

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

**ISAQUE VILSON BATISTA DA COSTA**

**DESENVOLVIMENTO DE UM SOFTWARE SUPERVISÓRIO PARA  
MONITORAMENTO DA QUALIDADE DA ÁGUA NA ESCOLA  
SUPERIOR DE TECNOLOGIA DA UEA ATRAVÉS DA AQUISIÇÃO DE  
DADOS DE SENSORES**

MANAUS

2022

**ISAQUE VILSON BATISTA DA COSTA**

**DESENVOLVIMENTO DE UM SOFTWARE SUPERVISÓRIO PARA  
MONITORAMENTO DA QUALIDADE DA ÁGUA NA ESCOLA  
SUPERIOR DE TECNOLOGIA DA UEA ATRAVÉS DA AQUISIÇÃO DE  
DADOS DE SENSORES**

Projeto de pesquisa desenvolvido durante a disciplina de Trabalho de Conclusão de Curso II e apresentada à banca avaliadora do Curso de Engenharia Elétrica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para obtenção do título de Engenheiro Eletricista.

Orientador: Prof. Dr. Fábio de Sousa Cardoso.

Manaus

2022

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

*Reitor:*

**André Luiz Nunes Zogahib**

*Vice-Reitor:*

**Kátia do Nascimento Couceiro**

*Diretora da Escola Superior de Tecnologia:*

**Ingrid Sammyne Gadelha Figueiredo**

*Coordenador do Curso de Engenharia Elétrica:*

**Israel Gondres Torné**

*Banca Avaliadora composta por: Data da defesa: 25/10/2022.*

**Prof. Fábio de Sousa Cardoso (Orientador)**

**Prof. Israel Gondres Torné**

**Prof. André Luiz Printes**

## **CIP – Catalogação na Publicação**

Costa, Isaque.

Desenvolvimento de um software supervisor para monitoramento da qualidade da água na escola superior de tecnologia da UEA através da aquisição de dados de sensores / Isaque Wilson Batista da Costa; [orientado por] Fábio de Sousa Cardoso. – Manaus: 2022.

96 p.: il.

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

1. software supervisor. 2. qualidade da água. 3. universidade. 4. greenmetric. 5.iot. I. Cardoso, Fábio de Sousa.

ISAQUE VILSON BATISTA DA COSTA

**DESENVOLVIMENTO DE UM SOFTWARE SUPERVISÓRIO PARA  
MONITORAMENTO DA QUALIDADE DA ÁGUA NA ESCOLA  
SUPERIOR DE TECNOLOGIA DA UEA ATRAVÉS DA AQUISIÇÃO DE  
DADOS DE SENSORES**

Projeto de pesquisa desenvolvido durante a disciplina de Trabalho de Conclusão de Curso II e apresentada à banca avaliadora do Curso de Engenharia Elétrica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para obtenção do título de Engenheiro Eletricista.

Nota obtida: 10 ( DEZ )

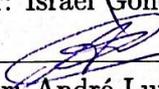
Aprovado em 25/10/2022

Área de concentração: Engenharia Elétrica

BANCA EXAMINADORA

  
Orientador: Fábio de Souza Cardoso, Dr.

  
Avaliador: Israel Gondres Torné, Dr.

  
Avaliador: André Luiz Printes, Me.

MANAUS

2022

## **Dedicatória**

Acima de tudo, agradeço a DEUS por mais esta realização.

Dedico este trabalho a todos os que me ajudaram ao longo desta caminhada.

## AGRADECIMENTO

À Jesus Cristo, que apesar da minha incapacidade de agradá-lo por meios próprios, sempre proveu todas as minhas necessidades para que, dia após dia, eu me mantivesse firme na luta em busca dos meus sonhos.

À minha família, que contribuiu grandemente na construção do meu caráter, da minha educação e do homem que me tornei. Por meio dos conselhos, aprendi que entregar o meu melhor em tudo que eu fizer não é uma opção, e sim um compromisso.

À minha amada esposa, Denise Pardo da Rocha, que há anos vem lutando ao meu lado em busca de dias melhores. Ela é diretamente responsável pela minha fonte de motivação, que consiste em me tornar um homem melhor, dia após dia, a fim de fazê-la feliz e sentir que é amada.

As boas amizades desenvolvidas ao longo da jornada acadêmica no curso de Engenharia Elétrica, em especial aos amigos Eduardo Sale e Tiago Ramos, que certamente tiveram impacto na minha formação acadêmica.

Ao meu orientador, Dr. Fábio de Sousa Cardoso, pelas orientações pessoais e técnicas fornecidas e por todas as oportunidades concedidas, frutos deste e de outros trabalhos desenvolvidos.

A todos que participaram, direta ou indiretamente do desenvolvimento deste trabalho de pesquisa, enriquecendo o meu processo de aprendizado.

Às pessoas com quem convivi a longo desses anos de curso, em especial ao meu glorioso, inderrubável, indormecível, inbugável e calvo time de Software do Laboratório de Sistemas Embarcados (LSE): Alison Almeida, Evaldo Cardoso, Vitor Norton, Lohan Victor, Gabriella Rabelo, Lucas Castro, Fábio Wendel e Maicon Wellington, que sempre estiveram próximos e unidos frentes aos desafios inerentes ao universo da programação e me ajudaram a se tornar um líder melhor. A todos do laboratório LSE, obrigado!

Aos meus dogs: Pretinha, Black e Tinho, que me ensinaram que sentar junto depois de um dia mau, mesmo sem falar português, é suficiente para tudo ficar melhor.

## RESUMO

A Universidade do Estado do Amazonas tem se destacado cada vez mais por adotar práticas de uso consciente dos recursos naturais dentro da instituição entrando para o ranking do *GreenMetric* de universidades sustentáveis. No entanto, por mais que tenha pontuado em todas as categorias do ranking, no quesito qualidade da água, recebeu nota zero, comprovando a falta de medidas que incentivem a sustentabilidade deste recurso natural. Com isso, o objetivo deste trabalho é o desenvolvimento de um sistema de monitoramento da água. Um software capaz de receber dados de sensores e exibi-los em tempo real aos usuários com o intuito de fazer um acompanhamento da qualidade da água consumida na Universidade do Estado do Amazonas e conseqüentemente elevar o nível da instituição no ranking *GreenMetric*. A implementação é feita através da utilização de um dispositivo apto a captar parâmetros físico-químicos da água. Para a comunicação, foi desenvolvido um serviço que recebe os dados a partir de um protocolo de comunicação e os publica em um tópico SNS da AWS que será posteriormente inserida em uma fila SQS integrada ao software de monitoramento desenvolvido. A utilização desse serviço torna a plataforma de monitoramento compatível com qualquer dispositivo de coleta de dados presente no mercado. Desta forma, foi obtido um sistema supervisorio para monitoramento dos parâmetros da água para acompanhamento de especialistas quanto à qualidade deste recurso natural.

**Palavras chave:** software supervisorio; qualidade da água; universidade; *GreenMetric*; iot.

## ABSTRACT

The University of the State of Amazonas has stood out for adopting practices of conscious use of natural resources. This fact helped the institution enter the GreenMetric ranking of sustainable universities scoring in all categories but water quality, proving the lack of actions that encourage the sustainability of this natural resource. Therefore, the main goal of this study is to develop a water monitoring system. A software that can receive data from sensors and display them in real-time for monitoring the water consumed at the University of the State of Amazonas to help raise the institution's scale in the GreenMetric ranking. For that, a device capable of capturing the physical-chemical parameters of the water helped the implementation. For data transmission, the development of a service that receives the data from this device through a communication protocol and publishes them in an SNS topic that passes to an SQS queue of AWS integrated into the monitoring software developed. Using this service makes the monitoring platform compatible with any data collection device in the market. The final result is a supervisory system for monitoring water parameters for expert observation of this natural resource quality.

**Keywords:** monitoring platform; water quality; university; *GreenMetric*; iot.

## Lista de Figuras

1	Desenvolvimento Econômico x Preservação dos Recursos Naturais. . . . .	11
2	Gráfico do consumo da água no Brasil. . . . .	12
3	Número de Universidades e de Países no UI GreenMetric World University. . . . .	13
4	Comparação entre sinal analógico e sinal digital. . . . .	13
5	Áreas de atuação da Internet das Coisas. . . . .	14
6	Aplicação dos protocolos de comunicação à arquitetura <i>Internet of Things</i> (IoT). . . . .	15
7	Funcionamento do <i>Message Queuing Telemetry Transport</i> (MQTT). . . . .	16
8	Fluxo de comunicação SNS e SQS da AWS. . . . .	17
9	Comparação entre arquitetura monolítica e microsserviços. . . . .	18
10	Princípios de SOLID. . . . .	19
11	Exemplo do <i>Single Responsibility Principle</i> na prática . . . . .	20
12	Exemplo do <i>Open Close Principle</i> na prática . . . . .	21
13	Exemplo do <i>Liskov Substitution Principle</i> na prática . . . . .	22
14	Exemplo do <i>Interface Segregation Principle</i> na prática . . . . .	23
15	Exemplo da <i>interface</i> com métodos da classe <i>Atendentes</i> . . . . .	24
16	Exemplo do <i>Dependency Inversion Principle</i> na prática . . . . .	25
17	Camadas da <i>Clean Architecture</i> . . . . .	26
18	Banco de dados PostgreSQL . . . . .	27
19	Banco de dados MongoDB . . . . .	27
20	Arquitetura <i>event loop</i> do NodeJS . . . . .	28
21	Comparação entre o modelo padrão tradicional e o <i>Single Page Application</i> . . . . .	29
22	Grandes empresas que implementam suas ferramentas com React. . . . .	29
23	Modelo em Cascata. . . . .	30
24	Diagrama de caso de uso. . . . .	40
25	Identidade visual da plataforma <i>Water Inspection</i> . . . . .	52
26	Prototipação da plataforma <i>Water Inspection</i> . . . . .	53
27	Arquitetura geral do sistema. . . . .	54
28	Arquitetura em nuvem do sistema. . . . .	55

29	Diagrama de relacionamentos da tabela de usuário . . . . .	56
30	Diagrama de relacionamentos da tabela de dispositivo . . . . .	57
31	Diagrama de modelo do documento de amostragem . . . . .	58
32	IDE utilizada no projeto . . . . .	59
33	Aplicação do ESLint para correção de código . . . . .	60
34	Configuração de padrão de código ESLint . . . . .	60
35	Utilização do Dbeaver no projeto . . . . .	61
36	Teste de rotas com Postman . . . . .	62
37	Arquivos do backend em <i>TypeScript</i> . . . . .	63
38	Arquitetura de serviço do <i>Backend</i> . . . . .	64
39	Estrutura de pasta desenvolvida no <i>Backend</i> . . . . .	65
40	Estrutura de pasta desenvolvida no <i>Frontend</i> . . . . .	67
41	Recebimento dos dados de parâmetros físico-químicos da água através de um dispositivo. . . . .	68
42	Mensagens com amostras de dados na fila <i>Amazon Simple Queue Service</i> (SQS) da <i>Amazon Web Services</i> (AWS) . . . . .	69
43	Página de login da plataforma <i>Water Inspection</i> . . . . .	70
44	Página de mapa com os postos de monitoramento em tempo real . . . . .	71
45	Página de listagem de posto de monitoramento . . . . .	72
46	Página de detalhes de um posto de monitoramento . . . . .	73
47	Página de eventos e alarmes de um posto de monitoramento . . . . .	74
48	Página de geração de relatório . . . . .	75
49	Planilha gerado pela plataforma . . . . .	75
50	Página de edição de informações de um PDM . . . . .	76
51	Página de configuração do modo de operação de um posto de monitoramento . . . . .	77
52	Página de configuração de alertas de um posto de monitoramento . . . . .	78
53	Página de listagem de usuários cadastrados na plataforma . . . . .	79
54	Página de cadastro de novo usuário na plataforma. . . . .	80
55	Página de edição de usuário na plataforma. . . . .	80
56	Página de detalhes de um posto de monitoramento para um usuário com perfil de analista. . . . .	81

## Lista de Tabelas

1	Lista de requisitos funcionais . . . . .	32
2	(Requisitos Funcionais (RF)) Exibir localização dos PDM's . . . . .	33
3	(RF) Exibir informações de análise da água em Dashboard . . . . .	33
4	(RF) Exibir alertas de parâmetros fora dos limites estabelecidos e violação de um PDM . . . . .	34
5	(RF) Criar conta de usuário . . . . .	34
6	(RF) Alterar dados de usuário . . . . .	35
7	(RF) Excluir conta de usuário . . . . .	35
8	(RF) Realizar autenticação . . . . .	35
9	(RF) Recuperar acesso . . . . .	36
10	(RF) Disponibilizar dados recebidos . . . . .	36
11	(RF) Receber eventos de dispositivos . . . . .	36
12	(RF) Gerar relatórios . . . . .	37
13	(RF) Configurar posto de monitoramento . . . . .	37
14	(RF) Solicitar Leitura Instantânea . . . . .	37
15	(RF) Exibir status de PDM's instaladas . . . . .	38
16	(RF) Fechar Evento de Anomalia de parâmetros . . . . .	38
17	(RF) Realizar Logout . . . . .	38
18	Requisitos Não-Funcionais . . . . .	39
19	(Caso de Uso) Manter analistas . . . . .	41
20	(Caso de Uso) Fazer Login . . . . .	42
21	(Caso de Uso) Consultar Postos de Monitoramento . . . . .	43
22	(Caso de Uso) Visualizar histórico dos parâmetros físico-químicos da água . . . . .	44
23	(Caso de Uso) Gerar relatório . . . . .	45
24	(Caso de Uso) Visualizar eventos e alarmes . . . . .	46
25	(Caso de Uso) Obter dados da amostragem . . . . .	47
26	(Caso de Uso) Manter Posto de Monitoramento . . . . .	48
27	(Caso de Uso) Configurar eventos de alertas . . . . .	49
28	(Caso de Uso) Configurar modo de operação . . . . .	50

## Lista de Siglas e Abrevaturas

**UEA** Universidade do Estado do Amazonas

**EST** Escola Superior de Tecnologia

**IoT** *Internet of Things*

**HTTP** *Hypertext Transfer Protocol*

**CoAP** *Constrained Application Protocol*

**MQTT** *Message Queuing Telemetry Transport*

**M2M** Máquina-para-Máquina

**TCP** Protocolo de Controle de Transmissão

**IP** Protocolo de Internet

**AWS** *Amazon Web Services*

**SNS** *Amazon Simple Notification Service*

**SQS** *Amazon Simple Queue Service*

**A2A** Aplicação para Aplicação

**A2P** Aplicação para Pessoa

**DTOs** *Data Transfer Objects*

**SPA** *Single Page Application*

**MPA** *Multiple Page Application*

**RF** Requisitos Funcionais

**RNF** Requisitos Não Funcionais

**ESS** Essencial

**IMP** Importante

**DES** Desejável

**PDM** Posto de Monitoramento

**UML** *Unified Modeling Language*

**UI** *User Interface*

**UX** *User Experience*

**API** *Application Programming Interface*

**IDE** *Integrated Development Environment*

**LSE** Laboratório de Sistemas Embarcados

## SUMÁRIO

INTRODUÇÃO . . . . .	8
1 <b>FORMULAÇÃO DO PROBLEMA</b> . . . . .	9
2 <b>HIPÓTESE</b> . . . . .	9
3 <b>OBJETIVO</b> . . . . .	9
4 <b>JUSTIFICATIVA</b> . . . . .	9
5 <b>REFERENCIAL TEÓRICO</b> . . . . .	10
5.1    Preservação de recursos naturais . . . . .	10
5.2    A importância da Água . . . . .	11
5.3 <i>UI GreenMetric World University Ranking</i> . . . . .	12
5.4    Sensores . . . . .	13
5.5 <i>Internet of things</i> - IoT . . . . .	14
5.6    Protocolos da transmissão de mensagem IoT . . . . .	14
5.7    Protocolo MQTT . . . . .	15
5.8    AWS . . . . .	16
5.8.1    SNS . . . . .	16
5.8.2    SQS . . . . .	17
5.9    Sistemas de supervisão . . . . .	17
5.10    Arquitetura de microsserviço . . . . .	18
5.11    Princípios SOLID . . . . .	19
5.11.1    S - <i>Single Responsibility Pattern</i> (Princípio da responsabilidade única) . . . . .	20
5.11.2    O - <i>Open Close Principle</i> (Princípio Aberto-Fechado) . . . . .	20
5.11.3    L - <i>Liskov Substitution Principle</i> (Princípio da substituição de Liskov) . . . . .	21
5.11.4    I - <i>Interface Segregation Principle</i> (Princípio da Segregação da Interface) . . . . .	22
5.11.5    D - <i>Dependency Inversion Principle</i> (Princípio da inversão da dependência) . . . . .	24

