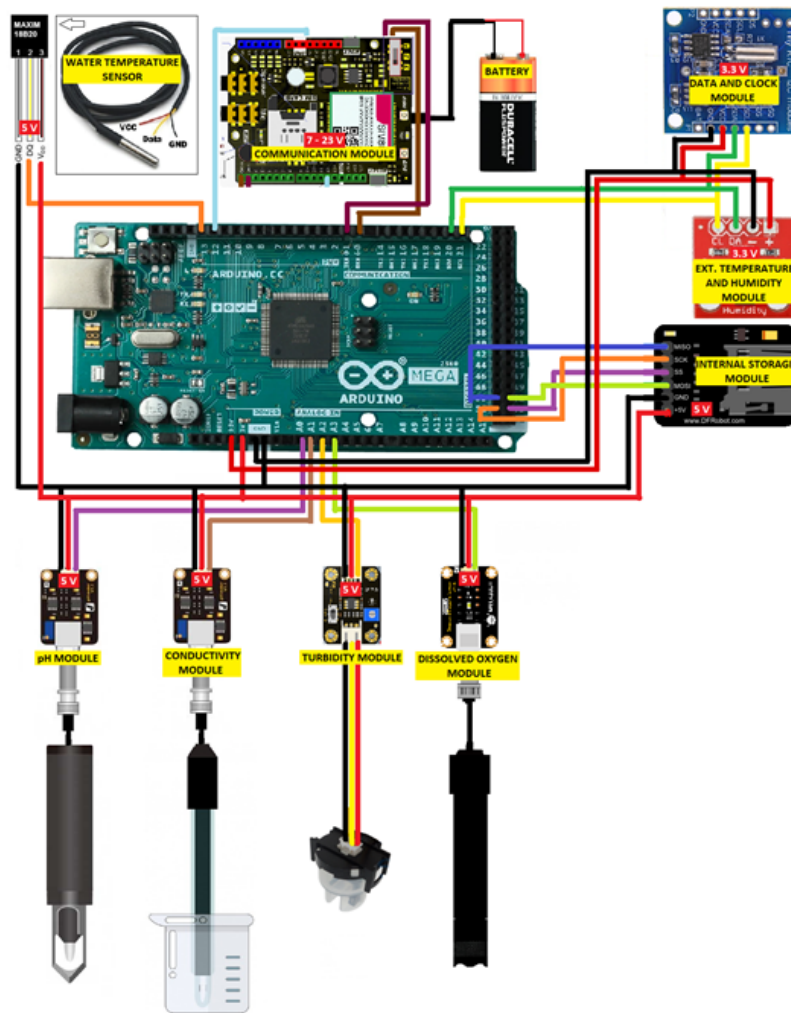


Figura 4.2: Diagrama geral da parte de Hardware. [65]

Com essa arquitetura, a parte de hardware da solução permite a obtenção dos seguintes parâmetros físico-químicos: pH, Condutividade, teor de oxigênio dissolvido, temperatura do meio e temperatura e umidade do ambiente externo. Além de enviar os dados através do módulo de comunicação via GPS, também é possível salvar os dados obtidos localmente, com a utilização de um módulo de armazenamento interno. O valor total de todos os componentes utilizados foi de aproximadamente 2.500 reais (375 euros).

Além das conexões realizadas, conforme a arquitetura definida (Figura 4.2), alguns dos componentes que fazem parte da solução só funcionam corretamente caso sejam executados os procedimentos definidos na documentação.

4.1.1 Microntronizador

Arduino

Além da alimentação e dos cabos de conexão, a ferramenta Arduino IDE é indispensável para a utilização e desenvolvimento no Arduino. Essa ferramenta permite que o desenvolvedor crie o código e o microncontrolador execute as operações definidas, fazendo com que os dados recebidos pelas conexões sejam estruturados e visualizados.

Essa ferramenta é disponibilizada tanto no site oficial do Arduino, quanto na Microsoft Store, podendo ser instalado de forma simples. Porém, para que ela possa ser utilizada, é necessário que o usuário realize a configuração da ferramenta antes de conectá-la ao Arduino. A primeira configuração a ser feita é a da placa Arduino que será utilizada, para o solução arquiteturada a placa arduino escolhida é a 'Arduino Mega 2560 or Mega 2560' (Figura 4.3) e a segunda configuração será a do processador, que para o projeto será o 'ATmega2560 (Mega 2560)' (Figura 4.4).

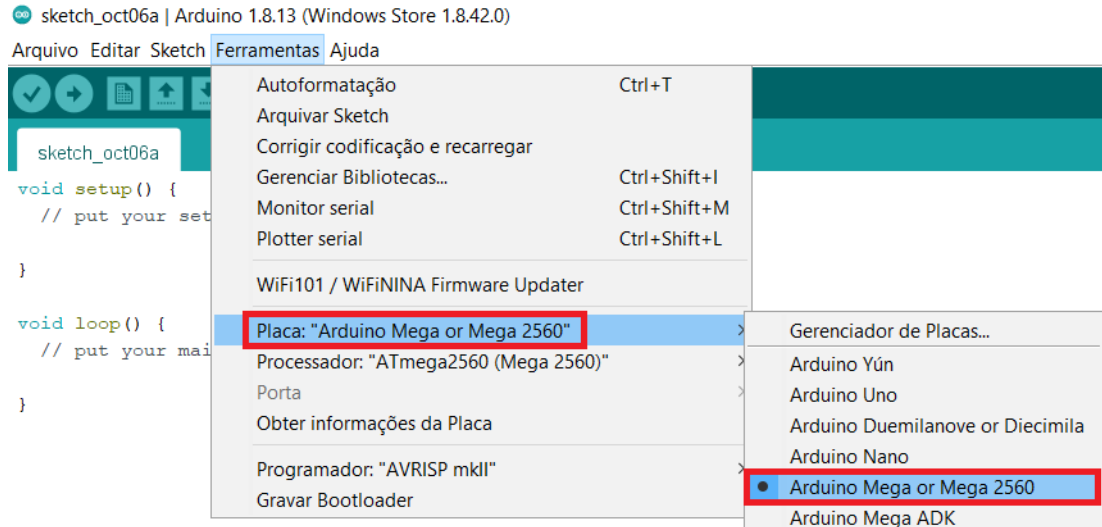


Figura 4.3: Processo de configuração da Placa Arduino a ser utilizada, na ferramenta Arduino IDE. [65]

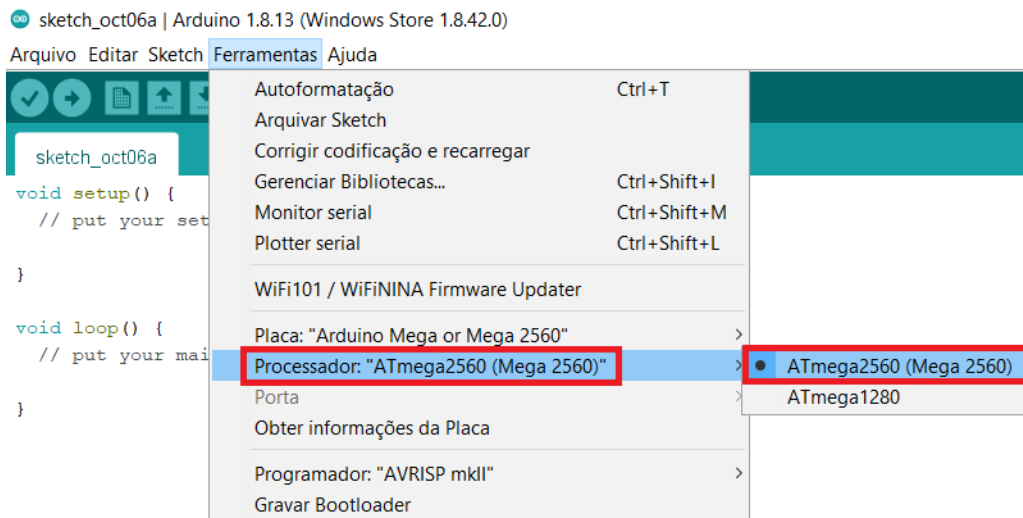


Figura 4.4: Processo de configuração da Placa Arduino a ser utilizada, na ferramenta Arduino IDE. [65]

Para desenvolver na ferramenta, o usuário deve utilizar a linguagem C ou C++. Para auxiliar no processo inicial de desenvolvimento, a própria ferramenta disponibiliza uma aba específica com alguns exemplos (Figura 4.5). Após desenvolver o código, o usuário deverá conectar o arduino no USB da máquina na qual está instalada a ferramenta e aguardar até que todos os drivers sejam instalados automaticamente. Depois dessa instalação e do desenvolvimento do código, o usuário pode verificar se o código está correto e enviá-lo ao microprocessador presente no Arduino, para que o mesmo seja executado (Figura 4.6).

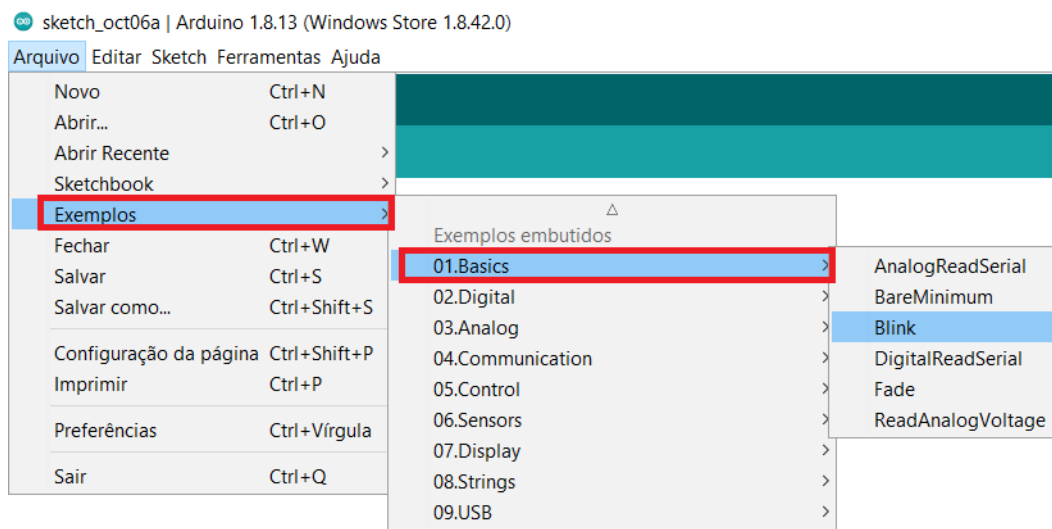


Figura 4.5: Exemplos disponibilizados na ferramenta Arduino IDE. [65]



Figura 4.6: Botões presentes na ferramenta Arduino IDE, para verificação, compilação e execução do código no micontrolador do Arduino.

4.1.2 Sensores

Módulo sensor de pH

Conforme arquitetado no diagrama de Hardware (Figura 4.2), o cabo 'Vcc' deve ser conectado a saída de 5V presente no Arduino e o cabo de transmissão de dados conectado na entrada analógica A0. Quando o módulo receber energia, a placa do conector BNC acenderá um led azul. Após esse primeiro processo o usuário deve utilizar o código desenvolvido pela fabricante [66] e após isso iniciar o processo de calibração do eletrodo.

Para iniciar o processo de calibração é necessário se preparar três soluções, uma neutra (cujo o pH é 7), uma ácida (cujo pH é 4) e uma básica (cujo pH é 9,18). Ao se inserir o eletrodo na solução neutra e executar o código da fabricante, será exibido no monitor serial o valor do pH com uma margem de erro embutida. Para corrigir essa margem, o usuário deve definir (no campo 'offset') o valor exato da diferença entre o pH real da solução e o pH exibido pelo monitor serial do Arduino IDE.

Após esse teste o usuário deverá limpar o eletrodo e inseri-lo na solução ácida, a partir do momento em que o pH exibido no monitor serial seja semelhante ao da solução, o eletrodo ficará calibrado para a leitura de soluções ácidas. Para realizar a calibração do

eletrodo em soluções básicas, basta repetir o processo anterior de calibração, porém deverá se limpar o eletrodo e utilizar a solução básica. A frequência de calibração, recomendada pelo fabricante, deve ser de 6 meses, porém caso a qualidade da água no local onde será situado o eletrodo seja ruim, é necessário que o mesmo seja calibrado com uma frequência maior. Outro fator importante a ser considerado é o fato de que as medições do eletrodo são realizadas em millivolts (mV), portanto o cálculo do pH é realizado seguindo os valores da tabela abaixo (Figura 4.7).

VOLTAGE (mV)	pH value	VOLTAGE (mV)	pH value
414.12	0.00	-414.12	14.00
354.96	1.00	-354.96	13.00
295.80	2.00	-295.80	12.00
236.64	3.00	-236.64	11.00
177.48	4.00	-177.48	10.00
118.32	5.00	-118.32	9.00
59.16	6.00	-59.16	8.00
0.00	7.00	0.00	7.00

Figura 4.7: Tabela a ser considerada na conversão da medição em millivolts (mV) em pH. [66]

Módulo sensor de temperatura e umidade

Conforme arquitetado no diagrama de Hardware (Figura 4.2), para o módulo de temperatura externa e umidade, o cabo 'Vcc' deve ser conectado a saída de 3.3V presente no Arduino e os cabos de transmissão de dados, SCA e SCL, conectados respectivamente nas entradas de comunicação 20 e 21. Já para a conexão do módulo de temperatura interna, é necessário que o cabo 'Vcc' seja conectado na saída de 5V do Arduino, e o cabo de transmissão de dados conectado na entrada digital 13.

Após a conexão ser realizada, deve-se executar o código exemplo definido na biblioteca 'Adafruit Unified Sensor' [67] e adapta-lo com o código presente de configuração do módulo de temperatura interna [68], definindo a porta digital 13, como a que irá receber os dados do sensor de temperatura interna. Caso os cabos estejam conectados corretamente, os dados irão ser exibidos no monitor serial da forma correta.

Módulo sensor de condutividade

Conforme arquitetado no diagrama de Hardware (Figura 4.2), para o módulo sensor de condutividade, o cabo 'Vcc' deve ser conectado a saída de 5V presente no Arduino e o cabo de transmissão de dados conectado na entrada analógica A1. Após a conexão ser realizada, devem ser realizadas duas calibrações antes de se obter dados precisos ao executar o código definido pela fabricante [69].

Para que a sonda possa capturar dados corretos, ela precisa passar por duas calibrações, sendo necessário a utilização de duas soluções tampão de 1413 us/cm e 12,88 ms/cm. Primeiramente, a sonda deve ser lavada com água destilada e após isso ser seca com papel absorvente. Com isso, a sonda deve ser inserida na solução tampão de 1413us/cm e movimentada até que os valores registrados no monitor serial fiquem estáveis. Após esse procedimento, é necessário que o usuário insira o comando 'enter' (Figura 4.8) para que se entre no modo de calibração e após isso, insira o comando 'cal' (Figura 4.9), para que o programa vincule a leitura obtida com a solução tampão de 1413 us/cm, e no final insira o comando 'exit' (Figura 4.10) finalizando a primeira calibração.

A sonda deve ser lavada novamente com água destilada e seca com papel absorvente, na segunda calibração, a sonda deve ser inserida na solução tampão de 12,88 ms/cm e movimentada até que os valores registrados no monitor serial fiquem estáveis. Após esse procedimento, é necessário que o usuário insira o comando 'enter' (Figura 4.8) para que se entre no modo de calibração e após isso, insira o comando 'cal' (Figura 4.9), para que o programa vincule a leitura obtida com a solução tampão de 12,88 ms/cm, e novamente, no final insira o comando 'exit' (Figura 4.10) finalizando-se a segunda calibração.

Depois de concluir as etapas acima, a calibração é concluída e, em seguida, o módulo pode ser usado para a medição real. Durante a inserção dos comandos 'enter', 'cal' e 'exit', os parâmetros relevantes no processo de calibração são salvos na EEPROM da placa de controle do Arduino.

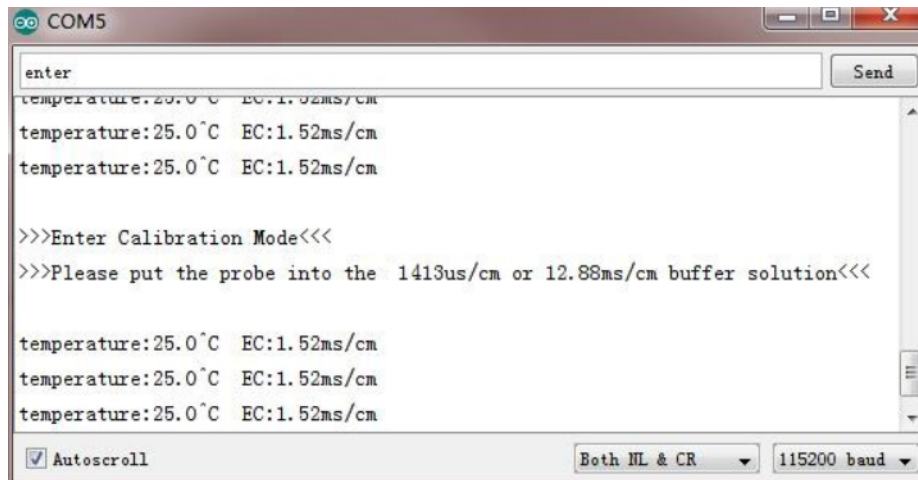


Figura 4.8: Comando 'enter' sendo inserido através do monitor serial. [69]

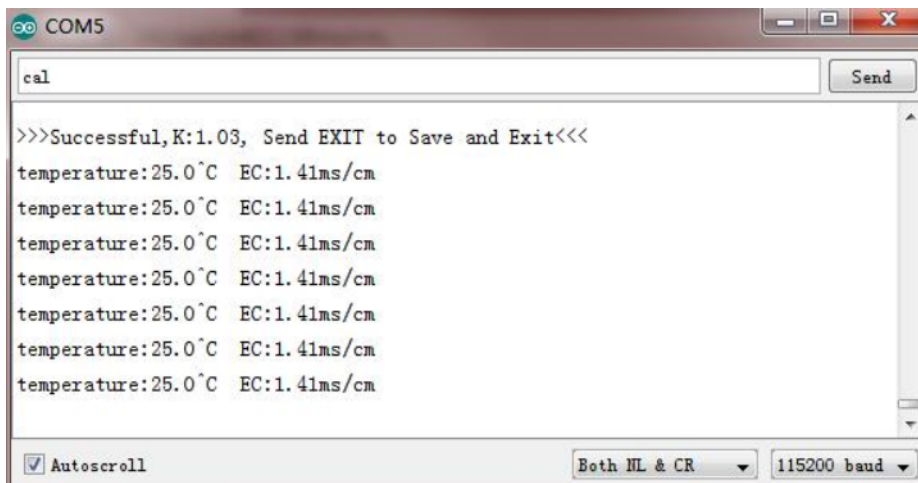


Figura 4.9: Comando 'cal' sendo inserido através do monitor serial. [69]

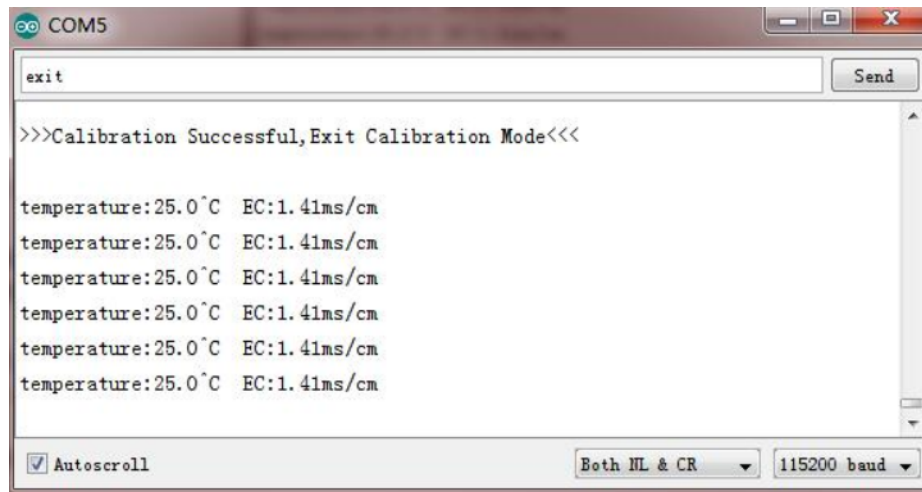


Figura 4.10: Comando 'exit' sendo inserido através do monitor serial. [69]

Módulo sensor de turbidez

Conforme arquitetado no diagrama de Hardware (Figura 4.2), para o módulo sensor de turbidez, o cabo 'Vcc' deve ser conectado a saída de 5V presente no Arduino e o cabo de transmissão de dados conectado na entrada analógica A2. Após a conexão ser realizada, deve-se utilizar o código definido pelo fabricante [70] e verificar se em um copo de água pura com temperatura entre 10 e 50 °C, o sensor retorna um valor abaixo de 0,5 NTU (4.10.3V). O sensor também é capaz de retornar a voltagem obtida e a relação entre a voltagem e a medição em NTU, presente na figura 4.11.

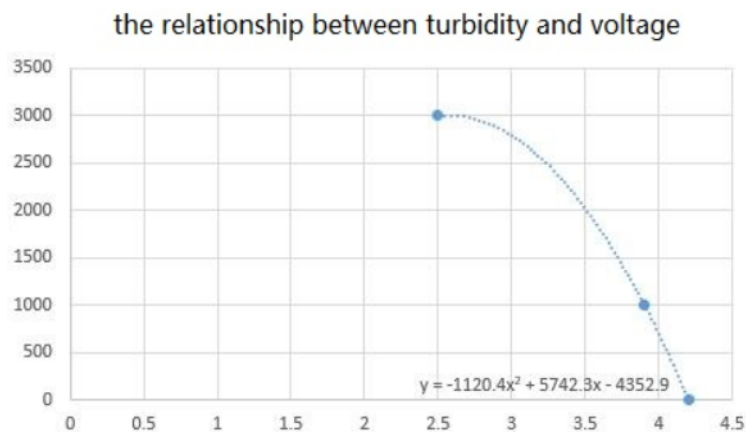


Figura 4.11: Tabela de relação entre a voltagem e o valor em NTU's. [70]

Módulo sensor de oxigênio dissolvido

Conforme arquitetado no diagrama de Hardware (Figura 4.2), para o módulo sensor de oxigênio dissolvido, o cabo 'Vcc' deve ser conectado a saída de 5V presente no Arduino e o cabo de transmissão de dados conectado na entrada analógica A3. Após a conexão ser realizada, deve-se retirar a tampa da membrana e adicionar (com o manuseio de luvas, devido a corrosividade) uma solução de NaOH 0.5 mol/L, até que se preencha 2/3 do volume. Após isso, deve-se enroscar a tampa na sonda em uma posição vertical, de forma que evite-se a formação de bolhas na solução [71]. A figura 4.12 ilustra a forma de como esse processo deve ser realizado.

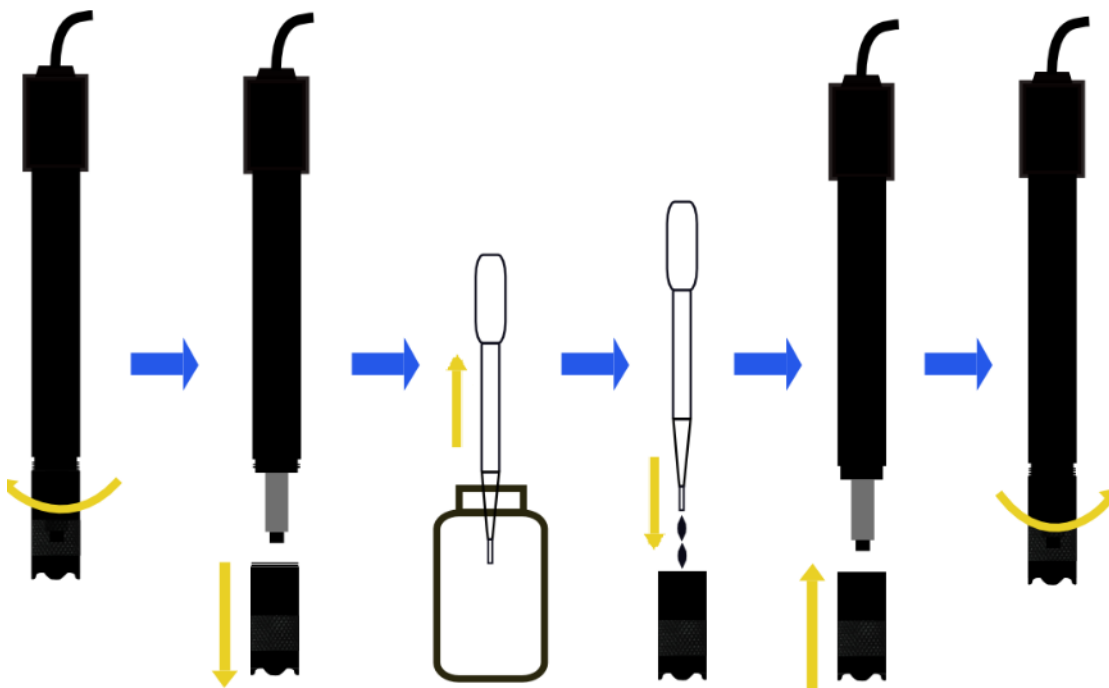


Figura 4.12: Ilustração sobre a forma de como a solução deve ser inserida na tampa da sonda. [71]

Após o processo de inserção da solução na sonda, é necessário que se realize o processo de calibração. Ela pode ser realizada através de dois processos, o primeiro garante que a sonda seja capaz de mensurar valores corretos em ambientes com temperatura estável, já o segundo permite que o sistema calcule a compensação de temperatura, quando a sonda

é utilizada em ambientes com temperatura variável.

Caso se deseje realizar o primeiro processo de calibração, é necessário manter a sonda úmida ao ar, preparar um copo com água purificada e agitar essa água (com auxílio de um agitador/batedor) por 10 minutos, de forma que o oxigênio dissolvido do meio fique saturado. Após esse processo, deve-se parar de agitar a água e esperar que as bolhas desapareçam, para que assim a sonda seja inserida no meio. Depois de colocar a sonda, é necessário que se continue mexendo lentamente (evitando bolhas) e esperar até que a tensão de saída esteja estável. Quando isso ocorrer, deve-se registrar a temperatura e a tensão lida pela sonda e alterar as variáveis 'CAL1V' e 'CAL1T' presentes no código definido pela fabricante.