5.12	<i>Clean Architecture</i> . . . . .	25
5.13	Banco de Dados - PostgreSQL . . . . .	27
5.14	Banco de Dados - MongoDB . . . . .	27
5.15	NodeJS . . . . .	27
5.16	React . . . . .	28
<b>6</b>	<b>METODOLOGIA</b> . . . . .	<b>29</b>
6.1	Análise de Requisitos . . . . .	30
6.1.1	Lista de Requisitos . . . . .	32
6.1.2	Descrição dos Requisitos. . . . .	33
6.2	Caso de Uso . . . . .	40
6.3	Caso de Uso detalhado . . . . .	41
<b>7</b>	<b>DESENVOLVIMENTO</b> . . . . .	<b>51</b>
7.1	<i>Water Inspection</i> : A plataforma para avaliação da qualidade da água . . . . .	51
7.2	Prototipação (UI/UX) . . . . .	52
7.3	Arquitetura geral do sistema . . . . .	53
7.4	Arquitetura em nuvem . . . . .	54
7.5	Modelagem do banco de dados . . . . .	55
7.5.1	Modelagem do banco relacional - PostgreSQL . . . . .	56
7.5.2	Modelagem do banco não-relacional - MongoDB . . . . .	57
7.6	Ferramentas e tecnologias utilizadas . . . . .	58
7.6.1	VS Code - Visual Studio Code . . . . .	58
7.6.2	ESLint . . . . .	59
7.6.3	Dbeaver . . . . .	61
7.6.4	Postman . . . . .	61
7.6.5	<i>TypeScript</i> . . . . .	62
7.7	Estrutura e padrão de projeto utilizado . . . . .	63
7.7.1	<i>Backend</i> . . . . .	63
7.7.2	<i>Frontend</i> . . . . .	65
<b>8</b>	<b>RESULTADOS</b> . . . . .	<b>68</b>
8.1	Validação da plataforma desenvolvida com um dispositivo para coleta de dados de parâmetros físico-químicos da água. . . . .	68
8.2	Página de autenticação . . . . .	69
8.3	Página de mapa com os postos de monitoramento . . . . .	70
8.4	Página de listagem de postos de monitoramento . . . . .	71
8.5	Página de detalhes de um posto de monitoramento . . . . .	72
8.6	Página de eventos e alarmes do posto de monitoramento . . . . .	73
8.7	Página de geração de relatório . . . . .	74

8.8	Página de configuração de um posto de monitoramento . . . . .	76
8.9	Página de listagem de usuários do sistema . . . . .	78
8.10	Página de cadastro/edição de um usuário . . . . .	79
8.11	Diferenças na página entre um usuário analista e um administrador . . . . .	81
<b>9</b>	<b>CONCLUSÃO . . . . .</b>	<b>82</b>
9.1	Sugestão de trabalhos futuros . . . . .	82
	<b>REFERÊNCIAS . . . . .</b>	<b>83</b>
<b>A</b>	<b>APÊNDICE - DIAGRAMA DE SEQUÊNCIA - UML . . . . .</b>	<b>86</b>
A.1	Diagrama de sequência para <i>login</i> do Administrador . . . . .	86
A.2	Diagrama de sequência para obter parâmetros físico-químico da água . . . . .	86
A.3	Diagrama de Sequência para editar dados do usuário . . . . .	87
A.4	Diagrama de Sequência para deletar dados do usuário . . . . .	87

## INTRODUÇÃO

A preocupação com a preservação dos recursos naturais para as gerações futuras tem se intensificado cada vez mais nos últimos anos. Neste âmbito, as universidades possuem a responsabilidade de dar bons exemplos ao adotar medidas sustentáveis, já que possuem um papel importante em suas comunidades locais, contribuindo para a formação acadêmica dos seus cidadãos (L, 2021). Por essa razão, há valor em desenvolver soluções e práticas que fomentem a sustentabilidade no ambiente universitário.

Levando em consideração este cenário, foi criado em 2010, o *UI GreenMetric World University Ranking*<sup>1</sup>, uma iniciativa da *Universitas Indonesia* (UI) para classificar as universidades ao redor do mundo, tendo como métrica suas medidas tomadas para a preservação dos recursos naturais. O objetivo, além de realizar este ranking anual, é encorajar práticas sustentáveis nas universidades (GREENMETRIC, 2021).

A Universidade do Estado do Amazonas (UEA) participou pela primeira vez do *UI GreenMetric World University Ranking* em 2020 e, apesar de ter pontuado nas outras categorias, obteve nota zero no quesito “Gestão da Água” (GREENMETRIC, 2021). Isso mostra a falta de medidas sustentáveis para a preservação dos recursos hídricos na instituição.

Tendo em vista este contexto, a implementação de uma solução tecnológica que possa medir e exibir os parâmetros físico-químicos da água no sistema de abastecimento proporcionará auxílio na identificação de medidas que possam ser tomadas para manter a boa qualidade e evitar desperdícios da água. Além disso, a solução possibilitará uma nota melhor no Ranking e uma maior visibilidade da universidade.

Este projeto de pesquisa tem como objetivo final o desenvolvimento de um software supervisor para o monitoramento da qualidade da água na Escola Superior de Tecnologia (EST), através de uma implementação que envolve os principais protocolos de comunicação em rede, adquirindo dados de sensores físicos.

---

<sup>1</sup>Para mais detalhes, acessar: <https://greenmetric.ui.ac.id/>

## 1 FORMULAÇÃO DO PROBLEMA

A Universidade do Estado do Amazonas ocupou a 855ª posição dentre as 912 instituições inscritas no *UI GreenMetric Ranking* em 2020. Nas categorias em que foi avaliada, obteve nota zero no quesito “Gestão da Água”. Foi possível notar que um dos fatores que contribuíram para a pontuação zero na categoria foi a falta de ferramentas implementadas na instituição que facilitem o monitoramento do sistema de abastecimento hídrico. Portanto, observou-se a necessidade de construir recursos que auxiliem no desenvolvimento sustentável, mais precisamente, sendo uma ferramenta tecnológica para o monitoramento da qualidade da água na Escola Superior de Tecnologia.

## 2 HIPÓTESE

É possível desenvolver um software supervisor para o monitoramento da qualidade da água na Escola Superior de Tecnologia da UEA o qual faz a aquisição de dados de sensores que captam os parâmetros físico-químicos da água. Isso pode ser feito através da implementação de um serviço que realize a comunicação entre dispositivos físicos e o software supervisor para monitoramento da qualidade da água. Os dados adquiridos serão armazenados como amostras em uma base de dados MongoDB e todo o monitoramento será feito através do sistema computacional desenvolvido em NodeJS e React.

## 3 OBJETIVO

Desenvolver um software supervisor de monitoramento que faça a comunicação com postos de monitoramento constituídos por dispositivos conectados a sensores de parâmetros físico-químicos da água, para que seja possível manter o acompanhamento das condições de qualidade deste recurso natural. Será desenvolvido um serviço computacional para realizar a aquisição de dados de qualquer dispositivo presente no mercado e armazená-los em uma base de dados que constituirá o sistema de monitoramento desenvolvido em NodeJs e React. Os parâmetros mencionados serão coletados do sistema de abastecimento hídrico da EST.

## 4 JUSTIFICATIVA

A implementação do software supervisor possibilitará o monitoramento da qualidade da água utilizada na Escola Superior de Tecnologia. Por meio deste sistema, será possível realizar uma melhor gestão dos recursos hídricos disponíveis na unidade, viabilizando a análise para o controle de desperdícios e contaminações. Conseqüentemente, também contribuirá para que a Universidade do Estado do Amazonas alcance colocações mais elevadas no *UI GreenMetric World University Ranking*, já que atenderá a uma das

suas categorias de avaliação. E para o desenvolvimento do projeto serão utilizados os conhecimentos adquiridos na formação acadêmica em Engenharia Elétrica, principalmente das disciplinas: Introdução às Ciências do Ambiente, Linguagem de Programação, Sistemas Microprocessados, Microcontroladores, Redes de Comunicações de Dados e Estágio Supervisionado.

## 5 REFERENCIAL TEÓRICO

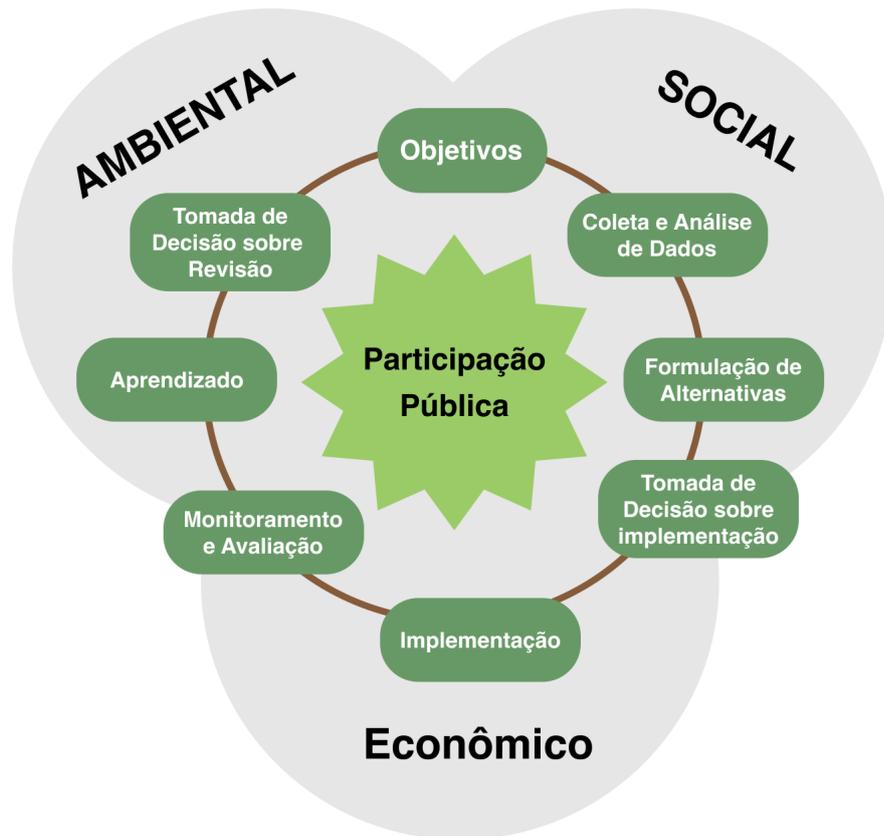
Neste capítulo, serão abordados os aspectos teóricos dos assuntos relacionados ao projeto. Inicialmente, será feita uma contextualização sobre a preservação dos recursos naturais e a importância da preservação da água. Em seguida, serão abordados conceitos introdutórios sobre o *UI Greenmetric World University Ranking*, a internet das coisas, sistemas de supervisão e as tecnologias para o desenvolvimento deste trabalho.

### 5.1 Preservação de recursos naturais

Profissionais da área de Conservação de Recursos Naturais enfrentam desafios sem precedentes que vêm aumentando gradativamente ao longo dos últimos anos. Desafios estes que são decorrentes das mudanças no uso dos solos, na biodiversidade da terra, no clima e dentre outras. Essas mudanças acabam introduzindo uma lista de incertezas que provocam o surgimento de novos obstáculos quanto à tomada de decisão de como fazer um bom uso dos recursos naturais para melhor preservá-los. Por mais que as incertezas não sejam algo novo no gerenciamento e consequentemente preservação dos recursos naturais, estas limitações na capacidade de prever quais são os verdadeiros efeitos das mudanças climáticas e outros provocadores de mudanças nos ecossistemas têm reforçado a necessidade de ferramentas que ajudem com tais incertezas (ROWLAND E.R; CROSS, 2014).

A figura 1 apresenta um diagrama de ações para a implantação de medidas sustentáveis e diminuir incertezas dos impactos causados pelas mudanças no ecossistema.

Figura 1 – Desenvolvimento Econômico x Preservação dos Recursos Naturais.

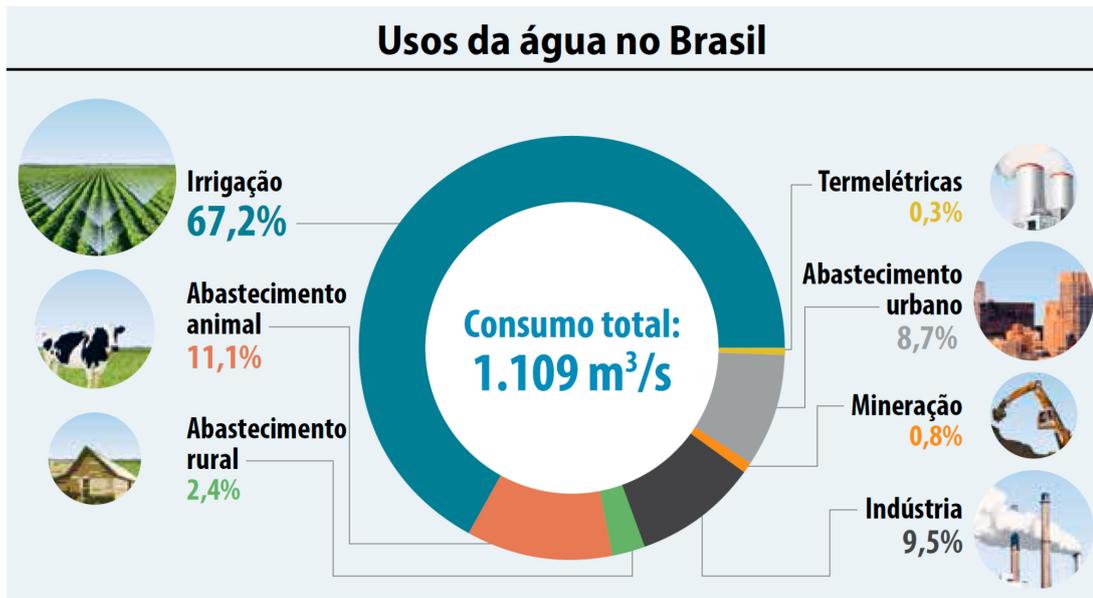


Fonte: (AMBIENTE, 2022)

## 5.2 A importância da Água

A água é muito importante para manter um ambiente produtivo para todos os organismos vivos, é necessária para manter as atividades humanas vitais, tais como nutrição, respiração, circulação, excreção e reprodução. Além disso, a água também pode ser um ecossistema, sendo uma das substâncias básicas para a formação de vida. Sua importância também se estende à sociedade, já que é essencial não só para um bom abastecimento de alimentos como também para a saúde pública, ajudando a manter os padrões de qualidade de vida mínimos de uma população. Por conta disso, à medida que as economias dos países e suas populações crescem, a demanda por este recurso natural também aumenta. No entanto, é possível notar um “estresse hídrico” em muitos países, sua falta reduz severamente a biodiversidade, assim como os ecossistemas terrestres e aquáticos, aumentando os impactos ambientais e a falta nos suprimentos de comida. Como resultado, vem se constatando ainda mais a necessidade urgente de conservar a água (KILIÇ, 2020). A figura 2 traz como este recurso natural é consumido no contexto brasileiro.

Figura 2 – Gráfico do consumo da água no Brasil.



Fonte: (SENADO, 2022)

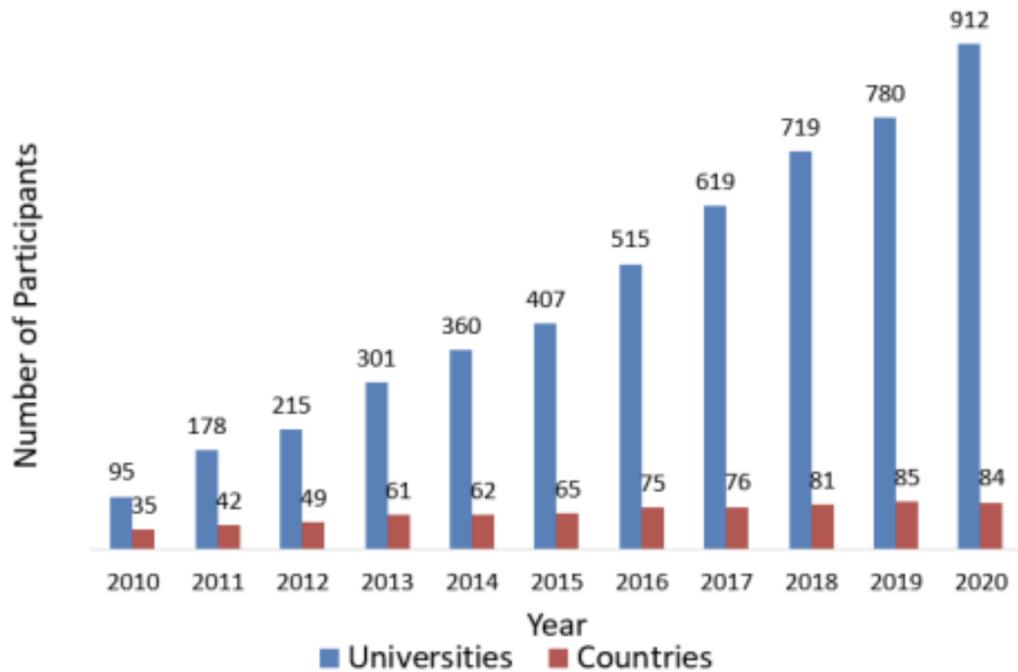
### 5.3 *UI GreenMetric World University Ranking*

O *UI GreenMetric World University Ranking* foi criado pela Universitas Indonesia e teve seu início em 2010. O objetivo é disponibilizar uma pesquisa online que apresente as medidas práticas adotadas pelas Universidades ao redor do mundo que contribuem para a sustentabilidade (GREENMETRIC, 2021). Abrange um total de 36 indicadores de sustentabilidade que se encaixam em seis categorias:

- (a) Condições e Infraestrutura;
- (b) Energia e Mudanças Climáticas;
- (c) Desperdício;
- (d) Água;
- (e) Transporte;
- (f) Educação e Pesquisa;

Desde sua primeira edição em 2010, o número de universidades e países participantes no *UI GreenMetric World University Ranking* vem aumentando expressivamente, o que mostra o reconhecimento mundial que o *Ranking* tem ganhado (GREENMETRIC, 2021). A figura 3 mostra a quantidade de países (em vermelho) e universidades (em azul) que se inscreveram nas edições de 2010 à 2020.