Já para o segundo processo de calibração, é necessário preparar dois copos contendo água purificada em temperaturas distintas, uma com temperatura abaixo dos 20 °C e outra com uma temperatura próxima aos 38 °C (Não ultrapassar os 40 °C, para não danificar a sonda). O resto do procedimento é semelhante ao do primeiro processo, deve-se agitar a água em cada um dos copos durante o mesmo período de tempo, esperar que as bolhas desapareçam e só assim inserir a sonda no meio. Também nesse processo, depois de se colocar a sonda, é necessário que continue mexendo-a lentamente (evitando bolhas) e esperar até que a tensão de saída esteja estável. Quando isso ocorrer, deve-se registrar a temperatura e a tensão lida pela sonda e alterar no código as variáveis 'CAL1V' e 'CAL1T', caso o copo esteja com a água medindo 38°C, ou 'CAL2V' e 'CAL2T' caso o copo esteja com a água medindo menos que 20 °C. Após medir a voltagem do oxigênio dissolvido saturado em duas temperaturas diferentes, é possível obter uma curva de compensação de temperatura, visto que a correspondência entre a temperatura e o oxigênio dissolvido saturado na pressão atmosférica padrão é conhecida (Figura 4.13 e 4.14).

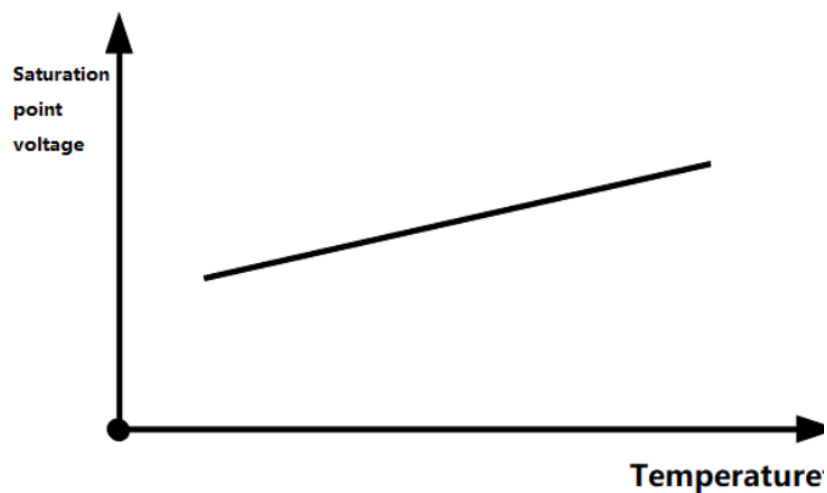


Figura 4.13: Gráfico com a relação do valor de tensão do ponto de saturação e a temperatura. [71]

T °C	DO mg/L	T °C	DO mg/L	T °C	DO mg/L
0	14.60	16	9.86	32	7.30
1	14.22	17	9.64	33	7.17
2	13.80	18	9.47	34	7.06
3	13.44	19	9.27	35	6.94
4	13.08	20	9.09	36	6.84
5	12.76	21	8.91	37	6.72
6	12.44	22	8.74	38	6.60
7	12.11	23	8.57	39	6.52
8	11.83	24	8.41	40	6.40
9	11.56	25	8.25	41	6.33
10	11.29	26	8.11	42	6.23
11	11.04	27	7.96	43	6.13
12	10.76	28	7.83	44	6.06
13	10.54	29	7.68	45	5.97
14	10.31	30	7.56	46	5.88
15	10.06	31	7.43	47	5.79

Figura 4.14: Tabela contendo a relação entre o teor de oxigênio dissolvido e a temperatura do meio. [71]

4.1.3 Demais módulos

Módulo data/hora

Conforme arquitetado no diagrama de Hardware (Figura 4.2), o cabo 'Vcc' deve ser conectado a saída de 5V presente no Arduino e os cabos de transmissão de dados, SCA e SCL, conectados respectivamente nas entradas de comunicação 20 e 21. Após a conexão ser realizada, deve-se executar o código definido pela fabricante [72] e alterar manualmente os campos 'TIME YR', 'TIME MTH', 'TIME DATE', 'TIME HR', 'TIME MIN', 'TIME SEC' para (respectivamente) o ano, mês, dia, hora, minutos e os segundos do momento da conexão.

Para verificar se a bateria do dispositivo está funcionando da forma correta, deve se desconectar a fonte de alimentação do arduino, esperar um intervalo de tempo, e após isso reconectar a fonte de alimentação no Arduino. Caso a bateria do módulo esteja carregada, os dados exibidos terão sido atualizados automaticamente, sendo desnecessário o reajuste manual.

Módulo armazenamento interno

Conforme arquitetado no diagrama de Hardware (Figura 4.2), o cabo 'Vcc' deve ser conectado a saída de 5V presente no Arduino e os cabos de transmissão de dados, MISO, MOSI, SCK e SS, conectados respectivamente nas entradas digitais 50, 51, 52 e 53. Após a conexão ser realizada, deve-se inserir no módulo um cartão de memória do tipo 'MicroSD' e executar o código definido pela fabricante [73], que irá permitir que sejam realizados os testes escrita e leitura de mensagens no cartão de memória inserido.

Módulo de comunicação de dados

Conforme arquitetado no diagrama de Hardware (Figura 4.2), o cabo 'Vcc' deve ser conectado a uma fonte externa com tensão superior a 7 V e inferior a 23 V (Recomenda-se utilizar uma bateria Li-ion 7.4V 2200mAh) e os cabos de transmissão de dados, RX1 (porta 0), TX1 (porta 1) e o Switch de funcionamento do módulo (porta 12), devem ser

conectados respectivamente nas entradas de conexão 0, 1 e 12 do Arduino. Para que o módulo seja capaz de se comunicar com a rede GPRS, é necessário inserir um chip (SIM card) sem bloqueio de PIN e que tenha acesso a rede 2G/3G (Possível através do pagamento do serviço a operadora telefônica escolhida), além de executar o código definido pela fabricante [74].

Durante o processo de execução do código, deve se aguardar o registro do chip na rede, e quando esse processo se finalizar, o LED indicador de rede piscará lentamente a cada 3 segundos. Durante a execução, o usuário deve estar atento em verificar o estado da conexão do chip a rede, sendo necessário que ele escreva no código de execução ou insira manualmente no monitor serial, alguns comandos que serão interpretados pelo módulo de comunicação de dados (Figura 4.15).

O primeiro comando é o 'AT', caso o retorno seja 'OK', significa que o módulo está ligado e a conexão ao arduíno está funcional. Já o comando 'AT + CSQ' verifica a intensidade (em dB) do sinal através da antena, para indicar que o chip está funcional e a antena consegue captar a intensidade necessária de sinal para conexão, o valor retornado necessita ser maior que 5. O comando 'AT + CREG?' verifica se o dispositivo está conectado na rede, caso o segundo dígito retornado pelo módulo seja 1 ou 5 (1 significa conexão em rede doméstica e 5 em rede roaming) é sinal de que o módulo está registrado e conectado a rede. O comando 'AT + COPS =?' retorna a lista de operadoras presentes na rede e o comando 'AT + CBC' retorna o percentual do total e a tensão real da bateria (em mV) que está conectada ao módulo [75].

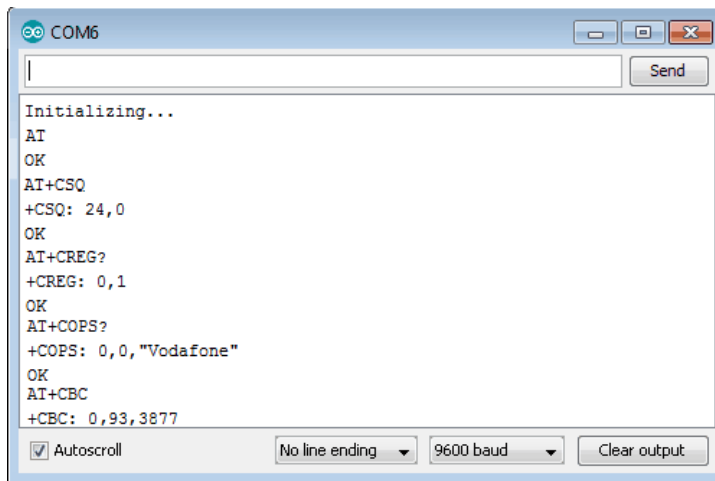


Figura 4.15: Sequencia de comandos inseridos no monitor serial para verificar se o módulo de comunicação está funcionando corretamente. [75]

4.2 Base de Dados

O principal papel da base de dados na solução é o de permitir que todos os dados obtidos pela parte de hardware sejam salvos e disponibilizados para a parte de software, para que o usuário possa visualizá-los. Primeiramente a base de dados receberá os dados das estações encapsulados no formato 'JSON'. A estação (parte de hardware) mandará as informações da seguinte forma: Identificador da estação 'Id Device', Data da mensagem 'Date', senha da estação 'Password', Identificador da variável 'n' (Id Parameter), e valor lido da variável 'n' (Value). Para que esses dados sejam tratados e inseridos na base de dados, foi criada uma API responsável pelo gerenciamento de dados da solução.

A partir da estrutura JSON recebida, a API realiza o seguinte tratamento: Primeiramente ela cria um novo registro na tabela 'Message' (Figura 4.16) e insere os valores Id Device, Date e o Message (Armazenará uma cópia de todo o conteúdo JSON recebido). Após a inserção, a base de dados definirá um ID para o registro 'Message' criado, e a API irá utilizar esse valor para vincular aos registros da tabela 'Message Data'. Segundamente, a API irá tratar separadamente cada valor retornado para cada uma das variáveis presentes na mensagem. Sendo assim, ela irá inserir um registro na tabela 'Message Data' com o Id Message, definido no processo anterior, e também com a identificação do parâmetro

(Id Parameter 'n') e seu respectivo valor (Value 'n').

Para que esses dados sejam inseridos na base de dados, também é necessário que o usuário crie seu registro, cadastre a estação, vincule as variáveis e defina a regra de análise que será aplicada a mesma. Todo esse processo será realizado na aplicação desenvolvida em Flutter. O processo de registro de conta e login de usuário utilizará a funcionalidade 'Auth' do Firebase. Através desse processo, o usuário passará seu 'e-mail' e 'senha' na tela 'NewUserScreen' (Figura 4.24) e caso não exista nenhum outro registro vinculado ao e-mail informado, a funcionalidade registrará a conta no Firebase e retornará a aplicação um identificador único (Id) vinculado a conta do usuário.

Já para armazenar os outros campos informados pelo usuário na tela de cadastro, a API irá criar um registro na tabela 'User', inserindo um registro com o Id único (ID User) retornado pelo Firebase, nome (NameUser), sobrenome (SurnameUser), E-mail (Mail), data de aniversário (BirthDate), cidade (IdCity), senha (Password) e a data de registro (DateRegister).

No processo de autenticação do usuário (Através da tela 'FirstScreen' - Figura 4.23), a funcionalidade Auth também será utilizada. Com os valores de 'e-mail' e 'senha' preenchidos, a funcionalidade visualiza os registros de conta presentes no Firebase e retorna o identificador único (UserId) do usuário, caso os dados estejam corretos.

Após o processo de autenticação ser realizado, o usuário tem a possibilidade de registrar uma nova estação no sistema. Para isso ele deverá acessar a tela 'StationsScreen' e clicar no botão 'Insert Station', que o redirecionará a tela 'AddStationsScreen'. Ao preencher todos os dados requeridos nessa tela e pressionar o botão 'Insert', a API irá inserir um novo registro na tabela 'Device', que conterá os valores: Nome do dispositivo (NameDevice), coordenadas da localização (LocationCoordinate), cidade (IDCity), tabela de regras a ser considerada (IdRulesTable), senha (Password), status do dispositivo (Status) e a data de registro (DateRegister).