Figura 3 – Número de Universidades e de Países no UI GreenMetric World University.

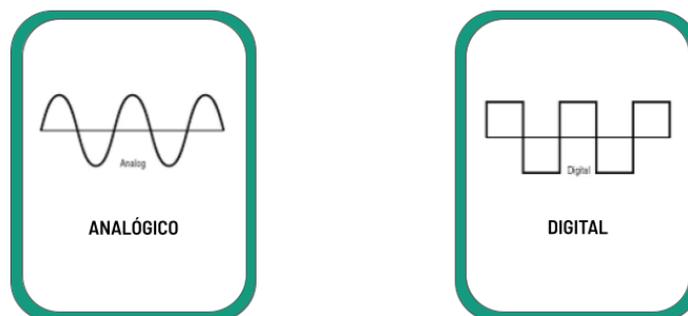


Fonte: (GREENMETRIC, 2021)

#### 5.4 Sensores

São dispositivos sensíveis a alguma forma de energia, sendo luminosa, térmica ou cinética. Os sensores podem ser divididos em analógicos e digitais, a diferença entre os estes dois tipos de sinais pode ser visualizado na figura 4, sendo que os analógicos assumem qualquer valor em um espaço de tempo, desde que estejam dentro de sua faixa de operação, já os digitais assumem apenas dois valores no seu sinal de saída ao longo do tempo, interpretados como zero ou um: desligado - *OFF* ou ligado - *ON* (Não existe grandezas físicas que assumam esses valores, mas eles são assim mostrados ao sistema de controle) (ALBUQUERQUE;, 2020).

Figura 4 – Comparação entre sinal analógico e sinal digital.



Fonte: (Elaborado pelo autor, 2022)

## 5.5 *Internet of things* - IoT

O termo IoT é a interconexão de dispositivos físicos que se integram em eletrônica, software e com sensores para uma troca contínua de dados. A IoT transforma objetos físicos em dispositivos inteligentes para que possam coletar, comunicar, monitorar e interpretar informações do seu ambiente. Os dispositivos se comunicam por meio de protocolos de comunicação que permitem o seu monitoramento e controle por meio de sistemas baseados em nuvem (BRIEF, 2019). A figura 5 ilustra as diferentes áreas de aplicação da internet das coisas.

Figura 5 – Áreas de atuação da Internet das Coisas.



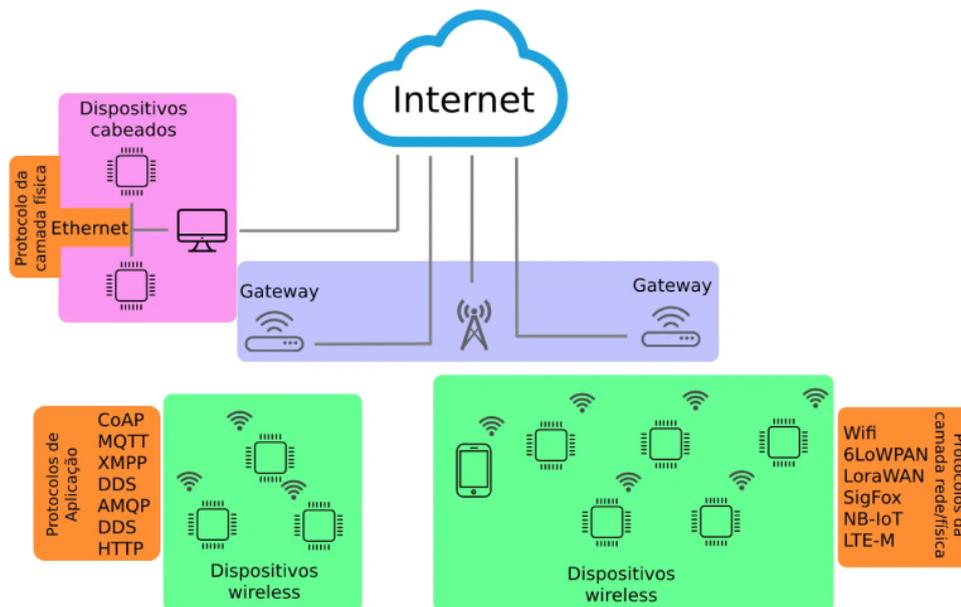
Fonte: (MELO, 2022)

## 5.6 Protocolos da transmissão de mensagem IoT

Sistemas embarcados, permitem a utilização de sensores, atuadores e transmissão de dados em hardwares com tamanho restrito. No geral, isto implica em processamento, memória primária e armazenamento limitado. Por outro lado, estes dispositivos apresentam baixo consumo energético.

No entanto, projetos de sensoriamento e transmissão de dados requerem que os desenvolvedores definam a arquitetura do hardware e os protocolos de comunicação que serão utilizados. O projeto de um dispositivo leva em consideração, custo, capacidade de processamento, memória e armazenamento, rede de comunicação e fonte de energia. Já a escolha de protocolos de comunicação dependerá do desempenho da arquitetura bem como das características da aplicação. Atuando na camada de aplicação, diversos protocolos de comunicação específicos para IoT têm sido propostos na literatura, como exposto na figura 6. Os protocolos da camada de aplicação IoT permitem uma melhor interação entre os dados sensoreados pelos dispositivos à aplicação alvo. Esses protocolos vão de versões similares ao *Hypertext Transfer Protocol* (HTTP) modificadas para o mundo de objetos, como o *Constrained Application Protocol* (CoAP), à protocolos de distribuição de mensagens em modo *multicast*, como o MQTT. O grande problema é que não existe uma definição muito precisa quando se deve utilizar determinado protocolo, nem quais os impactos provocados por essa escolha (FERNANDES, 2020).

Figura 6 – Aplicação dos protocolos de comunicação à arquitetura IoT.



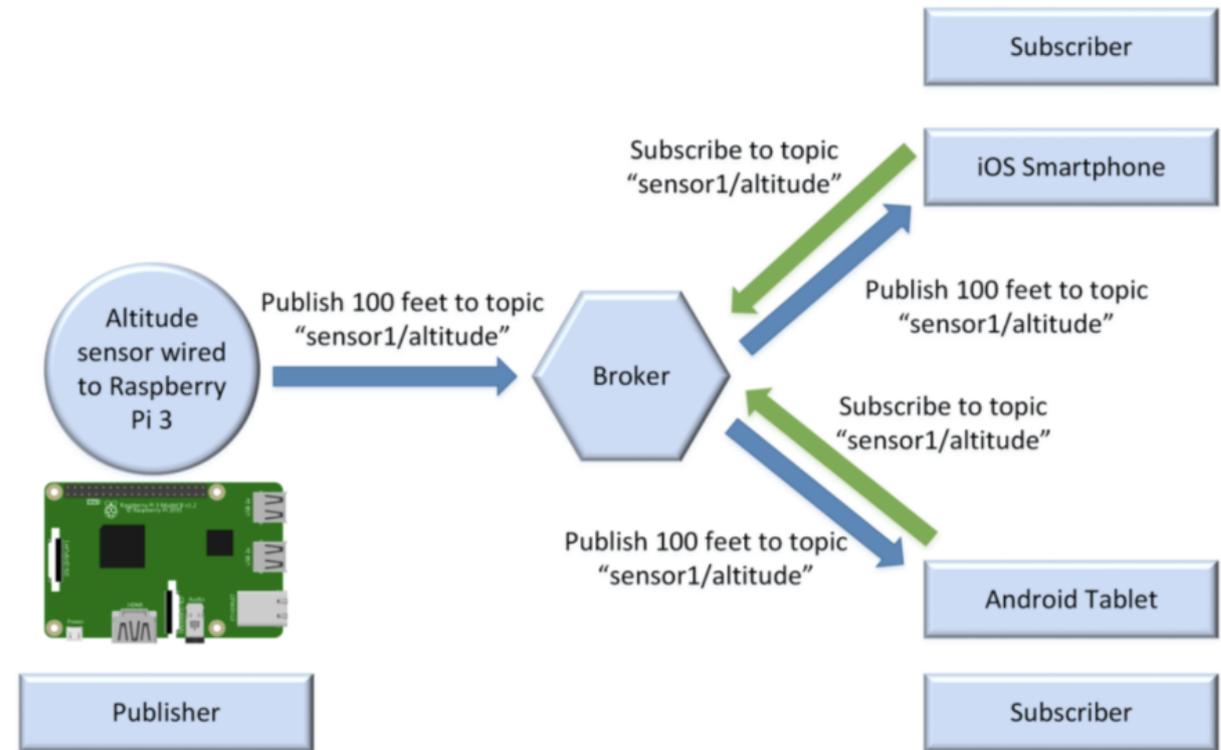
Fonte: (PALARMINI, 2020)

## 5.7 Protocolo MQTT

O MQTT é um protocolo de Máquina-para-Máquina (M2M) e IoT. A figura 7 apresenta seu funcionamento que ocorre com um *broker* em um sistema de inscrição e publicação, e a comunicação é feita em cima do protocolo Protocolo de Controle de Transmissão (TCP) e Protocolo de Internet (IP). O MQTT é mais leve que outros protocolos, como o HTTP. Com isso, é uma opção mais interessante quando se é necessário receber dados próximo ao tempo real no modelo de inscrição e publicação. É um protocolo muito

popular em IoT e tem se tornado cada vez mais popular nas aplicações web e mobile que requerem uma troca de mensagens estável e distribuição eficientemente (HILLAR, 2017).

Figura 7 – Funcionamento do MQTT.



Fonte: (HILLAR, 2017)

## 5.8 AWS

A AWS é a plataforma de computação em nuvem mais adotada e mais abrangente do mundo, oferecendo mais de 200 serviços completos de data centers em todo o mundo. Existem milhões de clientes, incluindo as *startups* de crescimento mais rápido, grandes empresas e os maiores órgãos governamentais. Esta plataforma de computação em nuvem promove a redução de custos, a agilidade na performance e a aplicação de inovações mais rapidamente (AWS, 2022c). Os tópicos 5.8.1 e 5.8.2 fazem a descrição de dois serviços da AWS.

### 5.8.1 SNS

O *Amazon Simple Notification Service* (SNS) é um serviço de mensagens totalmente gerenciado para a comunicação de Aplicação para Aplicação (A2A) e de Aplicação para Pessoa (A2P). A funcionalidade *publisher and subscriber* de A2A fornece tópicos para sistemas de mensagens de alta taxa de transferência para sistemas distribuídos, micros-serviços e aplicações sem servidor orientadas por eventos. Usando tópicos do *Amazon*

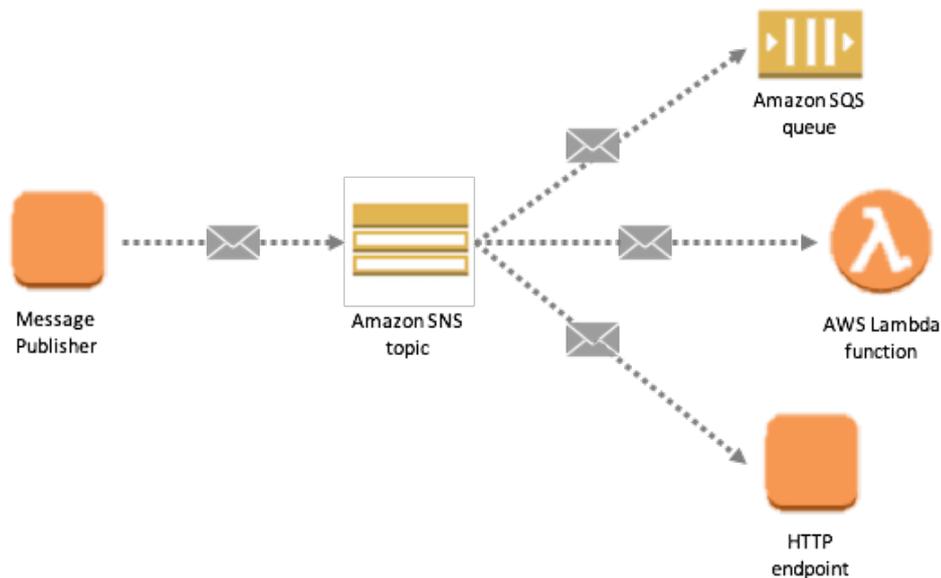
SNS, seus sistemas editores podem repassar mensagens para um grande número de sistemas de assinantes, incluindo filas do *Amazon SQS*, funções do *AWS Lambda* e *endpoints* HTTP e o *Amazon Kinesis Data Firehose* para processamento paralelo. A funcionalidade A2P permite enviar mensagens para usuários em grande escala por SMS, *push notification* de dispositivos móveis e e-mail (AWS, 2022b).

### 5.8.2 SQS

O SQS é um serviço de filas de mensagens gerenciado que permite o desacoplamento e a escalabilidade de microsserviços, sistemas distribuídos e aplicações sem servidor. O SQS elimina a complexidade e a sobrecarga associadas ao gerenciamento e à operação de mensagens, além de permitir que os desenvolvedores se dediquem a criar customização da comunicação. O SQS pode ser usado para enviar, armazenar e receber mensagens entre componentes de software em qualquer volume, sem perder mensagens ou precisar que outros serviços estejam disponíveis (AWS, 2022a).

A figura 8 apresenta o fluxo das mensagens dentre estes dois serviços.

Figura 8 – Fluxo de comunicação SNS e SQS da AWS.



Fonte: (AWS, 2022d)

## 5.9 Sistemas de supervisão

Os sistemas de supervisão eram, inicialmente, formados por painéis para onde a informação era levada por uma grande quantidade de cabos e exibida por um conjunto de lâmpadas. No entanto, estes equipamentos não apresentavam boa confiabilidade e frequentemente demonstravam problemas na comunicação (CARRARO, 2017).

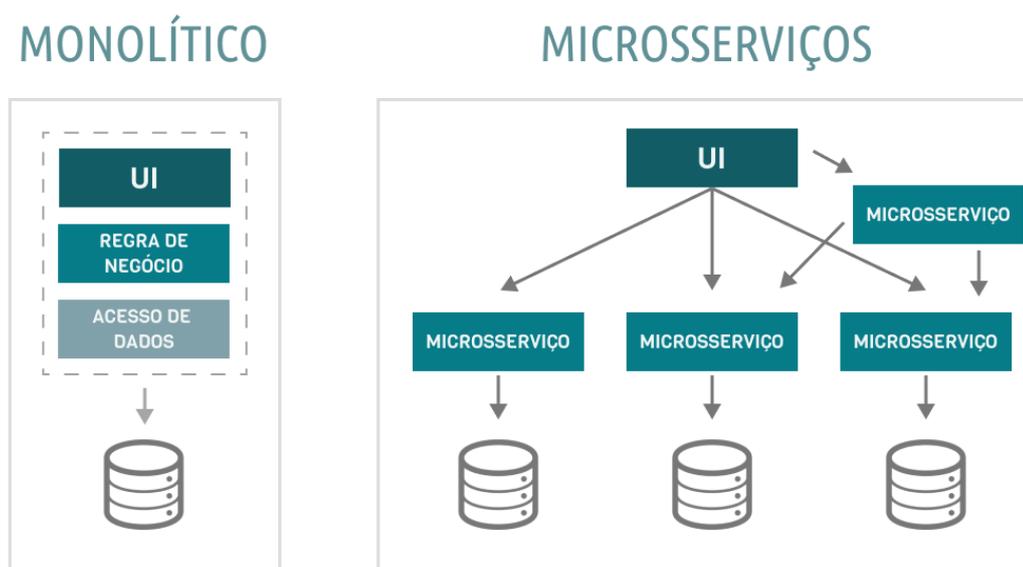
Atualmente, estes sistemas estão presentes em diversas aplicações industriais e utilizam técnicas computacionais que permitem a correção de problemas. O software supervisor reúne um conjunto de programas gerados e configurados no software básico de supervisão. Utiliza o controle e supervisão através de telas gráficas que facilitam a visualização, aquisição e tratamento dos dados do processo e gerência relatório e alarmes. A aplicação de supervisão, a rede industrial, o protocolo de comunicação constitui o sistema de supervisão (CARRARO, 2017).

### 5.10 Arquitetura de microsserviço

O modelo arquitetural de microsserviços propõe dividir o sistema computacional em serviços menores e mais leves, estes serviços são construídos com o propósito de desempenhar de forma coesa uma função na aplicação. A arquitetura de microsserviço é, na verdade, uma evolução do estilo arquitetural tradicional orientado a serviço. Dentre os benefícios deste modelo arquitetural estão incluídos: aumento na agilidade, produtividade dos desenvolvedores, resiliência, confiabilidade, fácil manutenção, descentralização de vulnerabilidades, além disso, o *deploy* da aplicação também se torna mais fácil (EVANS, 2021).

Conforme representado na figura 9, a segregação da aplicação por funcionalidade é a diferença entre a arquitetura de microsserviço e o processo de desenvolvimento monolítico tradicional. Cada serviço pode ser criado e implantado de forma independente, ou seja, cada serviço específico pode funcionar ou falhar sem afetar outros (REDHAT, 2021).

Figura 9 – Comparação entre arquitetura monolítica e microsserviços.

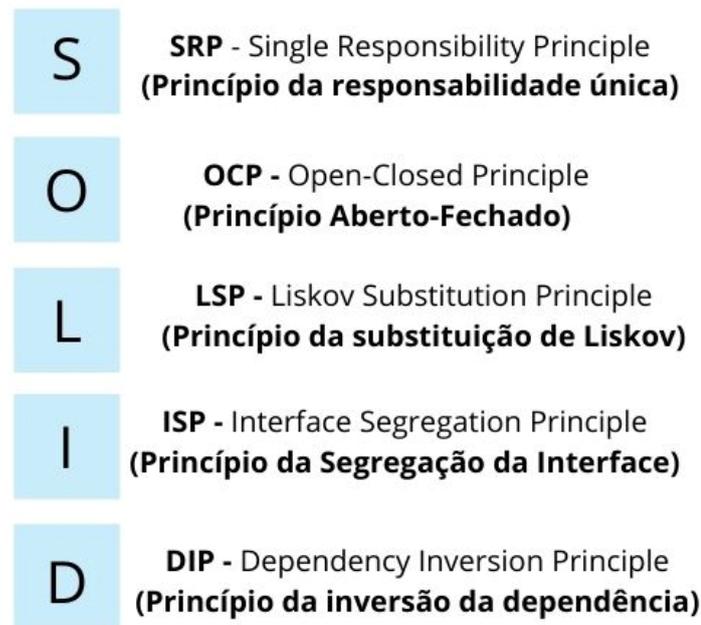


Fonte: Adaptado (REDHAT, 2021)

## 5.11 Princípios SOLID

O SOLID é o conjunto de boas práticas descritas em princípios. Têm como principal objetivo facilitar a manutenção e compreensão de código. Criado por Michael Feathers que identificou a possibilidades de unificar estes princípios na sigla SOLID (MARTIN, 2019). A figura 10 ilustra o acrônimo e seu significado.

Figura 10 – Princípios de SOLID.



Fonte: (ARAUJO, 2021)

Os princípios de SOLID atribuem ao desenvolvimento do projeto, muitos benefícios. É possível citar três deles: a segurança, a facilidade de manutenção e a qualidade do código se tornar reutilizável. O aumento da segurança na aplicação se dá devido à divisão de responsabilidades, não ocorrendo erros devido a mudanças em algum método ou classe. Já a facilidade de manutenção ocorre, pois cada parte do projeto está no seu devido lugar, devido a divisão de responsabilidades, facilitando o entendimento mesmo de desenvolvedores que não tenham participado do desenvolvimento do projeto. E por fim, o código é reutilizável, pois aplicando os princípios não é necessário ficar reescrevendo o mesmo código para executar uma determinada tarefa, basta estender de outra classe que tenha a função necessária.

Os princípios de SOLID serão melhor descritos nas seções a seguir.

### 5.11.1 S - *Single Responsibility Pattern* (Princípio da responsabilidade única)

Este princípio afirma que uma classe deve possuir apenas um objetivo, em outras palavras, ela deve possuir apenas uma função ou funções semelhantes (MARTIN, 2019).

É possível ilustrar este princípio com um caso da vida real. Por exemplo, em um show de música uma única pessoa não faz todo o trabalho, mas um conjunto de pessoas tem as responsabilidades compartilhadas do show em prol de um único objetivo, produzir música de qualidade como demonstrado na figura 11. O princípio de responsabilidade única pode ser comparado a este cenário pois seu objetivo é fazer com que uma classe execute funções que são associadas ao propósito daquela classe.

Figura 11 – Exemplo do *Single Responsibility Principle* na prática



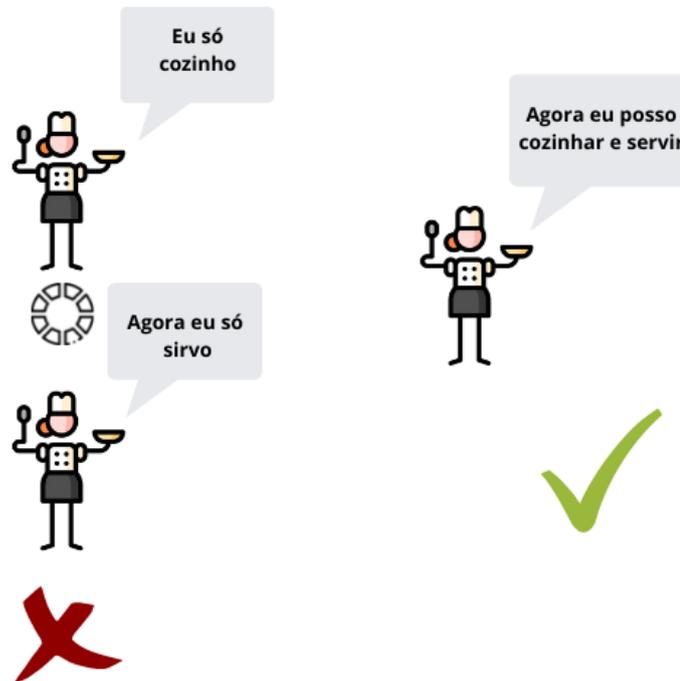
Fonte: (ARAUJO, 2021)

### 5.11.2 O - *Open Close Principle* (Princípio Aberto-Fechado)

Este princípio propõe que as classes de uma aplicação devem ser abertas para extensões e fechadas para modificações, isto é, outras classes podem ter acesso ao que aquela classe possui, no entanto não podem alterá-las, isto se dá pelo fato de não ser aconselhável alterar uma classe, já que outras classes podem estar utilizando-a (MARTIN, 2019).

Um exemplo prático deste princípio pode ser observado na figura 12, no caso à esquerda, a atendente em um restaurante realiza apenas uma atividade e não pode fazer outras coisas, ou seja, está fechada para extensão, já o caso à direita, ela pode realizar duas funções diferentes, ou seja, está aberta para extensão (ARAUJO, 2021).

Figura 12 – Exemplo do *Open Close Principle* na prática



Fonte: (ARAÚJO, 2021)

### 5.11.3 L - *Liskov Substitution Principle* (Princípio da substituição de Liskov)

Este princípio leva o nome da cientista americana Barbara Liskov que propôs a ideia de que “As classes derivadas devem ser substituíveis por suas classes bases” (MARTIN, 2019).

É possível exemplificar este princípio com a seguinte situação ilustrada na figura 13. Uma classe Pessoa possui atributos como CPF e RG. Caso exista uma classe Aluno, seria possível que todos os atributos da classe Pessoa também possam ser dados a classe Aluno, afinal, Aluno é uma pessoa. No entanto, não é interessante repetir todos os atributos para a classe Aluno, o ideal é que Aluno herde de Pessoa. Neste caso, Aluno poderia herdar os atributos de Pessoa e ainda possuir seus próprios atributos, como nota, presença, matérias matriculadas e etc (ARAÚJO, 2021).

Figura 13 – Exemplo do *Liskov Substitution Principle* na prática

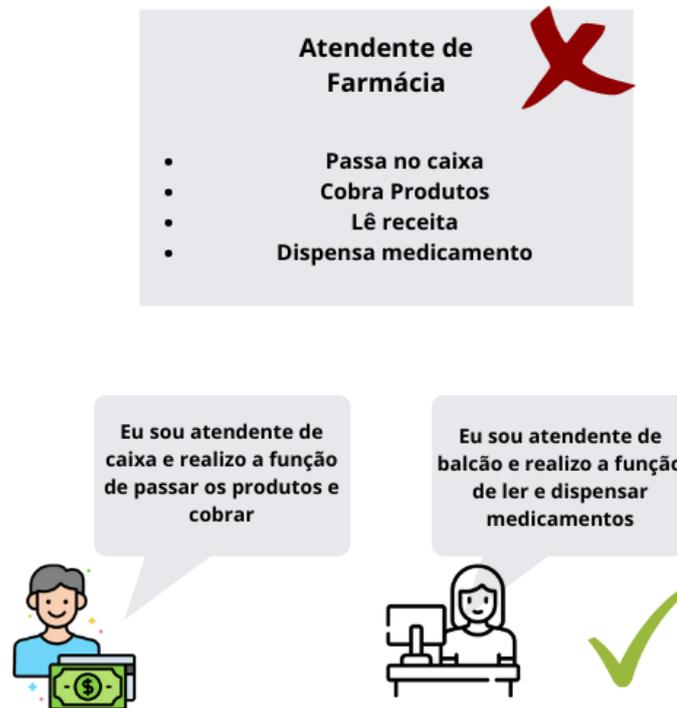


Fonte: (ARAÚJO, 2021)

#### 5.11.4 I - *Interface Segregation Principle* (Princípio da Segregação da Interface)

Este princípio declara que “Classes não devem ser forçadas a depender de métodos que não usam” (MARTIN, 2019). No exemplo dado para o princípio anterior foi demonstrado como o sistema se comporta diante de uma Herança, neste processo a classe filha (que herdou os atributos) é obrigada a implementar os métodos da classe Pai (quem deriva uma classe filha). No entanto, isto é contra os princípios de SOLID, pois não é nada interessante que uma classe implemente métodos que não são úteis (ARAÚJO, 2021).

A figura 14 faz referência a um exemplo capaz de esclarecer este princípio. Tem-se uma classe chamada “AtendenteFarmácia”, mas este cargo é dividido em dois, “atendente de caixa” e “balconista”, os quais executam funções diferentes, logo, haveriam as classes “AtendenteCaixa” e “Balconista”. O balconista tem como principal função ler a receita e entregar os medicamentos ao cliente, já o atendente de caixa, passa os produtos no caixa e recebe o pagamento. Se criar uma *interface*, seriam necessários os métodos exibidos na figura 15 (ARAÚJO, 2021).

Figura 14 – Exemplo do *Interface Segregation Principle* na prática

Fonte: (ARAUJO, 2021)

Ao implementar a *interface* ilustrada na figura 15 na classe "AtendenteCaixa" seria necessário implementar todos os métodos, inclusive os métodos que o atendente de caixa não executa. Para solucionar isso, basta implementar uma *interface* que atenda separadamente as funções de cada atendente (ARAUJO, 2021).

Figura 15 – Exemplo da *interface* com métodos da classe Atendentes

```
public interface IAtendente {  
  
    void leReceita();  
    void dispensaMedicamento();  
    void passaProdutos();  
    void cobraProdutos();  
  
}
```

Fonte: (ARAUJO, 2021)

#### 5.11.5 D - *Dependency Inversion Principle* (Princípio da inversão da dependência)

A proposta deste princípio é “Módulos de alto nível não devem depender de módulos de baixo nível, ambos devem depender da abstração”, e de que “Abstrações não devem depender de detalhes, os detalhes devem depender das abstrações”, em outras palavras, a função da inversão de dependência faz com que os softwares se desassociam dos módulos (MARTIN, 2019).

Um exemplo prático deste princípio pode ser observado na figura 16, o trabalhador que limpa a casa com o esfregão específico não atende ao princípio de inversão de dependência, já que o módulo de alto nível, neste caso é o trabalhador, depende de um módulo de baixo nível, neste caso é o esfregão específico. Em contrapartida, o trabalhador que limpa a casa com a ferramenta que estiver disponível condiz com o princípio, pois o módulo de alto nível está dissociado do módulo de baixo nível (ARAUJO, 2021).

Figura 16 – Exemplo do *Dependency Inversion Principle* na prática

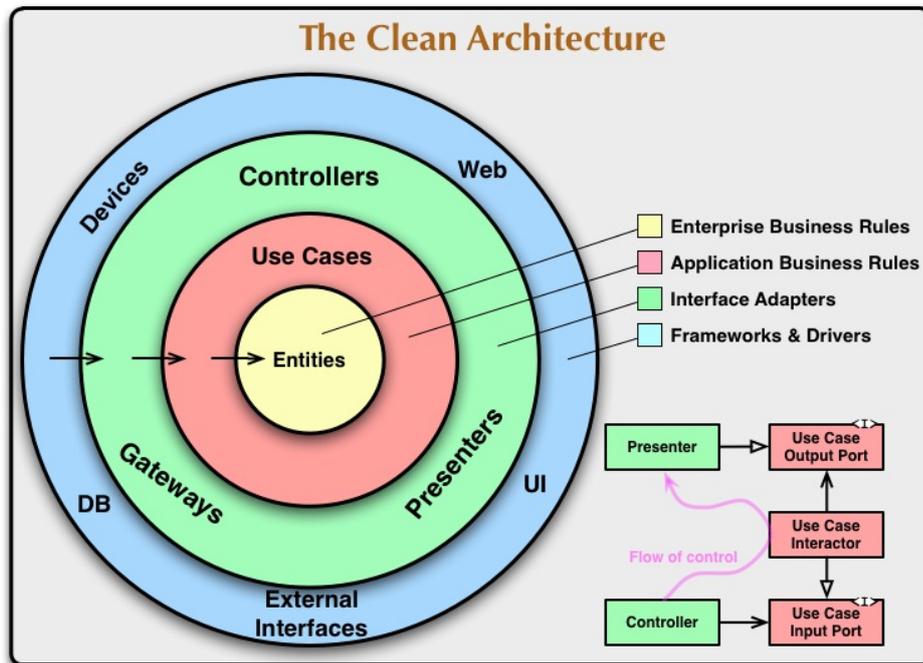


Fonte: (ARAUJO, 2021)

### 5.12 *Clean Architecture*

A arquitetura em camada é uma forma de organizar as partes de um software no intuito de decompor sistemas complexos para melhorar o entendimento e manutenção do mesmo. Como exemplo, pode ser citado a arquitetura Hexagonal e a arquitetura Cebola. Já a “*Clean Architecture*” é apresentada como uma alternativa de implementação dessas arquiteturas em camadas. O principal objetivo é a separação de preocupações, desmembrando regras de negócio e as interfaces. Cada camada no modelo da “*Clean Architecture*” representa uma parte diferente do software. A camada mais externa representa os mecanismos, como os *frameworks web* e banco de dados. Já as regras de negócio e o policiamento estão presentes nas camadas mais internas. A regra de dependência é o ponto principal desta arquitetura, ao qual estabelece que o nome de alguma coisa declarada em uma camada mais externa não pode ser mencionado no código nas camadas mais internas do ciclo. Isso inclui funções, classes, variáveis ou qualquer outra entidade de software. A dependência pode apontar apenas para dentro. Vale pôr em evidência que esta arquitetura não está limitada a certo conjunto definido de camadas, os desenvolvedores podem adicionar camadas contanto que sigam a regra de dependência. E para a transferência de dados através das camadas, pode-se criar um *Data Transfer Objects* (DTOs) (MARTIN, 2019).

A figura 17 é uma tentativa de integrar todas as arquiteturas em camadas em um única ideia associável sendo chamada de “*Clean Architecture*” (MARTIN, 2019).

Figura 17 – Camadas da *Clean Architecture*

Fonte: (MARTIN, 2019)

**Enterprise Business Rules:** Nessa camada geralmente são escritos estrutura de dados e algumas funções. Essa é a camada que fica no topo, portanto não é dependente de nenhuma outra camada. Depois de um projeto finalizado, uma mudança nessa camada afetaria todas as outras camadas, por isso temos que ter bastante cuidado nessa camada (MARTIN, 2019).

**Application Business Rules:** Aqui serão escritos todas as funcionalidades do projeto. Em outras palavras o que o projeto é capaz de fazer. Sendo os casos de uso as regras de negócios específicas do aplicativo, as alterações não devem impactar as Entidades. As alterações não devem ser afetadas pela infraestrutura, como por exemplo, um banco de dados (MARTIN, 2019).

**Interface Adapters:** Tem várias responsabilidades. A principal delas é ser a ponte entre *Application Business Rules* e *Framework and Drivers*. É aqui que você irá escrever a sequência dos seus processos para serem executados (MARTIN, 2019).

**Framework and Drivers:** É nessa camada que você finalmente pode acessar seu banco de dados e *framework*. Nessa camada você não pode acessar diretamente as camadas *Application Business Rules* ou *Enterprise Business Rules*. Essa camada só pode acessar a camada de *Interface Adapters* (MARTIN, 2019).

### 5.13 Banco de Dados - PostgreSQL

É um poderoso sistema de banco de dados relacionado a objeto, simbolizado pela figura 18, que utiliza a linguagem SQL combinada com muitas outras funcionalidades que armazenam e escalam de forma segura a mais complexa quantidade de dados. Tem uma boa reputação pela sua comprovada arquitetura, confiabilidade, integridade de dados, recursos de robustez, extensibilidade, e dedicação a comunidade de código aberto. Tem a preocupação de entregar software com melhores performances e desenvolver soluções inovadoras constantemente (POSTGRESQL, 2022).

Figura 18 – Banco de dados PostgreSQL



Fonte: (POSTGRESQL, 2022)

### 5.14 Banco de Dados - MongoDB

O MongoDB é um sistema de gerenciamento de banco de dados, simbolizado pela figura 19, feito para desenvolver aplicações *web* e infraestrutura da internet rapidamente. Oferece boa performance tanto para sistemas que precisam apenas de um node de banco de dados ou uma dúzia deles, já que oferece um bom suporte caso seja necessário escalar as aplicações. No entanto, as vantagens de usar o MongoDB não se limitam apenas a sistemas escaláveis. O seu grande diferencial está na construção mais intuitiva do seu modelo de dados. A informação é armazenada em forma de documento e não em tabelas, trazendo uma maior flexibilidade quanto à inserção de informações, pois não é necessário realizar grandes mudanças na modelagem do banco caso o formato de dados inserido mude (BANKER, 2016).