Através do registro do 'Device' e o retorno do ID vinculado a ele, a API insere o registro de cada uma das variáveis a serem vinculadas ao dispositivo na tabela 'DeviceParameters'. Nesse registro, serão inseridos o identificador do dispositivo que será vinculado (IdDevice),

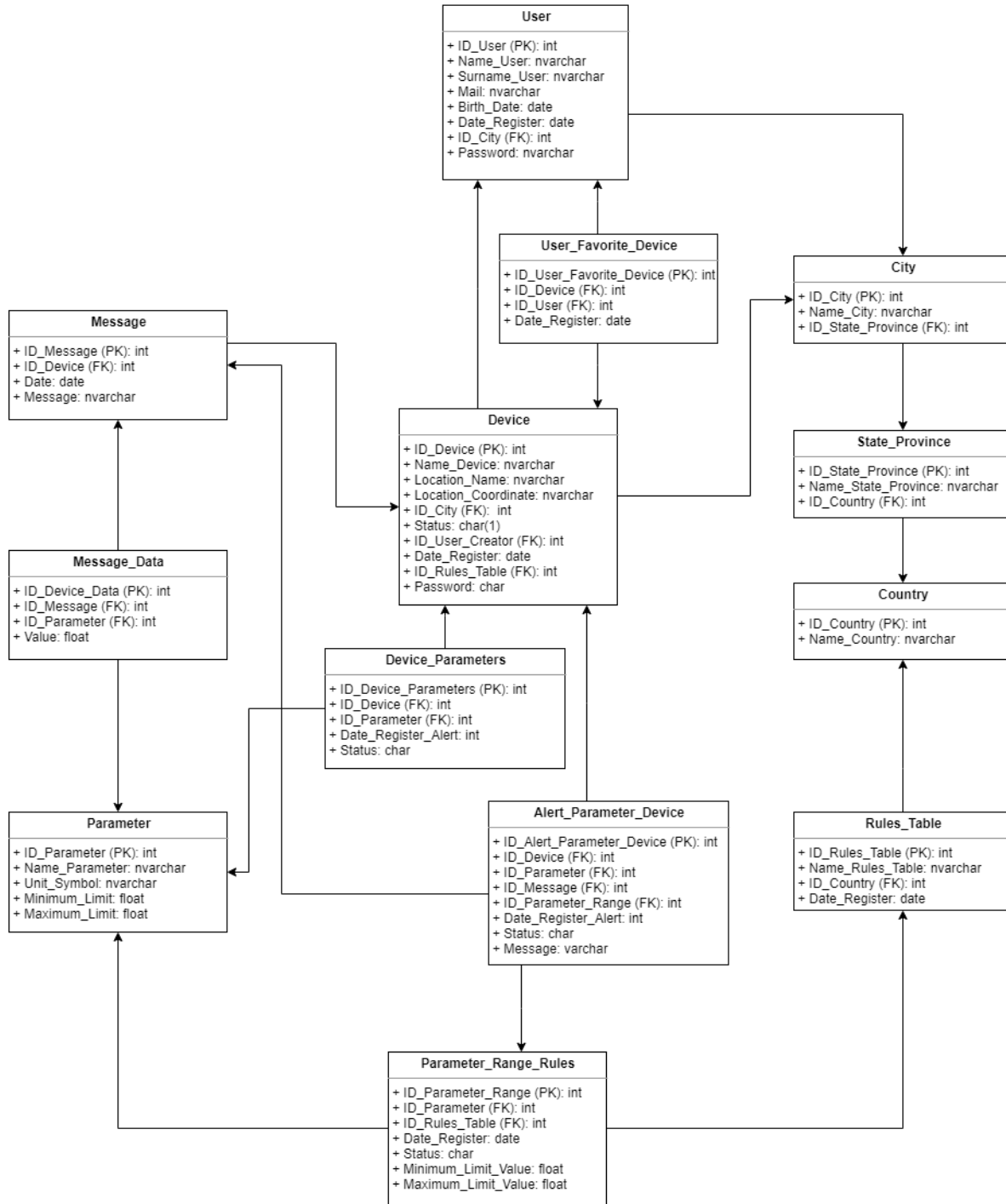


Figura 4.16: Diagrama geral da Base de Dados. [65]

o identificador da variável que será informada pelo dispositivo (`IdParameter`), o status desse vínculo (`Status`) e a data de registro (`DateRegister`).

Caso o usuário deseje favoritar alguma das estações já registradas, ele deverá ir na tela `'StationProfileScreen'` e clicar no botão `'Favorite Station'`. Após essa ação, a API irá criar um novo registro na tabela `'UserFavoriteDevice'`, que conterà o identificador do usuário (`IdUser`), o identificador da estação (`IdDevice`) e a data de registro da ação (`DateRegister`).

Outra tabela importante é a denominada `'Parameter'`, será nela que ficarão registradas todas as variáveis possíveis de serem registradas pelas estações. Cada registro nessa tabela, conterà um identificador próprio (`IdParameter`), um nome (`NameParameter`), um símbolo de unidade (`UnitSymbol`) - por exemplo, em % no parâmetro `'umidade'` e um valor mínimo e máximo possível - por exemplo, 0 e 14 no parâmetro `'pH'` (`MinimumLimit` e `MaximumLimit`).

Para que sejam definidas as regras, nas quais uma faixa de valores das variáveis é definida, é necessário criar os registros na tabela de regras (`RulesTable`). Cada registro dessa tabela conterà um identificador único (`IdRulesTable`), um nome para a regra (`NameRulesTable`), um apontamento para o país no qual ela é válida (`IdCountry`), o usuário criador da regra (`IdUserCreator`), a data de registro (`DateRegister`) e a situação da regra (`Status`).

Após o escopo geral da regra ser definido, é necessário cadastrar os intervalos limites para cada uma das variáveis nas quais a regra cadastrada se aplica. Esse cadastro é feito na tabela `'ParameterRangeRules'`, que conterà diversos valores, como: Um identificador próprio (`IdParameterRange`), o parâmetro (`IdParameter`) - no qual a regra se aplicará, a tabela de regras (`IdRulesTable`) - na qual a regra está vinculada, e o valor mínimo e máximo aceitável (`MinimumLimitValue` e `MaximumLimitValue`).

Caso o valor do parâmetro retornado pela estação (contido na tabela `'DeviceData'`), ultrapasse os valores máximos ou mínimos, definidos pela tabela `'ParameterRangeRules'` e/ou `'Parameter'`, um registro de alerta é criado e será mostrado aos usuários que seguem a estação. Esse registro de alerta é feito dentro da tabela `'AlertParameterDevice'`, e a cada novo registro são inseridos dados como: Identificador do dispositivo (`IdDevice`) - no qual o

alerta será vinculado, Identificador do parâmetro (IdParameter) - que ultrapassou a faixa de valores, o Identificador da mensagem (IdMessage) - no qual os valores ultrapassaram os limites, o identificador da regra (IdParameterRange), o status do alerta (Status), a mensagem de alerta (Message) e a data de registro (DateRegisterAlert).

Se o valor do parâmetro ultrapasse o limite definido na tabela 'ParameterRangeRules', a mensagem (Message) a ser registrada será 'Valor limite excedido' (Limit Value Exceeded). Caso o valor ultrapasse o limite definido na tabela 'Parameter', ou caso o valor do parametro não seja retornado na mensagem, a mensagem (Message) a ser registrada será 'Problema no sensor' (Sensor problem).

Já a tabela 'City', ficará responsável por armazenar os registros de todas as cidades nas quais os dispositivos e/ou usuários podem estar presentes. Para cada registro, existirá um identificador único (IdCity), um nome (NameCity) e um estado/província/distrito (IdStateProvince). A tabela 'StateProvince' armazenará os registros de estado/província/distrito, referentes a cada uma das cidades, onde cada registro dessa tabela possuirá um identificador único (IdStateProvince), um nome (NameStateProvince) e um país vinculado (IdCountry). E a tabela 'Country', irá conter o relação dos países, onde cada registro possuirá um identificador único (IdCountry) e um nome (NameCountry).

4.3 API

Para centralizar as informações e permitir que o compartilhamento de dados fosse feito entre a parte de software, a parte de hardware e o banco de dados, foi utilizada uma API RESTful para troca de mensagens, seguindo o modelo REST (Representational State Transfer), este que permite realizar operações através dos métodos GET (Busca), POST (Registro), PUT (Atualização) e DELETE (Exclusão). Para a criação dessa API foi utilizado o framework NestJS, que facilita a criação de aplicações em Node.js. Já para a comunicação com a base de dados, foi utilizado o TypeORM, um recurso existente do Nest.js.

O ORM (Mapeamento de objeto relacional) permite a interação com o banco de dados

relacional, utilizando-se um schema definido por módulos (Representando a orientação a objetos), conforme mostrado na Figura 4.17. E na API, dentro de cada um desses módulos, existem três tipos de scripts principais: Entity, Controller e Service.

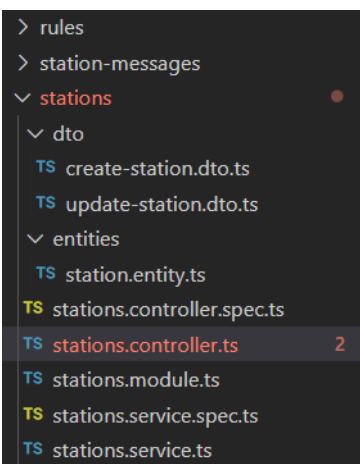


Figura 4.17: Módulos da API. [65]

O primeiro denominado 'Entity' define todos os campos presentes no objeto da base de dados que o módulo representa (Figura 4.18). Abaixo iremos utilizar como exemplo o objeto 'Stations', no código a primary key 'ID' é instanciada como uma variável do tipo 'PrimaryGeneratedColumn'. Já as variáveis do tipo foreign key, devem ser instanciadas com o tipo de relação que a tabela analisada faz com as outras - 'ManyToMany' para relações do tipo n:n, 'ManyToOne' para relações do tipo n:1 e 'OneToMany' para relações do tipo 1:n. Em relações do tipo 'ManyToOne' é necessário instanciar o 'JoinColumn', que irá relacionar a primary key da outra tabela com o campo referenciado na tabela analisada. Em relações do tipo ManyToMany é necessário instanciar um 'JoinTable', que permitirá que uma lista de ligações entre ambas as tabelas sejam armazenadas e estruturadas na base de dados. Já em relações do tipo OnetoMany, é necessário instanciar apenas a outra tabela que irá puxar os dados da tabela analisada. Para as demais variáveis presentes na tabela, basta definir o tipo da variável, da mesma forma que está registrada na base de dados.

```

@Entity()
export class Station extends AbstractModel<Station> {
  @PrimaryGeneratedColumn()
  id?: number;

  @Column()
  name?: string;

  @Column()
  country?: string;

  @Column()
  city?: string;

  @Column()
  coord_lat?: string;

  @Column()
  coord_long?: string;

  @ManyToOne(type => Rule, {nullable: false})
  @JoinColumn({ name: "rule_id"})
  rule?: Rule

  @ManyToMany(type => Variable, variable => variable.stations)
  @JoinTable({
    name: 'station_variables',
    joinColumn: {name: 'station_id'},
    inverseJoinColumn: {name: 'variable_id'}
  })
  variables?: Variable[];

  @ManyToOne(type => User, {
    eager: true
  })
  @JoinColumn({ name: "owner_id"})
  owner?: User;

  @OneToMany(type => StationMessage, station_message => station_message.station)
  messages?: StationMessage;
}

```

Figura 4.18: Script Entity. [65]

O segundo script principal é o 'controller', que realiza o primeiro tratamento dos comandos direcionados pela parte de hardware ou de software a URL (Figura 4.19). Os principais comandos que ela permite realizar são o 'GET', 'POST', 'PUT' e o 'DELETE'. Quando o método 'GET' é instanciado, a API retorna a busca de todos os dados referentes a tabela acessada através da URL (Por exemplo, 'GET' enviado para a URL `http://domain/station`, retorna todos os dados presentes na tabela 'stations'). O método 'POST', faz com que a API crie um novo registro na tabela instanciada, conforme os dados repassados pela parte de hardware ou software que enviaram esse comando. O método 'PUT' realiza o update do registro vinculado ao 'id' repassado ao se instanciar o método, alterando os dados daquele registro na tabela. E o método 'DELETE' é o responsável por deletar o registro vinculado ao 'id' repassado ao instanciar o método.

```
@Controller('stations')
export class StationsController {
  constructor(private readonly stationsService: StationsService) {}

  @Post()
  async create(@Body() createStationDto: CreateStationDto, @Req() request: Request) {
    const user = request['user'];

    if(!user) {
      throw new ForbiddenException();
    }

    const obj = await this.stationsService.create(createStationDto, user);
    return this.stationsService.findOne(obj.id);
  }

  @Get()
  findAll() {
    return this.stationsService.findAll();
  }

  @Get('/:id')
  findOne(@Param('id') id: string) {
    return this.stationsService.findOne(+id);
  }

  @Put('/:id')
  async update(@Param('id') id: string, @Body() updateStationDto: UpdateStationDto) {
    await this.stationsService.update(+id, updateStationDto);
    return this.stationsService.findOne(+id);
  }

  @Delete('/:id')
  remove(@Param('id') id: string) {
    return this.stationsService.remove(+id);
  }
}
```

Figura 4.19: Script Controller. [65]

Todos esses comandos são repassados a base de dados (Repository) através do script 'service', que executa de fato essas operações no banco e vincula os dados presentes na base de dados, as variáveis presentes no módulo da API (Figura 4.21). Os dados de conexão com a base de dados são setados no script 'domain.module' (Figura 4.20).

```
TypeOrmModule.forRoot({
  type: 'mysql',
  host: '████████████████████',
  port: █████,
  username: '██████████',
  password: '██████████',
  database: 'heroku_██████████',
  synchronize: true,
  autoLoadEntities: true,
}),
```

Figura 4.20: Conexão com a base de dados. [65]

```

@Injectable()
export class StationsService {
  constructor(
    @InjectRepository(Station)
    private repository: Repository<Station>,
  ) {}

  create(createStationDto: CreateStationDto, user: User) {
    return this.repository.save({
      ...createStationDto,
      owner: user
    });
  }

  findAll() {
    return this.repository.find();
  }

  findOne(id: number) {
    return this.repository
      .createQueryBuilder(['station'])
      .where({id})
      .select([
        'station',
        'owner',
        'rule',
        'variables',
        'messages',
        'variables_data',
        'variable.id'
      ])
      .leftJoin('station.owner', 'owner')
      .leftJoin('station.rule', 'rule')
      .leftJoin('station.variables', 'variables')
      .leftJoin('station.messages', 'messages')
      .leftJoin('messages.variables_data', 'variables_data')
      .leftJoin('variables_data.variable', 'variable')
      .orderBy('messages.date')
      .getOne();
  }

  findOneWithRule(id: number) {
    return this.repository
      .createQueryBuilder('station')
      .where({id})
      .select([
        'station',
        'rule',
        'rule_variables',
        'variable'
      ])
      .leftJoin('station.rule', 'rule')
      .leftJoin('rule.rule_variables', 'rule_variables')
      .leftJoin('rule_variables.variable', 'variable')
      .getOne();
  }

  async update(id: number, updateStationDto: UpdateStationDto) {
    return this.repository.save({
      id,
      ...updateStationDto
    });
  }

  async remove(id: number) {
    await this.repository.delete(id);
  }
}

```

Figura 4.21: Script Service. [65]

4.4 Software

Para permitir que os dados obtidos pelos sensores sejam visualizados de forma prática e organizada, a aplicação foi subdividida em diversas telas. A figura 4.22 mostra o diagrama geral da aplicação, interligando as diversas telas presentes na aplicação e demonstrando o fluxo que o usuário pode realizar.