Figura 19 – Banco de dados MongoDB



Fonte: (MONGODB, 2022)

### 5.15 NodeJS

De acordo com Moraes (2021), o NodeJS é uma plataforma que possibilita a execução do *Javascript* no servidor através do desenvolvimento de programas assíncronos de alta performance que não dependem da perspectiva *multithreading* tradicional, mas usam E/S

de forma assíncrona com um padrão de programação orientado a eventos. Com a arquitetura *event loop*, ilustrada na figura 20, é possível solucionar problemas de alto tráfego na rede e de aplicações em tempo real, pois todos os processos são executados com uma *thread* única, ao invés de criar uma nova *thread* para cada requisição feita ao servidor.

Figura 20 – Arquitetura *event loop* do NodeJS



Fonte: (MORAES, 2021)

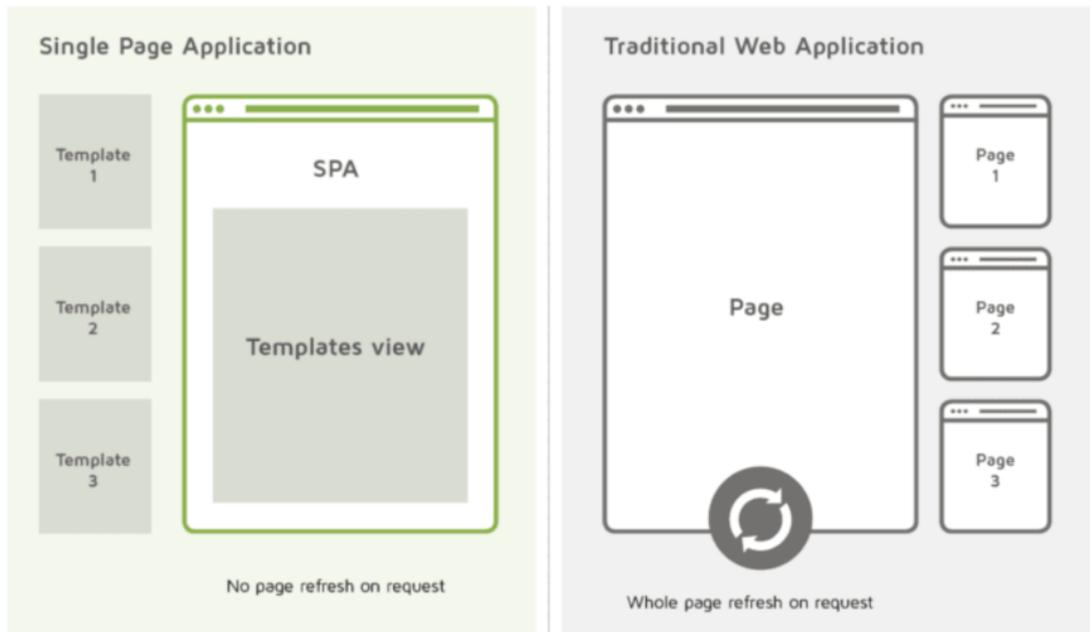
## 5.16 React

O React é uma biblioteca Javascript para criar interfaces de usuário em multiplataformas como *web*, *mobile*, *desktop* ou realidade virtual. A implementação de um sistema com React consiste na programação de fluxos de maneira assíncrona, ou seja, os componentes da interface podem ser processados e renderizados de forma simultânea para o usuário em tela. As bibliotecas designadas para o desenvolvimento da aplicação em sistema *Web* e *Mobile* são: ReactJS e React Native, respectivamente.

Segundo Wieruch (2017), React é a linguagem que define sua aparência e/ou comportamento através de bibliotecas HTML, CSS e Javascript, ou seja, uma vez que haja a definição de um componente, ele pode ser usado em uma hierarquia de itens para que um aplicativo inteiro possa ter funcionalidade. Ainda que o React tenha como finalidade o uso de itens como uma biblioteca, a estrutura na qual pertence o transforma em uma estrutura flexível e dinâmica em tela, tornando-a excelente em aplicações de tempo real.

A figura 21 ilustra um dos principais recursos pelos quais o React é utilizado por muitos desenvolvedores. Esta biblioteca de Javascript adota o modelo *Single Page Application* (SPA), que contrapõe ao modelo tradicional de estruturação de aplicações web chamado *Multiple Page Application* (MPA). Enquanto o primeiro permite a troca de páginas sem requisições custosas, o segundo necessita que cada página carregada abra esta conexão com o servidor (REDDY, 2020).

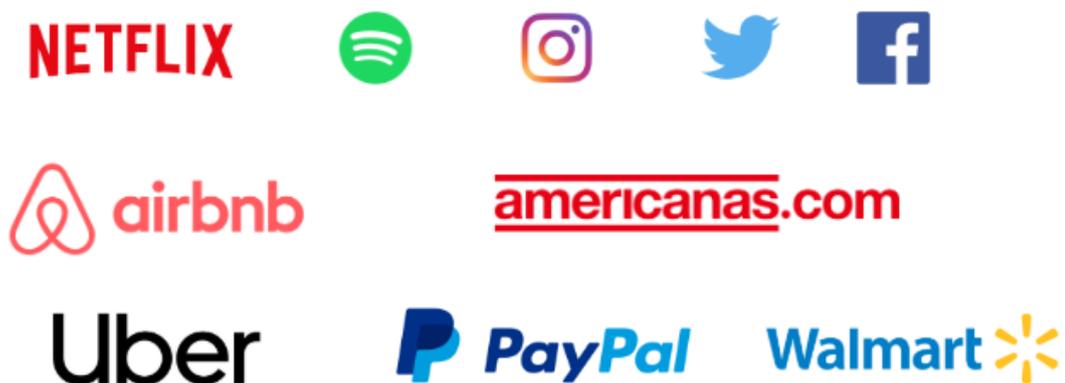
Figura 21 – Comparação entre o modelo padrão tradicional e o *Single Page Application*.



Fonte: (REDDY, 2020)

Até o momento deste trabalho, o react é utilizado por grandes empresas no mercado, indicadas na figura 22, como o próprio Facebook, Instagram, Uber, Netflix, Discord, Tesla, AirBnB e entre outras (FREIRE, 2019).

Figura 22 – Grandes empresas que implementam suas ferramentas com React.



Fonte: (FREIRE, 2019)

## 6 METODOLOGIA

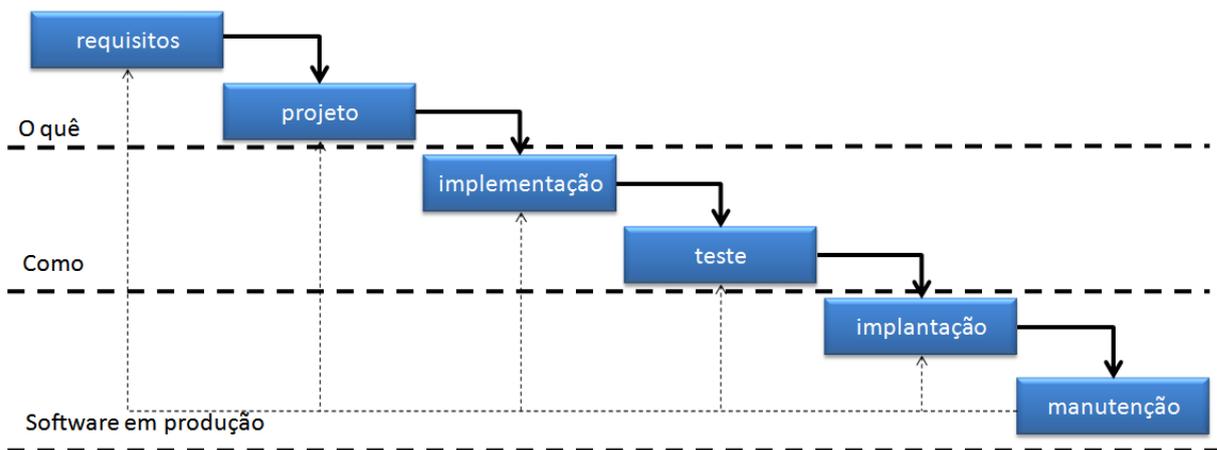
Na criação de todo projeto, um dos pontos mais importantes é o planejamento. Desenvolver um projeto que leva meses ou até anos para ser concluído e, no processo, descobrir que ele não atende ao usuário alvo, ou tem problemas estruturais, é uma situação indesejável para os desenvolvedores.

Neste trabalho, para melhorar o gerenciamento do projeto e diminuir a possibilidade de erros, foi utilizado alguns fundamentos da Engenharia de Software por estar ligado ao desenvolvimento de sistemas complexos dentro do prazo, com alta qualidade, e custos viáveis. Aspectos muito importantes para o sucesso do projeto.

A Engenharia de Software é o estudo e aplicação de uma abordagem sistemática, disciplinada e quantificável no desenvolvimento, na operação e na manutenção de software; isto é, a aplicação na engenharia de software (DEVMEDIA, 2013).

Conforme o processo representado na figura 23, o sucesso de um software envolve diversas tarefas como: análise de requisitos, modelagem de projeto, implementação de software, testes, implantação e manutenção. Estas etapas formam o chamado Modelo em Cascata.

Figura 23 – Modelo em Cascata.



Fonte: (FABRI, 2013)

O projeto descrito neste trabalho utilizou a área da engenharia de software chamada de análise de requisitos. Foram utilizados os casos de usos para apoiar o desenvolvimento. Após essa análise foi realizada a modelagem de bancos de dados levando em consideração as estruturas de dados identificadas nos casos de usos detalhados.

Nas sessões seguintes serão explanados com detalhes as áreas utilizadas. Por fim, é apresentado o diagrama de casos de uso da plataforma *Water Inspection* e suas descrições.

## 6.1 Análise de Requisitos

A análise de requisitos é o processo de levantamento das necessidades específicas para o resultado adequado de uma solução proposta. Nessa etapa é feito o reconhecimento, a avaliação e a especificação do problema. A coleta de informações geralmente é realizada através das seguintes técnicas:

- **Entrevista:** Etapa na qual os analistas conversam com um cliente;
- **Questionário:** Nessa técnica várias perguntas são respondidas podendo ser abertas ou fechadas. Esse método geralmente é utilizado quando as pessoas têm muito conhecimento sobre o problema;
- **Observação:** Nessa técnica o analista de requisito vai *in loco* observar as situações dia-a-dia do lugar;
- **Brainstorming:** São reuniões realizadas com todo o nível de usuário do projeto. Dessa forma surgem várias ideias e pontos de vistas, são todos ouvidos e depois são escolhidas as melhores ideias para o desenvolvimento;

Os requisitos funcionais são declarações de funções que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e como deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais podem também explicitamente declarar o que o sistema não deve fazer. Baseado nas técnicas mencionadas anteriormente, a seguir serão apresentados os RF e Requisitos Não Funcionais (RNF) do sistema desenvolvido neste trabalho.

### 6.1.1 Lista de Requisitos

Na tabela 1, representa todos os requisitos funcionais elaborados para o sistema e eles foram classificados em: Essencial (ESS), Importante (IMP), Desejável (DES).

Tabela 1 – Lista de requisitos funcionais

<b>CÓDIGO</b>	<b>NOME</b>	<b>ESS</b>	<b>IMP</b>	<b>DES</b>
WI-SW-01	Exibir localização dos PDM's	x		
WI-SW-02	Exibir informações de análise da água em Dashboard	x		
WI-SW-03	Exibir alertas de parâmetros fora dos limites estabelecidos e violação de um PDM	x		
WI-SW-04	Criar conta de usuário	x		
WI-SW-05	Alterar dados de usuário	x		
WI-SW-06	Excluir conta de usuário	x		
WI-SW-07	Realizar autenticação	x		
WI-SW-08	Recuperar acesso	x		
WI-SW-09	Disponibilizar Dados Recebidos	x		
WI-SW-10	Consultar Dados Recebidos	x		
WI-SW-11	Gerar relatórios			x
WI-SW-12	Configurar posto de monitoramento		x	
WI-SW-13	Solicitar Leitura Instantânea		x	
WI-SW-14	Exibir status de PDM's instaladas	x		
WI-SW-15	Fechar Evento de Anomalia de parâmetros			x
WI-SW-16	Realizar Logout	x		

Fonte: (Elaborado pelo autor, 2022)

### 6.1.2 Descrição dos Requisitos.

Tabela 2 – (RF) Exibir localização dos PDM's

WI-SW-01	Exibir localização dos PDM's
[x] Essencial [ ] Importante [ ] Desejável	
<b>Descrição:</b>	
<p>O sistema deve exibir um mapa apresentando a localização de um determinado Posto de Monitoramento (PDM), mostrando o status do dispositivo (online/offline), sendo possível selecionar qual PDM o usuário deseja visualizar as informações que estão sendo coletadas e exibidas em dashboard.</p> <p>Essa funcionalidade deverá ser acessível a todos os níveis de acesso.</p>	

Fonte: (Elaborado pelo autor, 2022)

Tabela 3 – (RF) Exibir informações de análise da água em Dashboard

WI-SW-02	Exibir informações de análise da água em Dashboard
[x] Essencial [ ] Importante [ ] Desejável	
<b>Descrição:</b>	
<p>O sistema deve exibir os dados recebidos do PDM de controle em dashboard:</p> <ul style="list-style-type: none"> <li>• Mostrar gráficos com os dados dos sensores (pH (pH), Oxigênio Dissolvido (ppm), Condutividade Elétrica (<math>\mu\text{S}/\text{cm}</math>), Turbidez (NTU), Temperatura da Água (<math>^{\circ}\text{C}</math>) e Temperatura do Ar (<math>^{\circ}\text{C}</math>)).</li> <li>• Mostrar os parâmetros que serão calculados a partir dos coletados (Sólidos Totais Dissolvidos (ppm) e Salinidade (ppt)).</li> </ul> <p>Essa funcionalidade deverá ser acessível ao usuário administrador e para o usuário analista.</p>	

Fonte: (Elaborado pelo autor, 2022)

Tabela 4 – (RF) Exibir alertas de parâmetros fora dos limites estabelecidos e violação de um PDM

<b>WI-SW-03</b>	<b>Exibir alertas de parâmetros fora dos limites estabelecidos e violação de um PDM</b>
[x] Essencial [ ] Importante [ ] Desejável	
<b>Descrição:</b>	
O sistema deve exibir alertas e alarmes dos dispositivos de coleta dos parâmetros da água: <ul style="list-style-type: none"> <li>• Alerta: variações fora do limite estabelecido no sistema nos dados coletados pelos sensores do dispositivo.</li> <li>• Alarme: falha ou ausência dos sensores, abertura do case e falha no módulo de energia.</li> </ul> Essa funcionalidade deverá ser acessível ao usuário administrador e para o usuário analista.	

Fonte: (Elaborado pelo autor, 2022)

Tabela 5 – (RF) Criar conta de usuário

<b>WI-SW-04</b>	<b>Criar conta de usuário</b>
[x] Essencial [ ] Importante [ ] Desejável	
<b>Descrição:</b>	
O sistema deve possuir a opção para a criação de um usuário. Devem ser atribuídas a cada usuário as seguintes informações: <ul style="list-style-type: none"> <li>• Nome;</li> <li>• Sobrenome;</li> <li>• Email;</li> <li>• Cidade;</li> <li>• Estado;</li> <li>• Função: Analista ou Administrador;</li> <li>• Status: Ativo ou inativo;</li> </ul> Essa funcionalidade deverá ser acessível apenas aos usuários administradores.	

Fonte: (Elaborado pelo autor, 2022)

Tabela 6 – (RF) Alterar dados de usuário

<b>WI-SW-05</b>	<b>Alterar dados de usuário</b>
[x] Essencial [ ] Importante [ ] Desejável	
<b>Descrição:</b>	
O sistema deve possuir a opção de editar os dados de um usuário já cadastrado. Os dados são conforme o requisito WI-SW-04.	
Essa funcionalidade deverá ser acessível apenas aos usuários administradores.	

Fonte: (Elaborado pelo autor, 2022)

Tabela 7 – (RF) Excluir conta de usuário

<b>WI-SW-06</b>	<b>Excluir conta de usuário</b>
[x] Essencial [ ] Importante [ ] Desejável	
<b>Descrição:</b>	
O sistema deve possuir a opção de excluir um usuário já cadastrado em um sistema:	
Essa funcionalidade deverá ser acessível apenas aos usuários administradores.	

Fonte: (Elaborado pelo autor, 2022)

Tabela 8 – (RF) Realizar autenticação

<b>WI-SW-07</b>	<b>Realizar autenticação</b>
[x] Essencial [ ] Importante [ ] Desejável	
<b>Descrição:</b>	
O sistema realizar a autenticação de um usuário com e-mail e senha já cadastrados para então ter acesso às demais funcionalidades do software:	
Essa funcionalidade deverá ser acessível ao usuário administrador e ao usuário analista.	

Fonte: (Elaborado pelo autor, 2022)

Tabela 9 – (RF) Recuperar acesso

<b>WI-SW-08</b>	<b>Recuperar acesso</b>
[x] Essencial [ ] Importante [ ] Desejável	
<b>Descrição:</b>	
O sistema deve prover a recuperação de senha perdida através do envio de email para a redefinição de senha.	
Essa funcionalidade deverá ser acessível ao usuário administrador e ao usuário analista.	

Fonte: (Elaborado pelo autor, 2022)

Tabela 10 – (RF) Disponibilizar dados recebidos

<b>WI-SW-09</b>	<b>Disponibilizar dados recebidos</b>
[x] Essencial [ ] Importante [ ] Desejável	
<b>Descrição:</b>	
O sistema deve mostrar ao usuário autenticado uma opção para consultar os valores instantâneos dos dados coletados no dispositivo da PDM (pH (pH), Oxigênio Dissolvido (ppm), Condutividade Elétrica ( $\mu\text{S}/\text{cm}$ ), Turbidez (NTU), Temperatura da Água ( $^{\circ}\text{C}$ ) e Temperatura do Ar ( $^{\circ}\text{C}$ ))	
Mostrar os parâmetros que serão calculados a partir dos coletados (Sólidos Totais Dissolvidos (ppm) e Salinidade (PSU)) fornecidos pelo dispositivo.	
Essa funcionalidade deverá ser acessível ao usuário administrador e ao usuário analista.	

Fonte: (Elaborado pelo autor, 2022)

Tabela 11 – (RF) Receber eventos de dispositivos

<b>WI-SW-10</b>	<b>Receber eventos de dispositivos</b>
[x] Essencial [ ] Importante [ ] Desejável	
<b>Descrição:</b>	
O sistema deve receber eventos de mensagens de um dispositivo de coleta de dados. Os possíveis eventos são: falha ou ausência dos sensores, abertura do case, bateria fraca, ou interrupção de energia elétrica.	
Essa funcionalidade deverá ser acessível ao usuário administrador e ao usuário analista.	

Fonte: (Elaborado pelo autor, 2022)

Tabela 12 – (RF) Gerar relatórios

<b>WI-SW-11</b>	<b>Gerar relatórios</b>
[ ] Essencial [ ] Importante [x] Desejável	
<b>Descrição:</b>	
<p>O sistema deve permitir que usuário gere relatórios com os dados coletados, a partir das informações coletadas no PDM. O Usuário deve escolher o intervalo de tempo e selecionar os parâmetros físico-químicos que deseja estar presente no arquivo csv gerado pelo sistema.</p> <p>O arquivo csv deve conter as amostras coletadas com a data/hora, os parâmetros selecionados pelo usuário, e a identificação do PDM. Essa funcionalidade deverá ser acessível ao usuário administrador e ao usuário analista.</p>	

Fonte: (Elaborado pelo autor, 2022)

Tabela 13 – (RF) Configurar posto de monitoramento

<b>WI-SW-12</b>	<b>Configurar posto de monitoramento</b>
[x] Essencial [ ] Importante [ ] Desejável	
<b>Descrição:</b>	
<p>O sistema deve permitir a configuração de um PDM, alterando dados como: a identificação das unidades remotas, configuração da comunicação, configuração da amostragem no modo normal, no modo anomalia, leitura em tempo real e faixa de valores de referência dos parâmetros para detecção de anomalias. Essa funcionalidade deverá ser acessível apenas ao usuário administrador.</p>	

Fonte: (Elaborado pelo autor, 2022)

Tabela 14 – (RF) Solicitar Leitura Instantânea

<b>WI-SW-13</b>	<b>Solicitar Leitura Instantânea</b>
[ ] Essencial [x] Importante [ ] Desejável	
<b>Descrição:</b>	
<p>O usuário pode solicitar uma leitura instantânea dos parâmetros físico-químicos da água fornecido pelos sensores para o PDM. Essa funcionalidade deverá ser acessível para o usuário administrador e o analista.</p>	

Fonte: (Elaborado pelo autor, 2022)

Tabela 15 – (RF) Exibir status de PDM's instaladas

<b>WI-SW-14</b>	<b>Exibir status de PDM's instaladas</b>
[x] Essencial [ ] Importante [ ] Desejável	
<b>Descrição:</b>	
O Sistema deve informar o status de comunicação do dispositivo associado a uma PDM, <i>Online</i> ou <i>Offline</i> . Além disso, deve informar quando o PDM estiver em estado de anomalia e qual o modo de energia o dispositivo está operando. Essa funcionalidade deverá ser acessível para o usuário administrador e o analista.	

Fonte: (Elaborado pelo autor, 2022)

Tabela 16 – (RF) Fechar Evento de Anomalia de parâmetros

<b>WI-SW-15</b>	<b>Fechar Evento de Anomalia de parâmetros</b>
[ ] Essencial [ ] Importante [x] Desejável	
<b>Descrição:</b>	
Pode haver o fechamento de evento detectado pela PDM de controle pelo usuário, após a identificação da mensagem de alerta. O sistema passará a identificar uma nova anomalia depois de um tempo determinado pelo usuário administrador na seção de configuração de um PDM.	
Essa funcionalidade deverá ser acessível somente para o usuário administrador.	

Fonte: (Elaborado pelo autor, 2022)

Tabela 17 – (RF) Realizar Logout

<b>WI-SW-16</b>	<b>Realizar Logout</b>
[x] Essencial [ ] Importante [ ] Desejável	
<b>Descrição:</b>	
Como o sistema salva a sessão do usuário autenticado, consequentemente, ele deve possuir a opção para realizar logout.	
Essa funcionalidade deverá ser acessível para o usuário administrador e analista.	

Fonte: (Elaborado pelo autor, 2022)

Na tabela 18, lista e descreve os requisitos não-funcionais do sistema, classificados por categoria na engenharia de software.

Tabela 18 – Requisitos Não-Funcionais

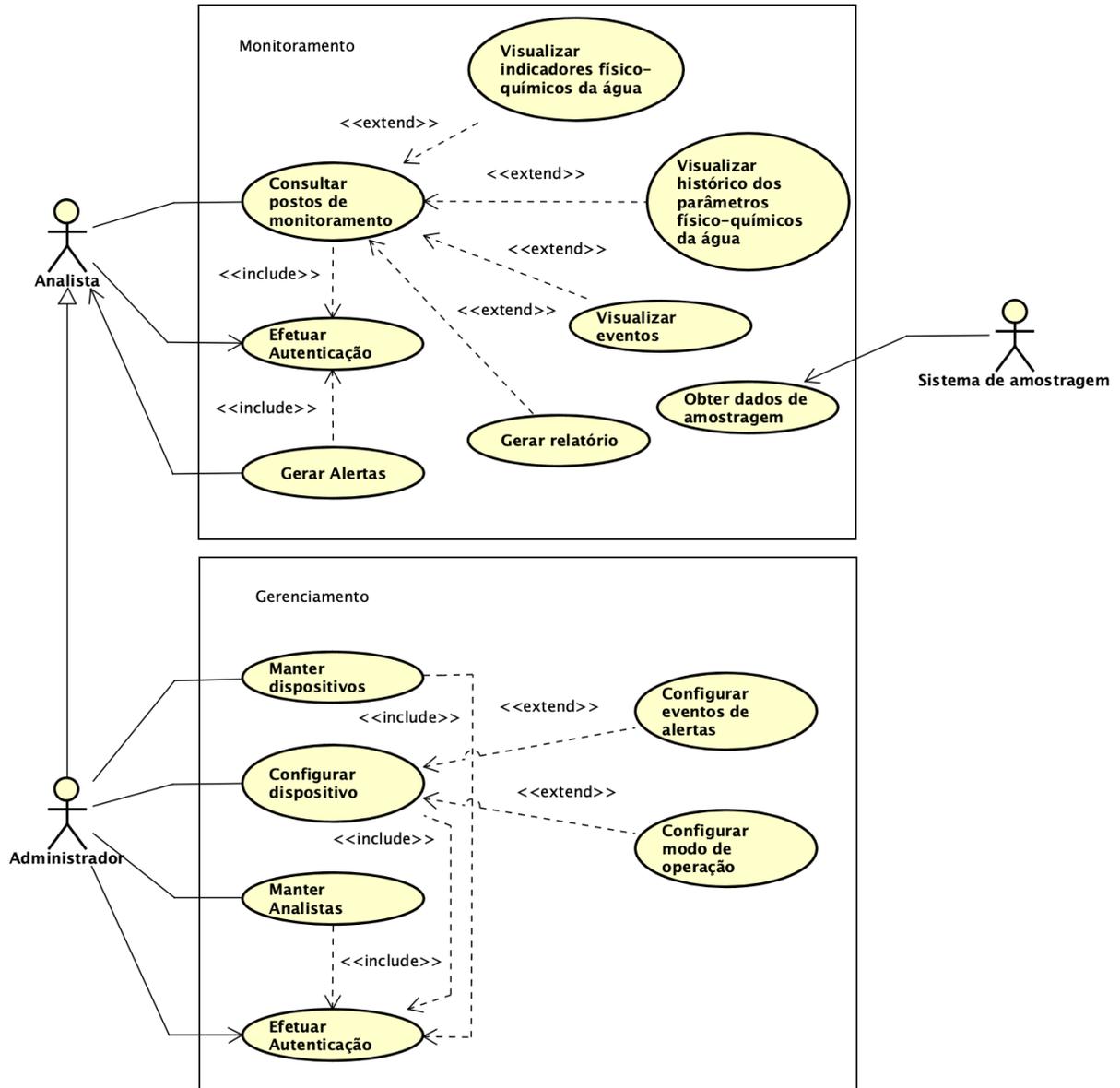
<b>CÓDIGO</b>	<b>DESCRIÇÃO</b>	<b>Categoria</b>
WI-SW-NF-01	O sistema deverá ser em uma plataforma web	Portabilidade
WI-SW-NF-02	O sistema deverá garantir que o acesso de senha seja criptografado pelo método JWT.	Segurança
WI-SW-NF-03	O sistema tem que está disponível 24 horas por dia sem interrupções (exceto para manutenções)	Disponibilidade
WI-SW-NF-04	O sistema deve possuir uma interface com boa usabilidade e um design responsivo.	Usabilidade
WI-SW-NF-05	O sistema deve exibir dados em tempo real	Usabilidade
WI-SW-NF-06	Contador de acessos realizados	Segurança
WI-SW-NF-07	O sistema deve ser capaz de receber os dados que estão sendo coletados por um dispositivo, através de uma fila SQS da AWS.	Comunicação

Fonte: (Elaborado pelo autor, 2022)

## 6.2 Caso de Uso

O diagrama da figura 24 contém todos os casos de uso do projeto. Estes serão melhor detalhados na seção 6.3.

Figura 24 – Diagrama de caso de uso.



Fonte: (Elaborado pelo autor, 2022)

### 6.3 Caso de Uso detalhado

A seguir, estão detalhados todos os casos de uso do projeto.

Tabela 19 – (Caso de Uso) Manter analistas

<b>UC001 – Manter analistas</b>	
Ator:	Administrador
Descrição:	Este caso de uso descreve os procedimentos de realizar o cadastro, editar, excluir e visualizar analistas.
Pré-Condição:	Não existe.
Pós-Condições:	Realizar login no sistema
Cenário Principal:	<ol style="list-style-type: none"> <li>1. Incluir um novo usuário.</li> <li>2. O administrador cadastra as informações dos analistas.</li> <li>3. Se as informações estiverem dentro dos padrões requeridos</li> <li>4. Gerar conta de usuário.</li> <li>5. Administrador solicita perfil de usuário (administrador ou analista).</li> </ol>
Cenário Alternativo:	<ol style="list-style-type: none"> <li>1. O administrador poderá editar as informações de cada analista.</li> <li>2. O administrador poderá excluir um analista do sistema.</li> <li>3. O administrador poderá visualizar a listagem de usuários em um sistema.</li> </ol>
Exceções:	O sistema identifica os campos que foram inseridos fora do formato esperado para a criação do usuário.

Fonte: (Elaborado pelo autor, 2022)

Tabela 20 – (Caso de Uso) Fazer Login

<b>UC002 – Fazer Login</b>	
Ator:	Administrador/Analista
Descrição:	Este caso de uso descreve os procedimentos para realizar o login do sistema.
Pré-Condição:	O usuário deve estar cadastrado no sistema.
Pós-Condições:	Acesso ao sistema.
Cenário Principal:	<ol style="list-style-type: none"> <li>1. O ator inicia o sistema.</li> <li>2. O sistema apresenta a tela para login.</li> <li>3. O administrador/analista informa o usuário e a senha, e clica na opção entrar.</li> <li>4. O sistema verifica os dados e encerra o caso de uso.</li> </ol>
Cenário Alternativo:	<ol style="list-style-type: none"> <li>1. O usuário já autenticado, não precisará efetuar o login novamente enquanto não fazer o logout.</li> <li>2. O usuário digita o login e clica no link “esqueci a senha”;</li> <li>3. A senha será enviada para o email do usuário.</li> <li>4. Uma mensagem será exibida na tela informando que a senha foi enviada para o email cadastrado pelo usuário.</li> </ol>
Exceções:	O sistema exibe a mensagem de que a senha ou login estão incorretos, e mantêm-se na tela de Login.

Fonte: (Elaborado pelo autor, 2022)

Tabela 21 – (Caso de Uso) Consultar Postos de Monitoramento

<b>UC003 – Consultar Postos de Monitoramento</b>	
Ator:	Administrador/Analista
Descrição:	O ator terá uma visão geral de todos os postos de monitoramento cadastrados no sistema. Sendo evidenciado seu status e localização.
Pré-Condição:	O usuário deve ser cadastrado e haver postos de monitoramento cadastrados no sistema.
Pós-Condições:	Ser possível acessar PDM.
Cenário Principal:	<ol style="list-style-type: none"> <li>1. Após realizar login, o usuário é direcionado à visualização de um mapa.</li> <li>2. O mapa conterà todos os postos de atendimento cadastrados no sistema.</li> <li>3. Será possível identificar o status de cada posto de monitoramento: alarme, online e offline.</li> <li>4. Ter uma visão geral da quantidade de cada dispositivo que apresenta determinado status.</li> </ol>
Cenário Alternativo:	Caso não seja possível visualizar as devidas informações no mapa, o ator poderá consultá-las em uma tabela.
Exceções:	Caso não haja nenhum Posto de Monitoramento cadastrado no sistema o mapa será exibido sem nenhum ícone de PDM, além disso, será exibido um modal com a informação de falta de Postos de Monitoramento cadastrados no sistema.

Fonte: (Elaborado pelo autor, 2022)

Tabela 22 – (Caso de Uso) Visualizar histórico dos parâmetros físico-químicos da água

<b>UC004 – Visualizar histórico dos parâmetros físico-químicos da água</b>	
Ator:	Administrador/Analista
Descrição:	O ator terá uma visão analítica do histórico dos parâmetros físico-químicos da água através de um gráfico.
Pré-Condição:	Selecionar um dispositivo para a visualização do histórico do mesmo.
Pós-Condições:	Gerar relatório do histórico dos parâmetros físico-químicos da água.
Cenário Principal:	<ol style="list-style-type: none"> <li>1. O ator selecionará um dispositivo para a visualização do histórico.</li> <li>2. O ator selecionará um parâmetro físico-químico para analisar o histórico, sendo eles: temperatura da água, temperatura do ar, turbidez, ph, oxigênio dissolvido, condutividade elétrica, sólidos totais dissolvidos e salinidade.</li> <li>3. O ator selecionará um intervalo de tempo para a visualização do histórico.</li> <li>4. Será possível visualizar os valores de cada parâmetro em um gráfico.</li> </ol>
Cenário Alternativo:	Não existe.
Exceções:	O sistema apresentará um alerta informando que não foi possível consultar as informações do histórico.

Fonte: (Elaborado pelo autor, 2022)

Tabela 23 – (Caso de Uso) Gerar relatório

<b>UC005 – Gerar relatório</b>	
Ator:	Administrador/Analista
Descrição:	O usuário obterá os dados de parâmetros físico-químicos da água dentro de um intervalo de tempo no formato csv.
Pré-Condição:	Selecionar os parâmetros físico-químicos da água e um intervalo de tempo.
Pós-Condições:	Não existe.
Cenário Principal:	<ol style="list-style-type: none"> <li>1. O ator seleciona os parâmetros físico-químicos das água sendo eles: temperatura da água, temperatura do ar, turbidez, ph, oxigênio dissolvido, condutividade elétrica, sólidos totais dissolvidos e salinidade. (também há a possibilidade de selecionar todos os parâmetros existentes)</li> <li>2. O ator seleciona o intervalo de tempo que deseja obter o relatório</li> <li>3. Gerado uma planilha csv disponível para download no dispositivo do ator.</li> </ol>
Cenário Alternativo:	Não existe.
Exceções:	Caso não seja possível gerar o relatório o sistema emite um alerta ao usuário com o respectivo problema.

Fonte: (Elaborado pelo autor, 2022)

Tabela 24 – (Caso de Uso) Visualizar eventos e alarmes

<b>UC006 – Visualizar eventos e alarmes</b>	
Ator:	Administrador/Analista
Descrição:	O ator poderá ter uma visão geral dos eventos emitidos pelos PDM's.
Pré-Condição:	Os Postos de Monitoramento devem captar valores de parâmetros físico-químicos da água que estejam fora dos limites estabelecidos.
Pós-Condições:	Não existe.
Cenário Principal:	<ol style="list-style-type: none"> <li>1. O usuário poderá visualizar os eventos dos PDM's. O evento é classificado por uma mensagem reportada pelo dispositivo de um PDM.</li> <li>2. O usuário poderá visualizar alarmes de uma PDM's. O alarmes é classificado por valores de parâmetros físico-químicos que estejam fora dos limites máximo e mínimo previamente estabelecidos.</li> <li>3. Cada alarme terá o parâmetro ao qual pertence destacado pelo sistema Cada evento e alarme terá a data e a hora em que ocorreu destacado pelo sistema</li> <li>4. Cada alarme terá a “Criticidade” destacado pelo sistema, sendo esta caracterizada dada pela variância do valor padrão e o valor atual dado em porcentagem.</li> </ol>
Cenário Alternativo:	Não existe.
Exceções:	Não existe.