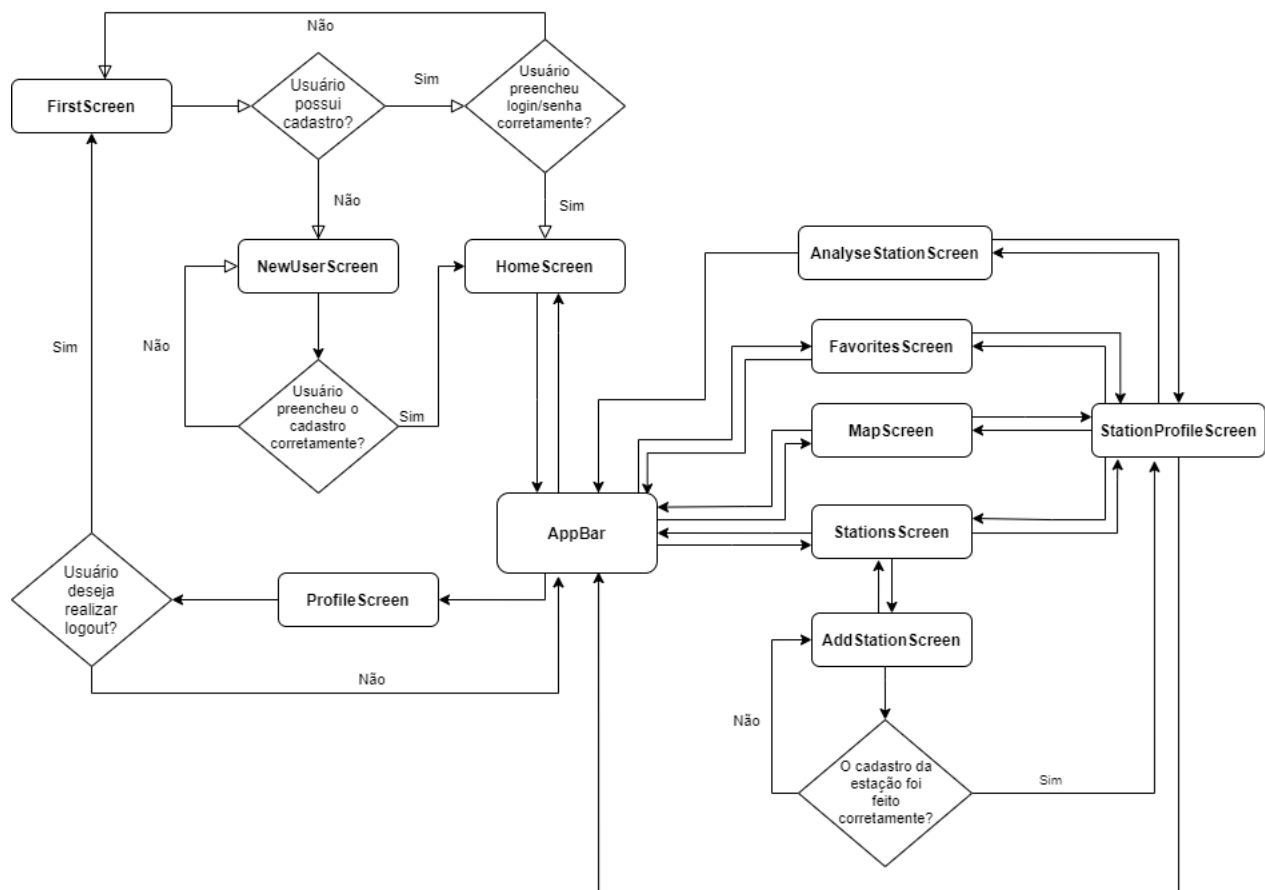


Figura 4.22: Diagrama geral da aplicação. [65]

4.4.1 FirstScreen

Caso o usuário nunca tenha acessado a aplicação, a primeira tela que irá aparecer será a 'FirstScreen' (Figura 4.23). Nessa tela o usuário pode realizar as seguintes ações: Iniciar a sessão com dados cadastrados, recuperar uma senha e registrar uma conta na aplicação.

Caso o usuário já possua uma conta registrada e deseje acessar a aplicação, basta que ele preencha os campos 'E-mail' e 'Password' com seus dados e clique no botão 'Log In', dessa forma os dados serão validados através do Firebase Auth, e caso estejam corretos o usuário será redirecionado para a tela 'HomeScreen'. Se o usuário não lembrar da senha cadastrada, basta ele clicar na opção 'Forgot account' e preencher o seu e-mail, nessa situação ele obterá por e-mail um link que permitirá a alteração da senha.

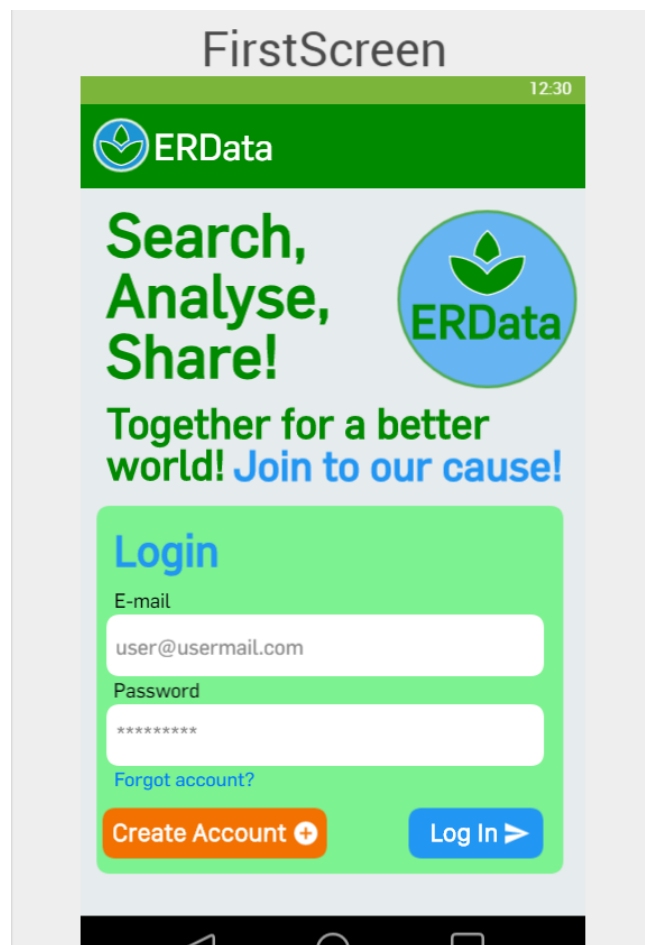
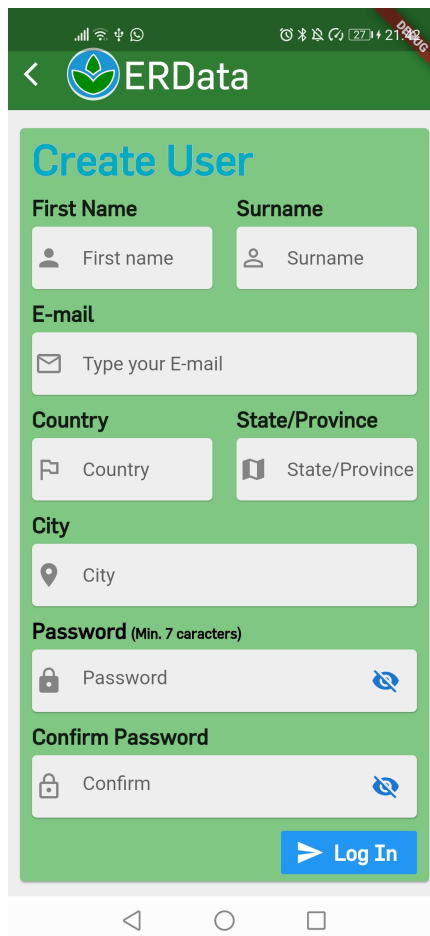


Figura 4.23: Primeira tela da aplicação. [65]

4.4.2 NewUserScreen

Mas se o usuário desejar criar uma conta, ele deverá selecionar a opção 'Create Account' que irá redirecioná-lo a tela 'NewUserScreen' (Figura 4.24). Nessa tela ele deverá informar os seguintes campos: Primeiro Nome ('First Name'), Sobrenome ('Surname'), E-mail ('E-mail'), País ('Country'), Estado/Distrito/Província ('State/Province'), Cidade ('City') e Senha ('Password' e 'Confirm Password'). Caso o usuário registre uma conta que não possua nenhum e-mail já registrado e uma senha que atenda o número mínimo de seis caracteres, a conta será criada e ele será direcionado a tela 'Homescreen'.



The screenshot shows a mobile application interface for creating a new user. At the top, there is a green header with a back arrow, the ERData logo, and the text 'ERData'. Below the header, the title 'Create User' is displayed in blue. The form consists of several sections: 'First Name' and 'Surname' (each with a person icon and a text input field), 'E-mail' (with an envelope icon and a text input field), 'Country' and 'State/Province' (each with a location icon and a text input field), 'City' (with a location pin icon and a text input field), 'Password (Min. 7 characters)' (with a lock icon, a text input field, and an eye icon for visibility), and 'Confirm Password' (with a lock icon, a text input field, and an eye icon for visibility). At the bottom right, there is a blue button with a right-pointing arrow and the text 'Log In'. The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps icons.

Figura 4.24: Tela para cadastro de um novo usuário. [65]

4.4.3 Homescreen

Nos casos onde o usuário já esteja logado no aplicativo, a primeira tela que aparecerá será a 'HomeScreen' (Figura 4.25). É obrigatório que o usuário já esteja logado para visualizar essa tela, pois será nela que ele poderá ter informações gerais sobre suas estações (Criadas e/ou favoritas). A primeira informação que essa tela permite mostrar ao usuário é a quantidade de estações que estão funcionando com ou sem alerta.

Os alertas podem ser definidos tanto quando a estação perde a conexão com a rede, quanto nos momentos em que algum dos parâmetros analisados pelos sensores ultrapassa a faixa de valores pré definidos. Quando um novo alerta é gerado, o sistema gera uma nova notificação que é exibida ao usuário na tela.

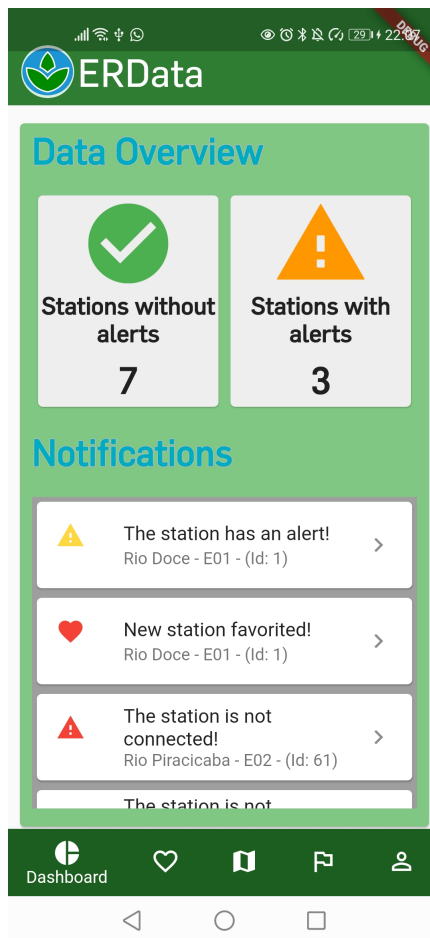


Figura 4.25: Tela de exibição da situação das estações criadas/favoritadas pelo usuário. [65]

4.4.4 MapScreen

Através da tela 'MapScreen' (Figura 4.26) o usuário pode visualizar as estações próximas a ele pelo mapa. Cada estação ficará registrada conforme a coordenada GPS definida na criação da estação, nessa coordenada ficará registrado o ponto que ficará vinculado a estação. Para localizar as estações pelo mapa, é necessário que o usuário clique no 'drop box' presente no campo 'Search station', para que assim o ponto onde a estação selecionada fique plotado no gráfico e através do mesmo, o usuário pode tanto verificar mais informações sobre a estação, quanto fazer com que uma rota seja definida entre a posição na qual ele se encontra e a estação escolhida.



Figura 4.26: Tela de exibição das estações próximas ao usuário, através de um mapa. [65]

4.4.5 StationsScreen

Outra forma que o usuário pode ter para localizar uma estação é acessando a tela 'StationsScreen' (Figura 4.27), que retornará toda a lista de estações registradas na aplicação. Essa lista pode ser filtrada através do nome, Id e localização da estação. Nessa tela o usuário também pode visualizar as estações criadas por ele, ao clicar no botão 'My Stations' (Figura 4.28), ou também inserir novas estações, através do botão 'Insert Station'.