Fonte: (Elaborado pelo autor, 2022)

Tabela 25 – (Caso de Uso) Obter dados da amostragem

<b>UC007 – Obter dados da amostragem</b>	
Ator:	Administrador/Analista
Descrição:	O sistema receberá dados dos Postos de Monitoramento através de amostras publicadas em uma fila SQS da AWS.
Pré-Condição:	Desenvolver o <i>script</i> compatível com o protocolo de comunicação ao qual o PDM atende.
Pós-Condições:	Armazenar os dados recebidos no banco de dados.
Cenário Principal:	<ol style="list-style-type: none"> <li>1. O ator irá captar os dados dos Postos de Monitoramento por meio do protocolo de comunicação correspondente.</li> <li>2. O ator publica os dados em um tópico SNS da AWS.</li> <li>3. O tópico SNS entrega esta informação a uma fila SQS da AWS.</li> <li>4. O sistema recebe os dados e exibe informações dos Postos de Monitoramento do sistema</li> </ol>
Cenário Alternativo:	Não existe.
Exceções:	Não existe.

Fonte: (Elaborado pelo autor, 2022)

Tabela 26 – (Caso de Uso) Manter Posto de Monitoramento

<b>UC008 – Manter Posto de Monitoramento</b>	
Ator:	Administrador
Descrição:	O administrador poderá realizar a criação e edição das informações básicas de um Posto de Monitoramento.
Pré-Condição:	Ter informações corretas a serem cadastradas.
Pós-Condições:	Não existe.
Cenário Principal:	<ol style="list-style-type: none"> <li>1. O ator insere as seguintes informações correspondentes para cada dispositivo: código, modelo, tipo de energia, bacia, rio, ponto de instalação, UF, município, latitude, longitude, vazão, cor da Água, tipo do sistema aquático, período de certificação, e classificação do curso da água.</li> <li>2. O ator confirma os dados;</li> <li>3. Os dados referentes ao Posto de Monitoramento são salvos no banco de dados</li> </ol>
Cenário Alternativo:	Caso o ator ache necessário realizar alguma atualização ou correção dos dados cadastrais de um Posto de Monitoramento, é possível realizar a edição de qualquer um dos campos previamente cadastrados.
Exceções:	Caso o ator perceba que o Posto de Monitoramento não seja mais produtivo ao sistema, será possível realizar a exclusão do mesmo dos banco de dados.

Fonte: (Elaborado pelo autor, 2022)

Tabela 27 – (Caso de Uso) Configurar eventos de alertas

<b>UC009 – Configurar eventos de alertas</b>	
Ator:	Administrador
Descrição:	O administrador poderá inserir os valores padrão para todos os parâmetros físico-químicos da água.
Pré-Condição:	Obter os valores a serem cadastrados como padrão no sistema.
Pós-Condições:	O sistema deve exibir emitir alarmes ao usuário tomando em consideração os valores padrões de cada parâmetro. Deve ser descrito nos eventos a criticidade com base nos valores padrões inseridos.
Cenário Principal:	<p>1. O ator deverá inserir os valores de máximo e mínimo, de acordo com dados base, dos seguintes parâmetros:</p> <ul style="list-style-type: none"> <li>• Oxigênio Dissolvido dado em ppm.</li> <li>• Condutividade Elétrica dado em uS/cm.</li> <li>• PH dado em Ph.</li> <li>• Turbidez dado em NTU.</li> <li>• Temperatura da Água dado em °C.</li> <li>• Temperatura do Ar dado em °C.</li> <li>• Sólidos Totais Dissolvidos dado em ppm.</li> <li>• Salinidade dado em PSU.</li> </ul>
Cenário Alternativo:	Caso não haja limites estabelecidos para algum dos parâmetros, o sistema não emitirá alarmes ao usuário referentes a aquele parâmetro.
Exceções:	Não existe.

Fonte: (Elaborado pelo autor, 2022)

Tabela 28 – (Caso de Uso) Configurar modo de operação

<b>UC010 – Configurar modo de operação</b>	
Ator:	Administrador
Descrição:	O ator terá a possibilidade de configurar o parâmetro “taxa de amostragem” de um dispositivo cadastrado no sistema.
Pré-Condição:	O dispositivo cadastrado no sistema deve ter o parâmetro “taxa de amostragem” (Tempo para coleta e envio de amostras dos parâmetros da água), configurável, incluindo a identificação da origem do fornecimento de energia sendo elas da própria concessionária, de uma bateria ou ainda de um sistema de energia solar.
Pós-Condições:	Não existe.
Cenário Principal:	<ol style="list-style-type: none"> <li>1. O ator acessa a seção de configuração de um Posto de Monitoramento.</li> <li>2. O ator pode configurar a taxa de amostragem para três casos: <ul style="list-style-type: none"> <li>• O dispositivo esteja conectado a energia da concessionária de energia.</li> <li>• O dispositivo tenha o seu suprimento de energia fornecido pela concessionária e tenha sido necessário o abastecimento por meio de uma bateria.</li> <li>• O dispositivo esteja conectado à energia solar</li> </ul> </li> </ol>
Cenário Alternativo:	O PDM encontra-se em estado de anomalia, ou seja, é quando há a coleta de parâmetros fora dos limites estabelecidos, entrando em status de Alerta.
Exceções:	Caso o dispositivo não apresente as pré-condições, este caso de uso não atenderá o mesmo.

Fonte: (Elaborado pelo autor, 2022)

Os fatores mais descritivos de modelagem do sistema foram colocados nessa seção. Os demais arquivos de modelagem *Unified Modeling Language* (UML) feitos encontram-se ao final deste documento no Apêndice A.

## 7 DESENVOLVIMENTO

Neste capítulo serão abordados os passos do desenvolvimento do sistema. Primeiramente, será discutido o conceito da plataforma e os indicadores escolhidos para contemplar a qualidade da água baseada em seus parâmetros físico-químicos. Depois, será descrita como foi feita a elaboração de *layout* das páginas do sistema. Após isto, serão discutidas a arquitetura geral e em nuvem do sistema para a implementação e o motivo de suas escolhas. Por fim, será abordado o processo de desenvolvimento do *frontend* e *backend* que constituem a plataforma, discorrendo sobre sua estrutura de pastas e arquivos com padrões utilizados.

### 7.1 *Water Inspection*: A plataforma para avaliação da qualidade da água

A plataforma desenvolvida para monitoramento da qualidade da água é chamada de *Water Inspection*, com tradução literal do inglês "Inspeção de água". Seu principal objetivo é ser uma plataforma para monitorar os parâmetros físico-químicos da água, permitindo que dados de qualquer sensor sejam integrados ao sistema. É importante ressaltar que este processo ocorre independente do fabricante e/ou protocolo de comunicação. Além disso, a plataforma possibilita exibir alguns parâmetros físico-químicos da água a fim de contribuir para o monitoramento de sua qualidade e auxiliar as ações de especialistas quanto à conservação dos recursos hídricos.

Uma parte importante do conceito da plataforma é a independência de um sensor específico de um determinado fabricante ou protocolo de comunicação, ou seja, com a plataforma desenvolvida neste trabalho, é possível integrar o sistema com qualquer sensor capaz de fornecer algum parâmetro físico-químico da água estabelecido no sistema, através dos protocolos de comunicação já existentes no mercado.

Os usuários da plataforma assumem dois principais papéis: o administrador do sistema e o analista. O administrador do sistema é aquele que:

- Configura um posto de monitoramento com informações básicas para identificação do posto de água em monitoramento.
- Estabelece limite inferior e superior para cada parâmetro físico-químico da água a fim de propiciar eventos de anomalias e gerar alertas a fim de estabelecer a situação da qualidade da água.
- Manter usuários com a função de analista ou outros gestores no sistema com cadastro, edição e remoção.

O usuário com o papel de analista é aquele que poderá acessar um determinado posto e consultar os valores em tempo real dos parâmetros físico-químicos da água, para assim, diagnosticar a situação da mesma e verificar o comportamento das mudanças de cada parâmetro através do gráfico no tempo. Além disso, o analista poderá obter os eventos de anomalia que o sistema pode detectar mediante aos dados obtidos dos sensores e os limites estabelecidos pelo usuário gestor. Caso o analista queira os dados brutos coletados pelo sistema, a plataforma possibilita exportar os dados coletados em planilha csv de acordo com a escolha dos parâmetros na seção de geração de relatórios. A identidade visual do sistema desenvolvido está representado na figura 25.

Figura 25 – Identidade visual da plataforma *Water Inspection*.



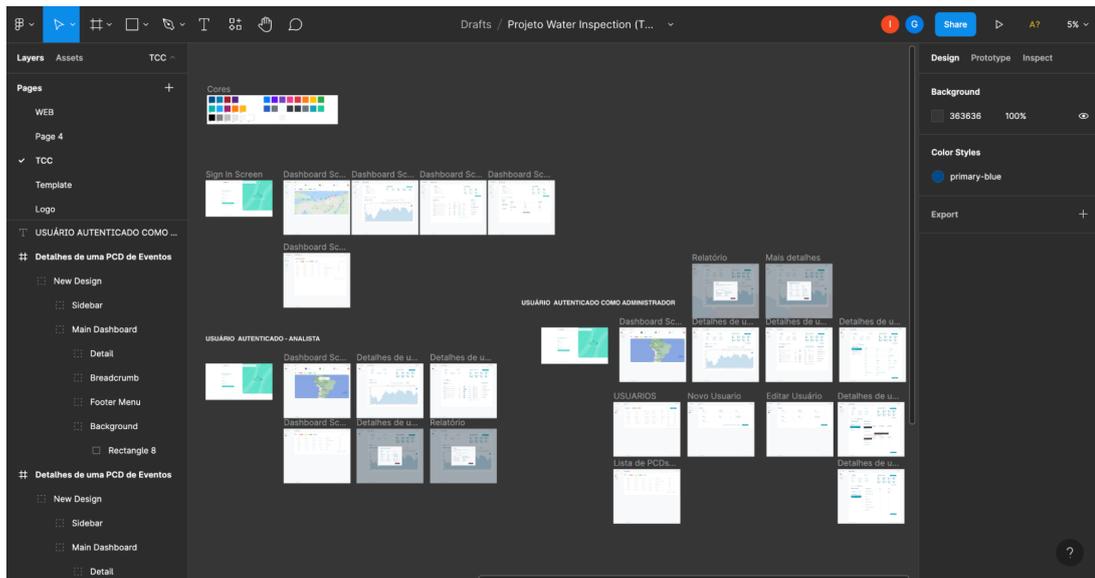
Fonte: (Elaborado pelo autor, 2022)

## 7.2 Prototipação (UI/UX)

No desenvolvimento de software, um protótipo é um modelo de funcionamento rudimentar de um produto ou sistema de informação, geralmente construído para fins de demonstração ou como parte do processo de desenvolvimento.

Neste projeto de pesquisa, conforme a figura 26, uma versão básica do sistema foi construída, testada e retrabalhada conforme necessário até que um protótipo aceitável fosse finalmente alcançado, a partir do qual o sistema completo pode ser desenvolvido. Para isso, foi utilizado o Figma. Esta ferramenta é um editor online de design gráfico focado na criação de *User Interface* (UI) e *User Experience* (UX) com ênfase colaborativa. Com essa plataforma foi possível projetar, prototipar, compartilhar e receber feedbacks de possíveis usuários do sistema.

Figura 26 – Prototipação da plataforma *Water Inspection*.



Fonte: (Elaborado pelo autor, 2022)

A prototipação foi fundamental para a identificação de problemas na idealização do software e facilitou o processo de validação através de *feedback* e materialização de uma ideia. Usando o *feedback*, tanto as especificações quanto o protótipo poderiam ser melhorados. Com isso, foi possível reduzir riscos e a partir do primeiro protótipo, foi possível evoluir mais a solução.

Além disso, a prototipação promoveu uma visão da precisão das estimativas iniciais do projeto e se os prazos e marcos propostos poderiam ser atendidos com sucesso.

### 7.3 Arquitetura geral do sistema

Para o desenvolvimento do sistema, a arquitetura escolhida foi a *Clean Architecture*, a fim de promover uma implementação que propicie uma legibilidade e entendimento de código para futuras possíveis manutenções. Além disso, este modelo arquitetural favorece a reusabilidade de código, coesão, independência de tecnologia e estabilidade.

O sistema visa disponibilizar uma solução para o monitoramento dos parâmetros da qualidade da água que estão sendo coletados em um dispositivo de sensoriamento.

A visão geral da arquitetura do sistema está dividida em três módulos como mostra a Figura 27.

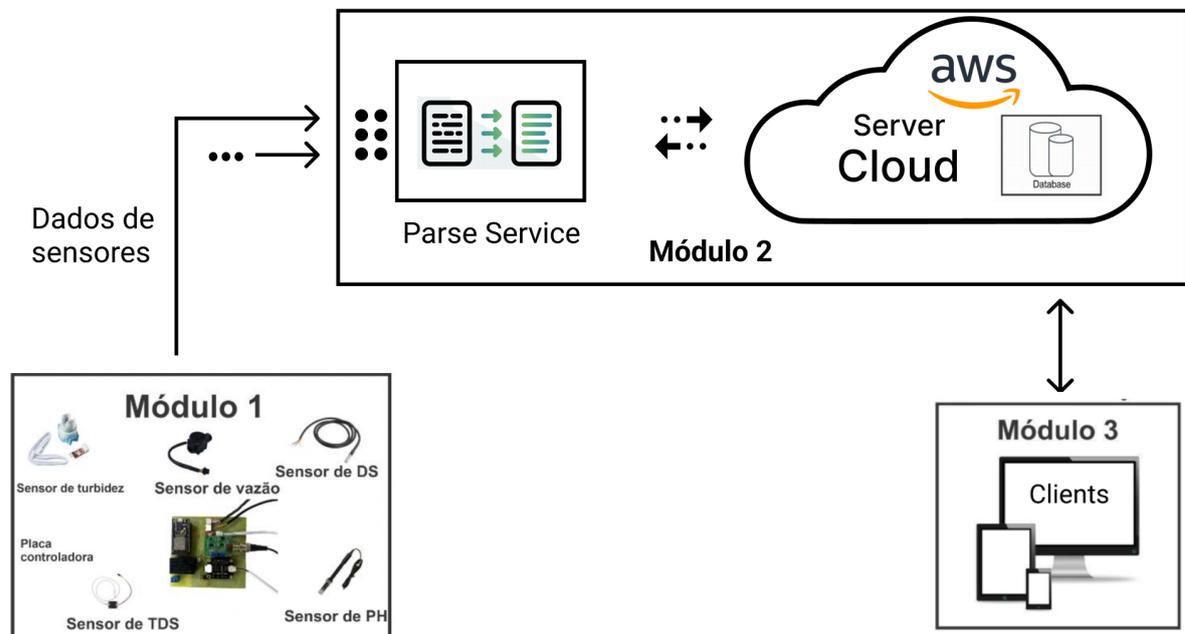
O Módulo 1 refere-se ao dispositivo de coleta de dados através de sensores referentes aos parâmetros físico-químicos da água. Este Módulo é capaz de integrar todos os sensores e se comunicar através de um protocolo de comunicação.

Já o Módulo 2 pode ser dividido em duas partes. A primeira é um serviço responsável por obter os dados de um dispositivo sensorial através de um protocolo de comunicação

específico do dispositivo e assim enviar à fila de processamento na nuvem por meio de um tópico. Vale pôr em evidência que o Parse Serve deve ser implementado a fim de corresponder ao Módulo 1, pois é a partir deste serviço que o sistema passa a ser generalizado em relação a todos os protocolos de comunicação. A segunda parte do Módulo 2 é onde fica hospedado o servidor da aplicação em nuvem sendo ele o responsável por processar e armazenar as amostras no banco de dados.

Por fim, o Módulo 3 é onde o usuário pode acessar todas as informações fornecidas pelas funcionalidades desenvolvidas para o sistema. Estas funcionalidades estão melhor descritas na seção de 5.

Figura 27 – Arquitetura geral do sistema.