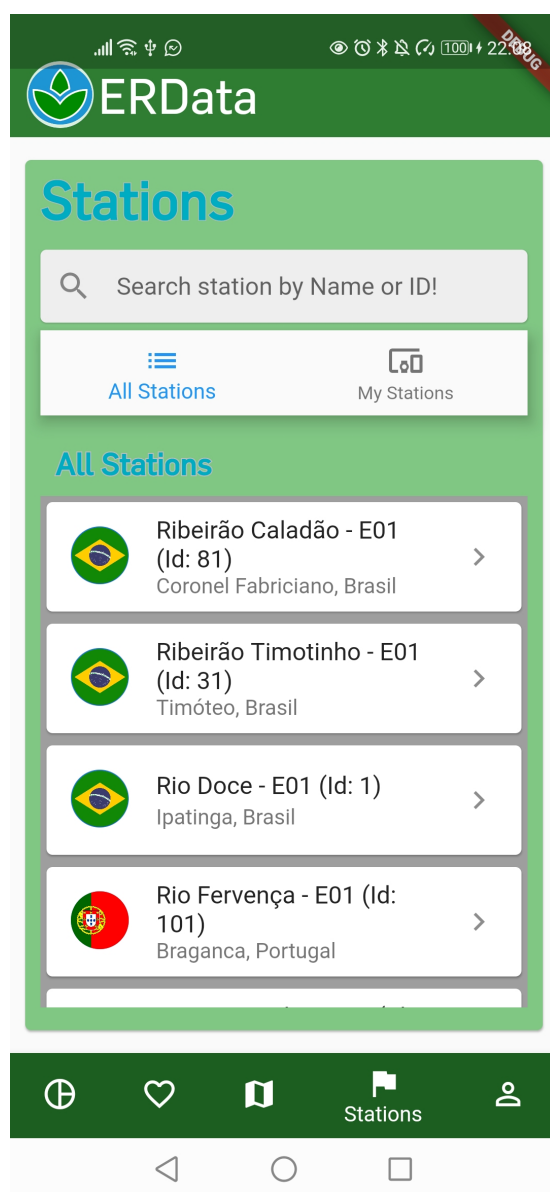


Figura 4.27: Tela de exibição da lista de todas as estações registradas. [65]

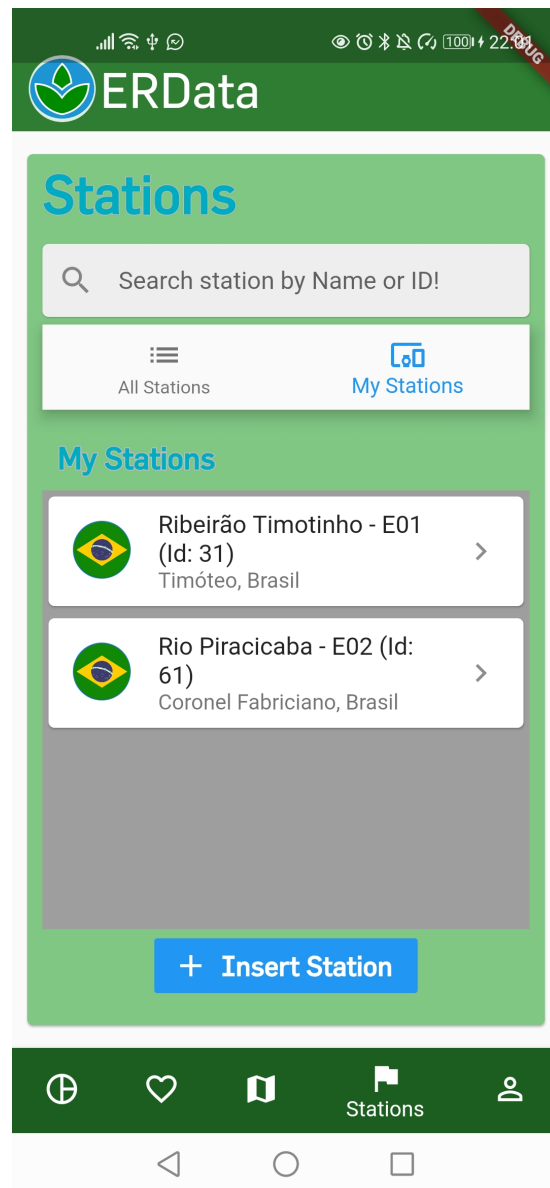


Figura 4.28: Tela de exibição da lista de estações criadas pelo usuário. [65]

4.4.6 AddStationScreen

Ao se clicar no botão 'Insert Station' o usuário é redirecionado a tela 'AddStationScreen'. Para adicionar uma nova estação, o usuário obrigatoriamente deve definir os campos: Nome da estação ('Station Name'), Tabela de regras ('Rules Table'), Coordenadas ('Coordinates'), País ('Country'), Estado/Provincia/Distrito ('State/Province'), Cidade ('City') e uma senha ('Password' e 'Confirm Password') (Figura 4.29). Essas informações são necessárias para que outros usuários possam localizar a estação criada. Nessa tela o usuário também deve definir as variáveis que serão obtidas através da estação. De forma que, através do campo 'Variable ID', os usuários possam fazer com que a estação envie ao banco de dados, as informações de cada uma das variáveis obtidas com os 'Variable's ID' corretos.

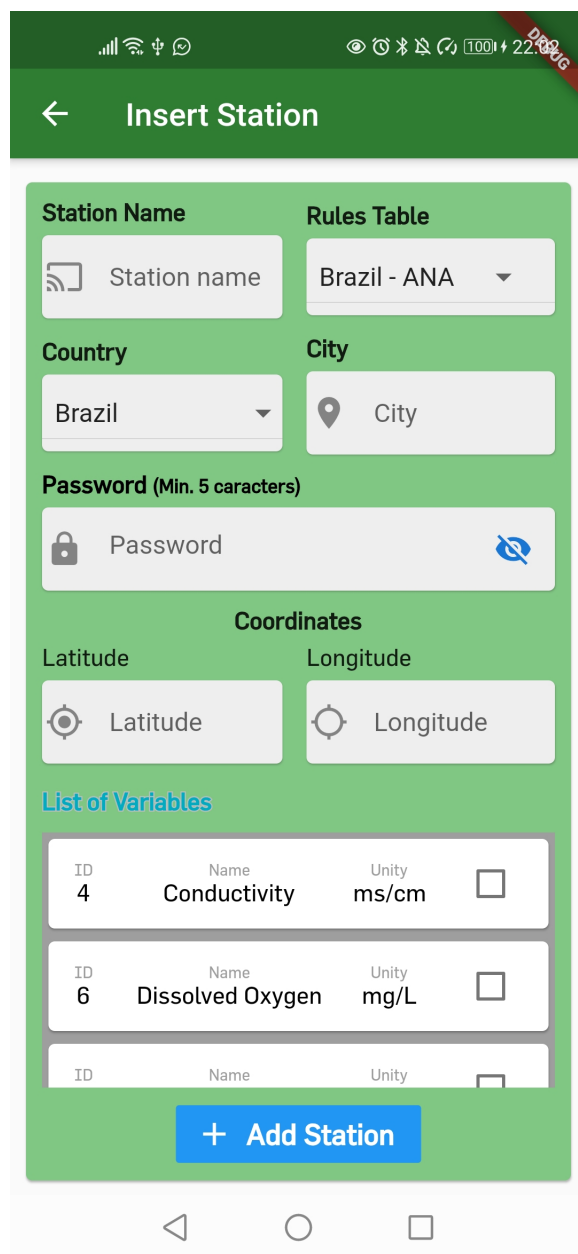


Figura 4.29: Tela de adiç3o de uma nova estaç3o. [65]

4.4.7 StationProfileScreen

Após o processo de criação da estação, os usuários podem visualizar os dados gerais da mesma, através da tela 'StationProfileScreen'. Essa tela contém os campos: Identificador ('ID'), Nome da estação ('Station Name'), Cidade ('City') e País ('Country'). Além dessas informações gerais sobre a estação, o usuário também consegue visualizar quais são as variáveis ('Variables') que são obtidas pelos sensores daquela estação (Figura 4.30).

Após a análise geral da estação, o usuário visualizará dois botões, o primeiro permite que se favorite aquela estação ('Favorite Station'), fazendo que aquela estação apareça na tela 'Favorite' e caso os dados vinculados a essa estação ultrapassem os valores limites, o usuário receba a notificação. Já o segundo botão, permite que o usuário possa analisar, de forma detalhada, os dados gerados por aquela estação. Ao se clicar nesse botão, o usuário é direcionado a tela 'AnalyseScreen'.

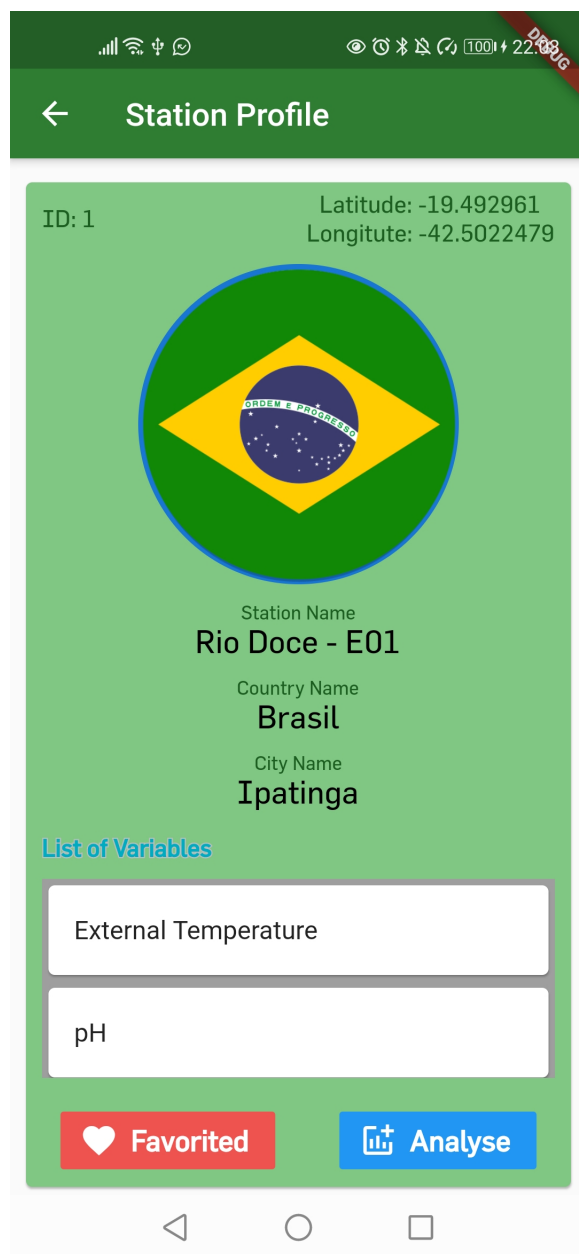


Figura 4.30: Tela de visualização geral da estação escolhida. [65]

4.4.8 FavoriteScreen

A tela 'FavoriteScreen' mostra a lista de estações favoritadas pelo usuário. Por meio dessa tela, o usuário consegue visualizar de forma simples todas as estações favoritadas por ele, podendo acompanhar facilmente os dados obtidos por elas. Nessa tela, o usuário conseguirá visualizar a lista de estações da mesma forma que na tela 'StationsScreen', a diferença estará na quantidade de estações retornadas - Serão exibidas apenas as favoritadas (Figura 4.31).

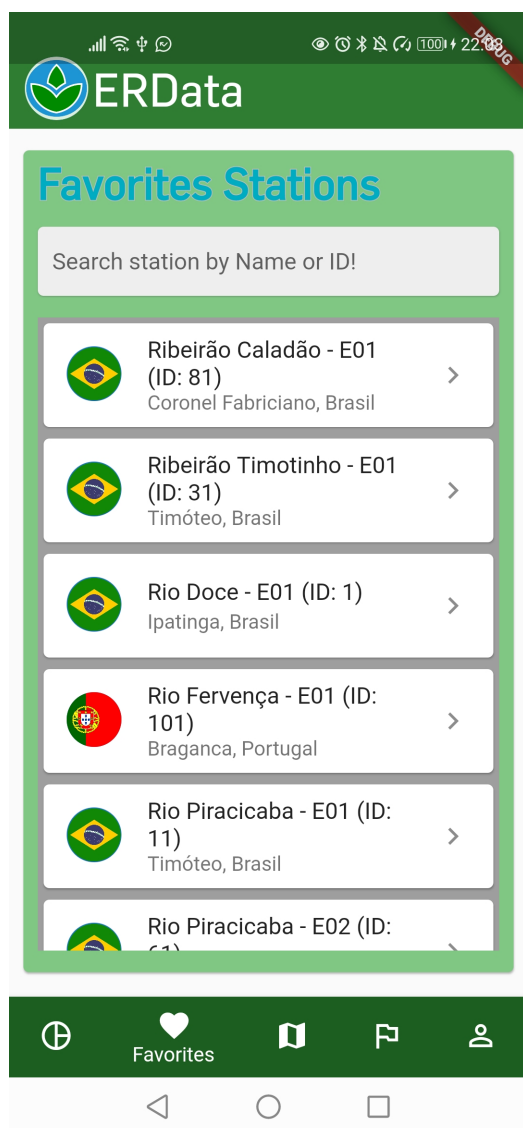


Figura 4.31: Tela de exibição das estações favoritadas pelo usuário. [65]

4.4.9 AnalyseScreen

Já a tela 'AnalyseScreen', mostrará as informações obtidas pela estação de uma forma mais detalhada ao usuário. Primeiramente, ela exibirá um gráfico ('Chart') que mostrará ao usuário a evolução dos dados relacionados a cada uma das variáveis ao longo do tempo. O usuário, por sua vez também poderá baixar a tabela com todos os dados em excel (Atraves do 'View Complete Chart'), visualizar o gráfico em outra aplicação e por meio desta, compartilhar os dados por e-mail (Figura 4.32).

Nessa tela, o usuário também poderá visualizar individualmente, os dados enviados por cada mensagem submetida pela estação. Primeiramente, ele poderá visualizar os dados gerais da mensagem, como ID, a data de envio ('Date'), e se algum dos parametros contidos dentro daquela mensagem, ultrapassou ou não os limites definidos. Ao se clicar em uma dessas mensagens, o usuário pode visualizar o valor retornado por cada uma das variáveis ('Value'), o ID ('Variable ID'), o nome ('Name') e a unidade ('Unity'). Caso o valor da variável esteja dentro dos limites, será exibido um 'check' na cor verde, caso contrário, se exibirá um alerta, contendo a descrição do tipo de alerta vinculado ao valor obtido.

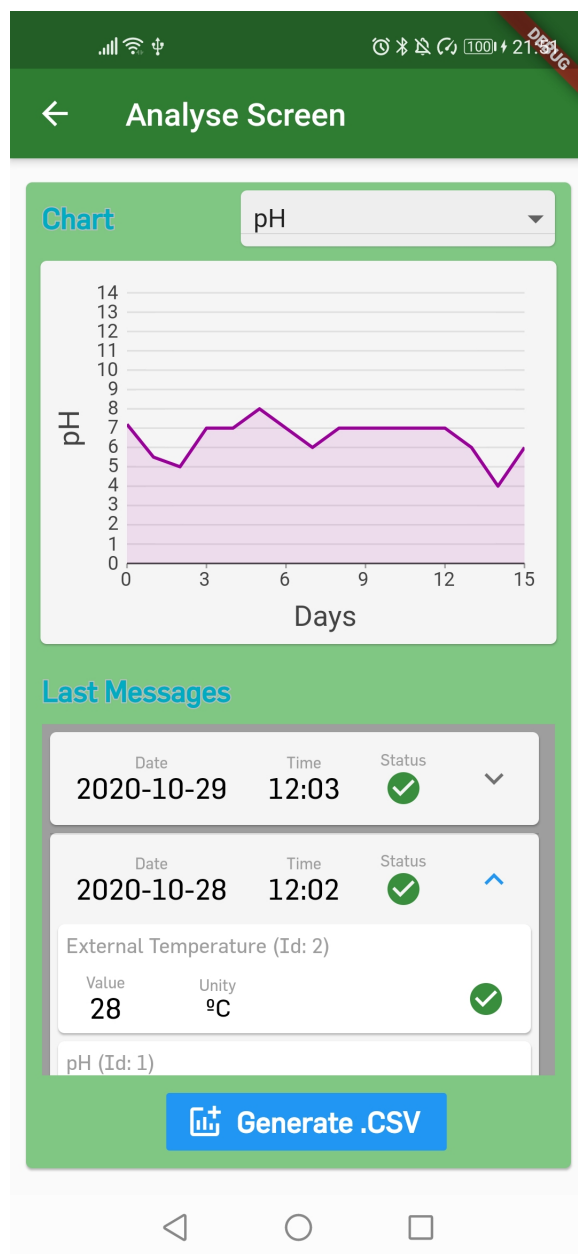


Figura 4.32: Tela de análise dos dados registrados pela estação escolhida. [65]

4.4.10 ProfileScreen

A tela 'ProfileScreen' possibilitará que o usuário possa visualizar informações sobre seu próprio perfil, como 'ID', data de criação 'User', Nome 'User Name', país 'Country', 'E-mail', número total de estações criadas 'Created Stations' e número de estações favoritadas 'Favorite Stations'. Através dessa tela o usuário, conseguirá alterar seus dados e também realizar o logout da aplicação (Figura 4.33).

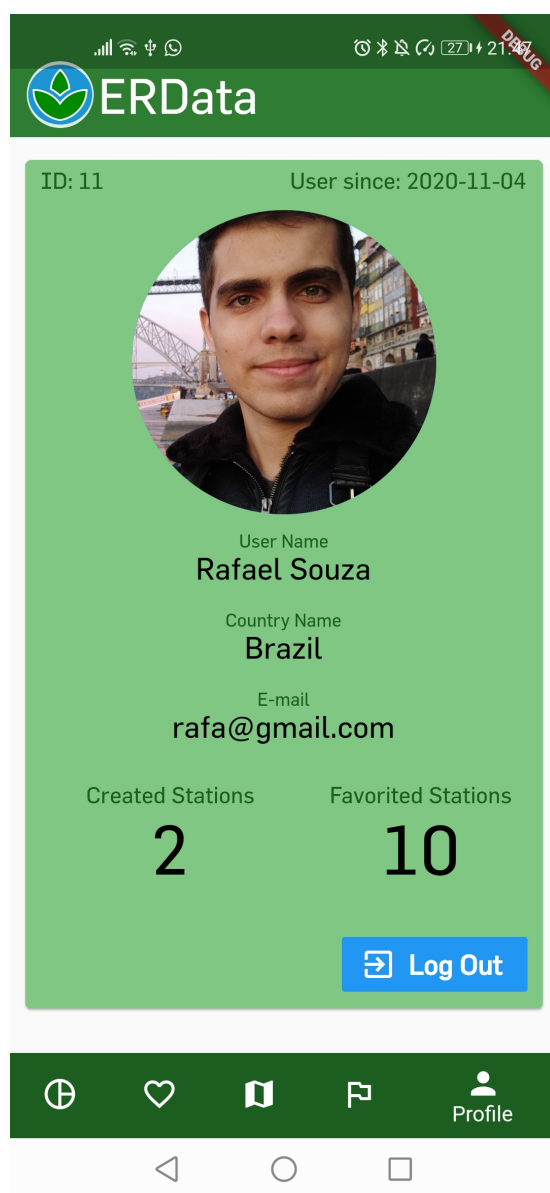


Figura 4.33: Tela de exibição dos dados do usuário. [65]

Capítulo 5

Conclusões

Em primeiro lugar quero dizer que considero cumpridos os objetivos que me havia planejado para o projeto e, visto que além de ter estendido meus conhecimentos de programação, o desenvolvimento desse projeto também aumentou minha habilidade de análise e tomada de decisões, sendo esta a primeira vez que realizo o desenvolvimento completo de um projeto desse tamanho, incluindo o desenvolvimento de um aplicativo. Mesmo que nunca eu tenha programado para o sistema operativo Android, nem utilizado o framework Flutter e a plataforma Firebase, foram os conhecimentos prévios de Java que aprendi durante meu curso de graduação em conjunto com a orientação dos meus orientadores, que me auxiliaram a realizar a tarefa de desenvolver a aplicação.

Também cabe destacar que parte do êxito ao cumprir os objetivos do projeto foram graças a utilização do arduino para a simulação da troca de dados via GPRS, e sua capacidade de simulação. Visto que, devido a pandemia, não conseguimos obter os sensores para realizar a leitura real, mas que com o estudo das documentações de cada um dos sensores, conseguimos simular de forma mais próxima do real o envio de dados das variáveis para a base de dados.

A respeito do uso do Firebase, me pareceu uma plataforma muito útil para aplicações como a que desenvolvi, principalmente pelo fato do processo de autenticação ser simples e robusto. Por mais que o projeto tenha utilizado em boa parte o armazenamento de dados no sistema de gerenciamento MySQL devido o projeto ser estruturado de forma estrutural,

a aplicação não seria segura sem a utilização do Firebase. Espero ter a oportunidade de poder voltar a utiliza-la no futuro em projetos profissionais.

Já a respeito das dificuldades encontradas durante o processo de desenvolvimento do projeto, tenho que destacar a complexidade de ser ligar a parte de hardware e a parte de software ao banco. Visto que na maior parte do projeto, tentei de forma equivocada, utilizar a intermediação do envio de dados da aplicação através do protocolo MQTT e da utilização de instâncias AWS para hospedar os dados. Apenas na parte final do projeto que verificamos a inviabilidade dessa implementação e a 'facilidade' dessa troca de dados através do desenvolvimento de uma API em NestJS para a comunicação entre o MySQL e a plataforma Heroku, que disponibilizou os dados em JSON para a leitura de dados da aplicação.

Como possível extensão do projeto, recomendo que seja realizada a compra e montagem dos sensores conforme a arquitetura definida nesse projeto, em conjunto com testes em campo para agregar os dados reais obtidos com a base de dados e aplicação móvel desenvolvida. Também acredito que o projeto teria um maior valor agregado, com a aplicação de um análise com Inteligência Artificial, de forma que o próprio Software possa analisar os dados obtidos, e através de um histórico de contaminações com motivações identificadas e dados antigos armazenados, classifique os tipos de contaminação que possam estar afetando momentaneamente o ambiente analisado.

Com relação a viabilidade comercial da aplicação, penso que o software desenvolvido, deve ser disponibilizado de forma gratuita para os usuários. De forma, que idenpendente das condições financeiras dos usuários, eles possam auxiliar no acompanhamento, análise e até mesmo cobrança dos órgãos públicos, com relação a possíveis contaminações a serem identificadas pelas estações de hardware presentes nos ambientes aquáticos. O custo maior, seria por parte dos responsáveis por comprar os componentes e montar a estação para leitura de dados, visto que além do custo inicial dos sensores, microcontrolador e estrutura, os implantadores devem se atentar com relação a alimentação e manutenção dos componentes, além do plano de GPRS que deve ser pago para permitir o envio remoto de dados.

O auge atual da utilização da tecnologia em tarefas do cotidiano (Principalmente aplicações móveis) aumentou consideravelmente a demanda de programadores, analistas e engenheiros. Em paralelo, ocorre também o crescimento da interferência do ser humano no meio ambiente aliado a falta de fiscalização por parte dos órgãos públicos, fazendo com que que novas soluções tecnológicas sejam desenvolvidas e aplicadas para que possamos prevenir a destruição lenta e irreversível do meio ambiente. Considero que o desenvolvimento desse projeto, permitiu que eu aumentasse não só meus conhecimentos de programação e arquitetura, como também, ver que com certa dedicação ainda é possível se desenvolver ferramentas e projetos que possam auxiliar as pessoas a visualizar de forma simples e portátil, a situação atual dos ecossistemas aquáticos em sua volta. Sem dúvida, recomendaria a outros estudantes e pesquisadores, a exploração e desenvolvimento de propostas de pesquisa relacionadas a área ambiental, visto o impacto dos mesmos a nossa geração atual e nas futuras.

Bibliografia

- [1] P. B. NETTO, “Sustentabilidade ameaçada”, *Olhares sobre o Lago Paranoá. Brasília, DF*, 2001.
- [2] J. F. ARTIOLA, I. L. PEPPER e M. BRUNSSEAU, “Environmental Monitoring and Characterization”, *San Diego, California: Elsevier Academic Press*, 2001.
- [3] J. F. MARTINELLI, “Meio ambiente: Crescimento populacional, pobreza e degradação ambiental”, *Revista Orlândia Online*, 2012.
- [4] M. POMPÊO, V. MOSCHINI e et al, “Ecologia de reservatórios e interfaces”, *Sao Paulo: Instituto de Biociencias da Universidade de Sao Paulo*, 2015.
- [5] INEMA, *Inema disponibiliza divulgação de dados das unidades de monitoramento remoto*. <http://www.inema.ba.gov.br/2016/09/inema-disponibiliza-divulgacao-de-dados-das-unidades-de-monitoramento-remoto/>, Setembro de 2019.
- [6] IGAM, *Qualidade das águas do Rio Doce após 3 anos do rompimento da Barragem de Fundão*. http://www.igam.mg.gov.br/images/stories/2018/QUALIDADE_DA_AGUA/ENCARTE_Tres_ANOS.pdf, Abril de 2018.
- [7] ANA, *Acesso a informação*. <https://www.ana.gov.br/acesso-a-informacao/institucional>, jun. de 2019.
- [8] S. AHUJA, “Chapter One – Overview: Sustaining Water, the World’s Most Crucial Resource”, *Chemistry and Water*, 2017.

- [9] J. LIU, Y. WANG e et al, “A comprehensive analysis of blue water scarcity from the production, consumption, and water transfer perspectives”, *Ecological Indicators*, 72, 2017.
- [10] E. MARRIS, “‘Water scarcity’ affects four billion people each year”, *Nature News*, Fevereiro de 2016.
- [11] M. MEKONNEN e A. HOEKSTRA, “Four billion people facing severe water scarcity”, *Journal Science Advances*, 2016.
- [12] M. PERA TITUS, V. GARCIA-MOLINA e et al, “Degradation of chlorophenols by means of advanced oxidation processes: a general review”, *Applied Catalysis B: Environmental*, 47, 2004.
- [13] E. D. MINAS, *Barragem em brumadinho tem volume de 1 milhão de m³ em rejeito*. https://www.em.com.br/app/noticia/nacional/2019/01/25/interna_nacional,1024510/barragem-em-brumadinho-tem-volume-de-1-milhao-de-m-de-rejeito-de-mine.shtml, jan. de 2019.
- [14] O. GLOBO, *Acidente em Mariana é o maior da História com barragens de rejeitos*, <https://oglobo.globo.com/brasil/acidente-em-mariana-o-maior-da-historia-com-barragens-de-rejeitos-18067899>, nov. de 2015.
- [15] E. D. MINAS, *Comitês de bacias hidrográficas apontam o desaparecimento de dois córregos após tragédia de brumadinho*, https://www.em.com.br/app/noticia/gerais/2019/01/28/interna_gerais,1025405/comites-de-bacias-hidrograficas-desaparecimento-dois-corregos.shtml, jan. de 2019.
- [16] M. F. FELLIPE, JUNIOR e A. P. M., “Impactos ambientais macroscópicos e qualidade das águas em nascentes de parques municipais em Belo Horizonte – MG”, *Universidade Federal de Minas Gerais*, 2008.
- [17] BRASIL, “Lei n 9.433, de 8 de jan. de 1997. Institui a Política Nacional de Recursos Hídricos, cria o Sistema Nacional de Gerenciamento de Recursos Hídricos, regula 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.”, <http://www2.mma.gov.br/port/conama/legiabre.cfm?codlegi=370>, 1997.
- [18] M. GERAIS, “Deliberação Normativa Conjunta COPAM/CERH-MG nº 01. Dispõe sobre a classificação dos corpos de água e diretrizes ambientais para o seu enquadramento, bem como estabelece as condições e padrões de lançamento de efluentes, e dá outras providências.”, <http://www.siam.mg.gov.br/sla/download.pdf?idNorma=8151>, 2008.
- [19] IGAM, “Rede de monitoramento de qualidade de água superficial em minas gerais.”, <http://portalinfohidro.igam.mg.gov.br/sem-categoria/318-%E1gua-superficial>, 2019.
- [20] FUNASA, *Manual de controle da qualidade da água para técnicos que trabalham em ETAS*, http://www.funasa.gov.br/site/wp-content/files_mf/manualcont_quali_agua_tecnicos_trab_emetas.pdf, 2014.
- [21] H. B. GLASGOW e et al, “Real-time remote monitoring for water quality: a review of current applications, and advancements in sensor, telemetry, and computing technologies.”, *Journal of Experimental Marine Biology and Ecology*, p. 409-448, 2004.
- [22] M. C. F. PINTO, *Manual medição in loco. Site da CPRM, 2007*. http://www.cprm.gov.br/publique/media/gestao_territorial/geologia_medica/manual_medicoes_T_%20pH_0D.pdf, 2014.
- [23] F. Environmental, “*Oxigênio dissolvido*”. *Fundamentos de Medições Ambientais*, <https://www.fondriest.com/environmentalmeasurements/parameters/water-quality/dissolved-oxygen/>, 2013.
- [24] M. V. SPERLING, “Introdução à Qualidade das Águas e ao Tratamento de Esgotos.”, 3ª. ed. *Belo Horizonte: Departamento de Engenharia Sanitária e Ambiental*, v. I., 2005.
- [25] CETESB, “Guia Nacional de Coleta e Preservação de Amostra.”, *Agência Nacional das Águas. Brasília*, p. 51., 2011.

- [26] R. SOUSA, *POTENCIOMETRIA. Departamento de Química – UFJF. Juiz de Fora*. <http://www.ufjf.br/baccan/files/2010/10/Aula-13-POTENCIOMETRIA-2s-2012.pdf>, 2012.
- [27] D. C. HARRIS, “Análise Química Quantitativa”, 6ª Edição, LTC-Livros Técnicos e Científicos Editora S.A., Rio de Janeiro-RJ, 2005.
- [28] P. BERTOLETI, *Medição de turbidez da água com o sensor SKU (MIC324)*. Instituto Newton Braga, 2019. <https://www.newtoncbraga.com.br/index.php/saude-e-meio-ambiente/16379-medicao-da-turbidez-da-agua-com-o-sensor-sku-mic324>, 2019.
- [29] METTLER-TOLEDO, *Description pratique de la manière de mesurer la conductivité*, https://www.mt.com/dam/LabDiv/guides-glen/ph-conductivity-guide/pH_Conductivity_Guide_fr.pdf, 2014.
- [30] P. Vizcaíno, *Temporal River Contamination Measurements with Electrochemical Probes and Mobile Sensor Networks*, https://www.researchgate.net/publication/324858957_Spatio-Temporal_River_Contamination_Measurements_with_Electrochemical_Probes_and_Mobile_Sensor_Networks, 2018.
- [31] Horiba, *Galvanic vs Optical Dissolved Oxygen Sensors*, <https://www.horiba.com/sg/application/material-property-characterization/water-analysis/water-quality-electrochemistry-instrumentation/support/product-selection-support/technical-tips/electrodes/galvanic-vs-optical-dissolved-oxygen-sensors/#:~:text=When%20galvanic%20D0%20sensor%20is,related%20to%20the%20oxygen%20concentration>, 2013.
- [32] ARDUINO, *What is Arduino?*, <https://www.arduino.cc/>, 2019.
- [33] —, *Arduino Mega 2560 REV3*. <https://store.arduino.cc/arduino-mega-2560-rev3>, 2019.
- [34] F. MONSIGNORE, “Sensoriamento de ambiente utilizando o padrão zigbee”, *Dissertação de Mestrado - Universidade de São Paulo, São Carlos*, 2007.

- [35] J. M. MENON, “Dimensionamento de tráfego de rede de dados para sistema celular IXEV-DO”, *Dissertação de Mestrado – Pontifícia Universidade Católica, Campinas*, 2006.
- [36] A. D. B. NOGUEIRA, “Uma proposta de integração das redes UMTS e IEEE 802.11 com suporte a mobilidade”, *Dissertação de Mestrado - Universidade Federal do Ceará, Fortaleza*, 2007.
- [37] 2. STATISTA, *Number of smartphone users worldwide*, <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>, 2020.
- [38] 2. VENTURE BEAT, *Android passes 2.5 billion active devices*, <https://venturebeat.com/2019/05/07/android-passes-2-5-billion-monthly-active-devices/>, 2020.
- [39] 2. SUPER, *Historia do Android, sistema operacional da Google*, <https://super.abril.com.br/galeria/conheca-a-historia-do-android-o-sistema-operacional-mobile-da-google/>, 2020.
- [40] 2. STATISTA, *Global Smartphone marketshare of Android*, <https://www.statista.com/statistics/236027/global-smartphone-os-market-share-of-android/>, 2020.
- [41] E. Coimbra, *Xamarin Forms. Desenvolvimento de Aplicações Móveis Multiplataforma*, <https://www.amazon.com.br/Xamarin-Forms-Desenvolvimento-Aplicacoes-Multiplataforma/>, 2018.
- [42] G. CORREIA, H. ROCHA e S. RISSINO, “Automação de um sistema de irrigação com monitoramento via aplicativo web”, *Revista Engenharia na Agricultura*, Vol.24(4), p.314., 2016.
- [43] M. JORDÃO, “Low-cost automatic station for compost temperature monitoring.”, *Revista Brasileira de Engenharia Agrícola e Ambiental*, v.21, n.11, p.809-813, 2017.

- [44] E. EPSTEIN, *Industrial composting. Environmental engineering and facilities management. New York: Taylor and Francis Group, 2011. 388p*, <https://doi.org/10.1201/b10726>, 2011.
- [45] M. V. DOS SANTOS e et al., *Desenvolvimento de Sistema Automático de Análise de pH e Temperatura da Água para Aquicultura, IX Computer on the Beach*, <https://siaiap32.univali.br/seer/index.php/acotb/article/view/12765>, 2018.
- [46] C. R. SILVEIRA e et al., *O uso do celular no monitoramento do ambiente de piscicultura. X Congresso Brasileiro de Agroinformática. Anais do X SBIAGRO*. http://www.eventos.uepg.br/sbiagro/2015/anais/SBIAGro2015/pdf_resumos/17/17_leandro_henrique_correia_179.pdf, 2015.
- [47] B. BRAGA e et al., “Introdução à Engenharia Ambiental”, *Editora Pearson Educação do Brasil, São Paulo*, 2002.
- [48] SEBRAE., “Manual como Iniciar Piscicultura com Espécies Regionais.”, *Sebrae, Brasília*, pg. 46., 2013.
- [49] L. V. ARANA, “Princípios químicos de qualidade da água em aquicultura: uma revisão para peixes e camarões.”, *Florianópolis, Ed. da UFSC, 231p*, 2004.
- [50] G. MORO, L. TORATI e et al., “Monitoramento e manejo de qualidade da água em pisciculturas.”, *In: Piscicultura de água doce: multiplicando conhecimentos. Brasília, DF Embrapa*, pg. 440, 2013.
- [51] M. L. 2019a, *Arduino Mega 2560 R3 Atmel Mega16u2 + Cabo Usb*. https://produto.mercadolivre.com.br/MLB-947337168-arduino-mega-2560-r3-atmel-mega16u2-cabo-usb-_JM?quantity=1, 2019.
- [52] 2. MERCADO LIVRE, *Módulo Sensor + Ph Eletrodo Sonda Bnc Arduino/phmetro*. https://produto.mercadolivre.com.br/MLB-1042318108-modulo-sensor-ph-eletrodo-sonda-bnc-arduinophmetro-_JM?quantity=1, 2019.

- [53] 2. MERCADO LIVRE, *Sensor De Temperatura - Ds18b20*. https://produto.mercadolivre.com.br/MLB-1196865052-sensor-de-temperatura-ds18b20-prova-dagua-arduino-pic-_JM?quantity=1, 2019.
- [54] 2. MERCADO LIVRE, *Sensor De Temperatura E Umidade Htu21d*. https://produto.mercadolivre.com.br/MLB-980825497-sensor-de-temperatura-e-umidade-htu21d-arduino-_JM?quantity=1, 2019.
- [55] 2. DFROBOT, *Gravity: Analog Electrical Conductivity Sensor*. <https://www.dfrobot.com/product-1123.html>, 2019.
- [56] DFRobot, *Gravity: Analog Dissolved Oxygen Sensor / Meter Kit For Arduino*, https://wiki.dfrobot.com/Gravity__Analog_Dissolved_Oxygen_Sensor_SKU_SEN0237, 2020.
- [57] 2. DFROBOT, *Analog Turbidity Sensor*. <https://www.dfrobot.com/product-1394.html?search=turbidity>, 2019.
- [58] 2. MERCADO LIVRE, *Módulo Relógio Rtc Ds1307*. https://produto.mercadolivre.com.br/MLB-711170864-modulo-relogio-rtc-ds1307-e-at24c32-com-bateria-arduino-_JM?quantity=1, 2019.
- [59] 2. MERCADO LIVRE, *Cabo Adaptador Alimentação Arduino Bateria 9v*. https://produto.mercadolivre.com.br/MLB-914121014-cabo-adaptador-alimentaco-arduino-bateria-9v-_JM?quantity=1, 2019.
- [60] 2. MERCADO LIVRE, *Módulo Gprs Gsm - Arduino Sim808*. https://produto.mercadolivre.com.br/MLB-721788716-modulo-gprs-gsm-arduino-sim8001-_JM, 2019.
- [61] C. Orlandi, *Firestore: serviços, vantagens, quando utilizar e integrações*, <https://blog.rocketseat.com.br/firebase/>, 2018.

- [62] C. SEGURA, *DISEÑO Y DESARROLLO DE UN APLICATIVO MÓVIL EN LA PLATAFORMA ANDROID STUDIO PARA LA EMPRESA ALLIED ELECTRONICS SERVICE(SERVICIOS ELECTRÓNICOS ALIADOS) PERÚ*. <https://core.ac.uk/download/pdf/84108432.pdf>, 2016.
- [63] W. Wu, *React Native vs Flutter, cross-platform mobile application frameworks*, <https://www.theseus.fi/bitstream/handle/10024/146232/thesis.pdf?sequence=1>, 2018.
- [64] H. Raj, *Fludex a redux based state managment library specialy build only for Flutter*. <https://github.com/hemanthrajv/fludex>, 2017.
- [65] R. Souza, *Automatização do processo de monitoramento da qualidade das águas em cursos d'água*, <https://bibliotecadigital.ipb.pt/handle/10198/1673/>, 2020.
- [66] DFRobot, *SEN0161 PH meter*, https://wiki.dfrobot.com/PH_meter_SKU_SEN0161_, 2020.
- [67] —, *SEN0137 DHT22 Temperature and humidity module*, https://wiki.dfrobot.com/DHT22_Temperature_and_humidity_module_SKU_SEN0137_, 2020.
- [68] —, *DS18B20 Arduino Temperature sensor*, https://wiki.dfrobot.com/Waterproof_DS18B20_Digital_Temperature_Sensor__SKU_DFR0198_, 2020.
- [69] —, *DFR0300 Gravity Analog Electrical Conductivity Sensor Meter V2 K=1*, https://wiki.dfrobot.com/Gravity__Analog_Electrical_Conductivity_Sensor__Meter_V2__K%3D1__SKU_DFR0300_, 2020.
- [70] —, *Gravity arduino turbidity sensor*, https://wiki.dfrobot.com/Turbidity_sensor_SKU_SEN0189_, 2020.
- [71] —, *SEN0237 Gravity Analog Dissolved Oxygen Sensor*, https://wiki.dfrobot.com/Gravity__Analog_Dissolved_Oxygen_Sensor_SKU_SEN0237_, 2020.
- [72] —, *DS1307 Real Time Clock*, https://wiki.dfrobot.com/Real_Time_Clock_Module__DS1307__SKU_DFR0151_, 2020.

- [73] —, *Micro SD(TF) module*, https://wiki.dfrobot.com/MicroSD_card_module_for_Arduino__SKU_DFR0229_, 2020.
- [74] —, *SIM808 GPS GPRS GSM Shield*, https://wiki.dfrobot.com/SIM808_GPS_GPRS_GSM_Shield_SKU__TEL0097_, 2020.
- [75] L. M. Engineers, *Send Receive SMS Call with SIM800L GSM Module Arduino*, <https://lastminuteengineers.com/sim800l-gsm-module-arduino-tutorial/>, 2020.