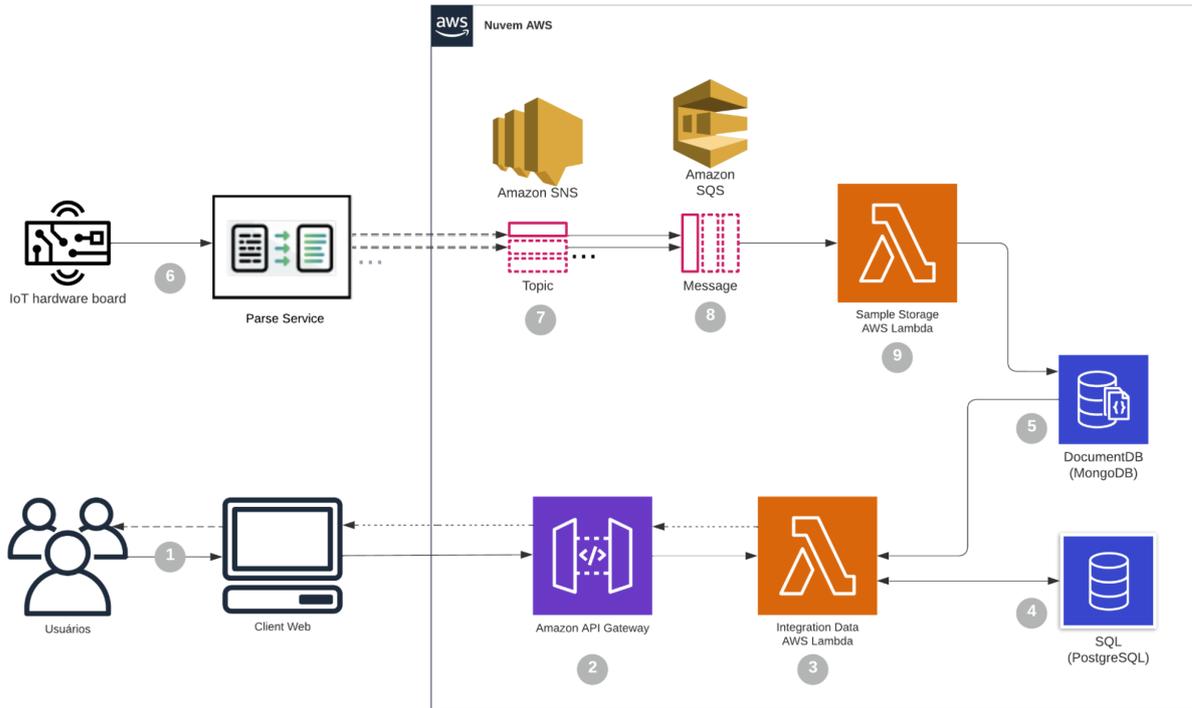


Fonte: (Elaborado pelo autor, 2022)

#### 7.4 Arquitetura em nuvem

Para suprir as demandas de infraestrutura do projeto, foi escolhido o serviço de computação em nuvem AWS com o intuito de atender as necessidades de comunicação e armazenamento de dados. A figura 28 demonstra o fluxo em que os usuários acessam um cliente Web o qual recebe dados de um *Application Programming Interface (API) Gateway* da AWS. Este *Gateway* é responsável por realizar requisições e espera respostas de uma *Lambda* (uma função que recebe uma entrada e devolve uma saída). Este *Lambda* tem acesso a dois banco de dados diferentes do sistema: o *PostgreSQL* o qual armazena dados relacionados a usuários e informações cadastrais dos PDM's; o *MongoDB* que armazena amostras coletadas dos PDM's.

Figura 28 – Arquitetura em nuvem do sistema.



Fonte: (Elaborado pelo autor, 2022)

É possível observar na figura acima que outra *Lambda* recebe dados de uma fila SQS e os armazena no banco de dados *MongoDB*. A fila SQS, por sua vez, recebe estes dados pela publicação do tópico SNS. O ator responsável por publicar estas amostras no tópico SNS é o *Parse Service*, o qual possui um *script* configurável para realizar a comunicação apropriada com o IoT *hardware board*. Estes dispositivos podem possuir diferentes formas de comunicação, portanto, o *Parse Service* é responsável por garantir que este requisitos dos dispositivos não seja um impeditivo à coleta de dados para o sistema de monitoramento.

## 7.5 Modelagem do banco de dados

Para obter uma demonstração de como serão estruturados os dados dentro do sistema proposto e para que a regra de negócio fique em operação da forma necessária, foi realizado o desenvolvimento da modelagem do banco de dados *PostgreSQL* (banco de dados relacional) e do *MongoDB* (banco de dados não relacional). Ambas as modelagens podem ser visualizadas na figura 29 e na figura 30.

### 7.5.1 Modelagem do banco relacional - PostgreSQL

No diagrama ilustrado na figura 29 demonstra a operação das regras de negócio para os usuários. Neste caso, tem-se os campos presentes para a descrição de cada usuário e sua regra de relacionamento com as tabelas de *Permissions* e *Roles*. Cada usuário terá um *role* no sistema, Administrador ou Analista. Cada *Role* possui as *Permissions* de usabilidade dentro do sistema.

Figura 29 – Diagrama de relacionamentos da tabela de usuário

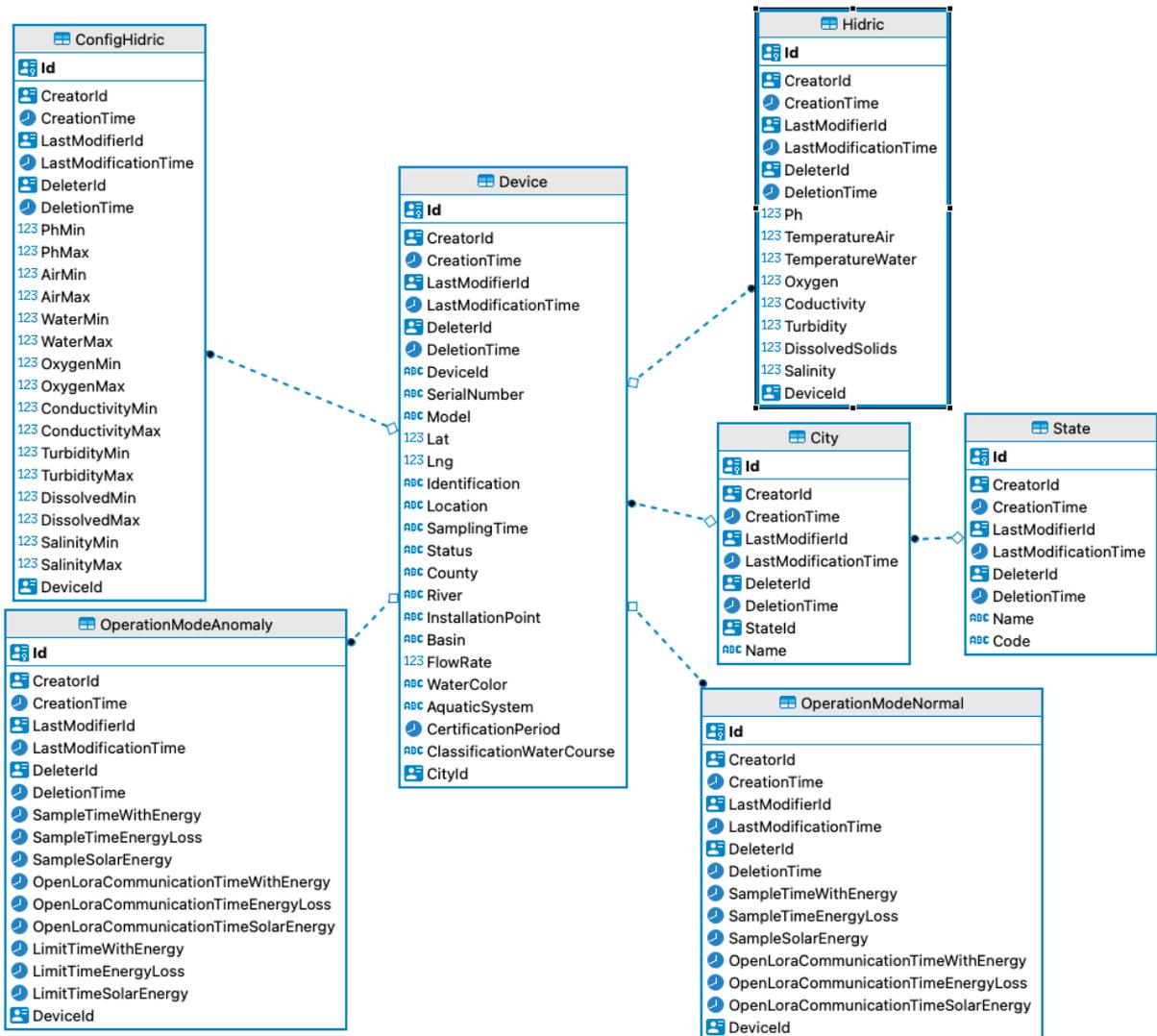


Fonte: (Elaborado pelo autor, 2022)

Na figura 30 é possível ver a tabela de *Devices* que representa o dispositivo no Posto de Monitoramento (PDM). Como é possível visualizar, o dispositivo terá relacionamento com *OperationModeNormal* que organizará os dados da taxa de amostragem para caso o dispositivo esteja conectado a energia da concessionária de energia, para caso o dispositivo tenha o seu suprimento de energia fornecido pela concessionária e tenha sido necessário o abastecimento por meio de uma bateria e para o caso de o dispositivo esteja conectado à energia solar. Também tem relacionamento com *OperationModeAnomaly* em que guarda os mesmos dados da tabela descrita anteriormente, mas para casos em que o dispositivo não está se comportando adequadamente. Além destas, a tabela *Device* também possui

relacionamento com a tabela *ConfigHidric* que guarda os valores de máximo e mínimo de cada parâmetro físico-químico da água.

Figura 30 – Diagrama de relacionamentos da tabela de dispositivo

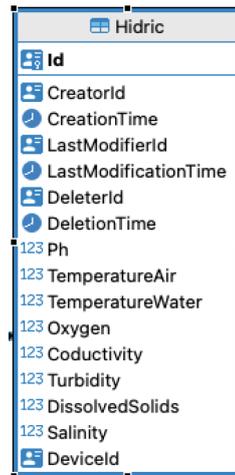


Fonte: (Elaborado pelo autor, 2022)

### 7.5.2 Modelagem do banco não-relacional - MongoDB

A figura 31 mostra o modelo de documento a ser salvo no banco de dados MongoDB. Como se trata de um banco de dados não-relacional, não se trata de uma tabela modelada, mas de documentos a serem salvos. Estes documentos são os registros dos parâmetros físico-químicos coletados da água, por um determinado Posto de Monitoramento (PDM) com uma data e hora específica.

Figura 31 – Diagrama de modelo do documento de amostragem



Fonte: (Elaborado pelo autor, 2022)

## 7.6 Ferramentas e tecnologias utilizadas

Nesta seção serão descritas as ferramentas e tecnologias utilizadas para o desenvolvimento, assim como sua aplicabilidade dentro do projeto.

### 7.6.1 VS Code - Visual Studio Code

A ferramenta Visual Studio Code, criada pela Microsoft, foi escolhida como editor de texto para a escrita do código-fonte do projeto. A figura 32 ilustra esta *Integrated Development Environment* (IDE) sendo utilizada para desenvolvimento de funcionalidades dentro do projeto. É importante ressaltar que, mesmo com diversas outras ferramentas de edição de texto disponíveis no mercado, o VS Code se destaca pela sua característica de ser customizável, o que possibilita configurá-la de modo a atender precisamente as necessidades de desenvolvimento deste projeto. Além disso, é possível abrir e utilizar mais de um terminal dentro da IDE, o que eleva os níveis de produtividade, pois permite que um terminal seja usado para executar os *scripts* criados e outro para a adição e remoção de pacotes.

Figura 32 – IDE utilizada no projeto

```

1 {
2   "name": "water-inspection-backend",
3   "version": "0.0.1",
4   "description": "NodeJS Backend for App and Website",
5   "main": "index.js",
6   "scripts": {
7     "build": "babel src --extensions \".js,.ts\" --out-dir dist --copy-files --no-copy-ignored",
8     "test-local": "cross-env NODE_ENV=test_local LOG_LEVEL=ERROR jest --runInBand",
9     "clear-jest": "jest --clearCache",
10    "dev": "cross-env NODE_ENV=local ts-node-dev -p tsconfig-paths/register --inspect=0.0.0.9229 --respawn --transpile-only --",
11    "start": "ts-node-dev -p tsconfig-paths/register --inspect --respawn --transpile-only --ignore-watch node_modules --no-notif",
12    "eslint": "eslint ./src/**/*.ts"
13  },
14  "repository": {
15    "type": "git",
16    "url": "git+https://gitlab.com/hub-lse-lab-rios-water-inspection/water-inspection-backend.git"
17  },
18  "author": "Isaque Vilson <isaquevilson@gmail.com >",
19  "license": "ISC",
20  "bugs": {
21    "url": "https://gitlab.com/hub-lse-lab-rios-water-inspection/water-inspection-backend/issues"
22  },
23  "homepage": "https://gitlab.com/hub-lse-lab-rios-water-inspection/water-inspection-backend#readme",
24  "devDependencies": {
25    "@automapper/types": "^6.3.1",
26    "@types/bcrypt": "^5.0.0",
27    "@types/bcryptjs": "^2.4.2",
28    "@types/cons": "^2.8.12",
29    "@types/express": "^4.17.13",
30    "@types/jest": "^27.4.0",
31    "@types/jsonwebtoken": "^8.5.8",
32    "@types/safe-json-stringify": "^1.1.2",
33    "@types/supertest": "^2.0.12",
34    "@typescript-eslint/eslint-plugin": "^3.3.0",
35    "@typescript-eslint/parser": "^3.3.0",
36    "cross-env": "^7.0.3",
37    "eslint": "^7.3.0",
38    "eslint-config-prettier": "^6.11.0",
39    "eslint-config-standard": "^14.1.1",
40    "eslint-plugin-import": "^2.21.2",
41    "eslint-plugin-node": "^11.1.0",
42    "eslint-plugin-prettier": "^3.1.4",
43    "eslint-plugin-promise": "^4.2.1"
44  }
45 }

```

Fonte: (Elaborado pelo autor, 2022)

### 7.6.2 ESLint

O ESLint foi utilizado como ferramenta para análise de código, visto que possui extensão de mesmo nome disponível para uso no VSCode. Ela permite identificar erros quanto ao padrão de escrita de código definido. Como é possível visualizar na imagem 33, um determinado código do projeto foi escrito fora dos padrões estabelecidos onde existia espaçamentos indevidos, uso incorreto de ponto e vírgula e linha excedendo o máximo determinado. Após serem feitas as devidas correções, o ESLint indica que agora o código está adequado. A importância de se ter o código padronizado reside na prevenção de problemas de compilação simples e legibilidade do código.

Figura 33 – Aplicação do ESLint para correção de código

```

34 private middlewarees(): void {
35   const server = this.express;
36   server
37     .setConfig((app) => {
38       app.use(bodyParser.urlencoded({
39         extended: true
40       }));
41     })
42     .use(cors());
43   app.use(cors());
44   app.use(bodyParser.json());
45   app.use(express.static("public"));
46   .setErrorConfig((app)
47     => {
48       app.use((_err: Error, _request: Request, _response: Response)
49         => {
50         if (_err instanceof AppError) {
51           return _response.status(_err.statusCode).json({
52             message: _err.message,
53             status: _err.statusCode
54           });
55         }
56       });
57     });
58   });
59 }

```

Fonte: (Elaborado pelo autor, 2022)

Na figura 34 estão as configurações necessárias para que o ESLint possa sinalizar onde o código não atendeu a padronização estabelecido para o projeto.

Figura 34 – Configuração de padrão de código ESLint

```

.eslintrc.json — water-inspection-backend
.eslintrc.json > ...
1 {
2   "env": {
3     "es2020": true,
4     "node": true,
5     "jest": true
6   },
7   "extends": ["standard", "plugin:@typescript-eslint/recommended"],
8   "parser": "@typescript-eslint/parser",
9   "overrides": [
10    {
11      "files": ["*.ts"],
12      "parserOptions": {
13        "project": ["/tsconfig.json"]
14      }
15    }
16  ],
17  "parserOptions": {
18    "ecmaVersion": 11,
19    "sourceType": "module"
20  },
21  "plugins": ["@typescript-eslint"],
22  "rules": {
23    "@typescript-eslint/no-unused-vars": "off",
24    "@typescript-eslint/no-unused-vars-experimental": "error",
25    "@typescript-eslint/explicit-module-boundary-types": "off",
26    "no-unused-vars": "off",
27    "@typescript-eslint/no-floating-promises": "error",
28    "space-before-function-paren": "off"
29  }
30 }
31

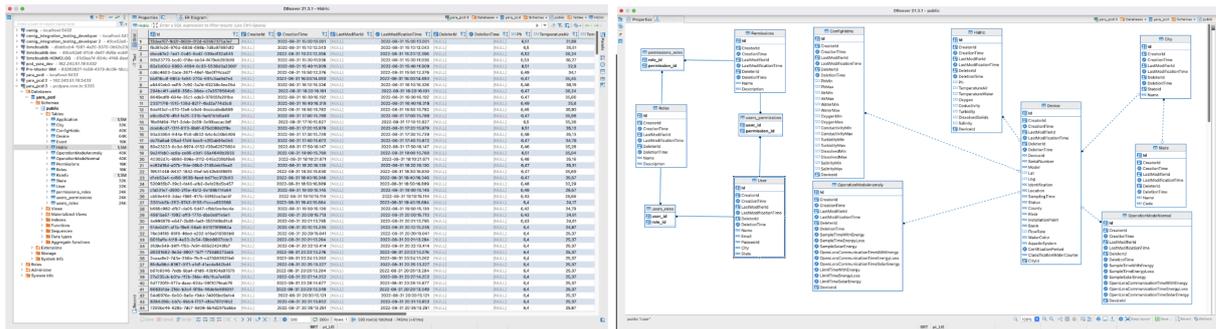
```

Fonte: (Elaborado pelo autor, 2022)

### 7.6.3 Dbeaver

O DBeaver foi utilizado como gestão de banco de dados. Esta ferramenta tem suporte para diferentes bases de dados e, como o este trabalho usa dois tipos: PostgreSQL e MongoDB, foi possível gerí-los facilmente. Além disso, conforme demonstrado na figura 35, foi possível obter o digrama da modelagem planejada automaticamente e realizar *debug* com a entrada de dados por meio da plataforma web desenvolvida.

Figura 35 – Utilização do Dbeaver no projeto

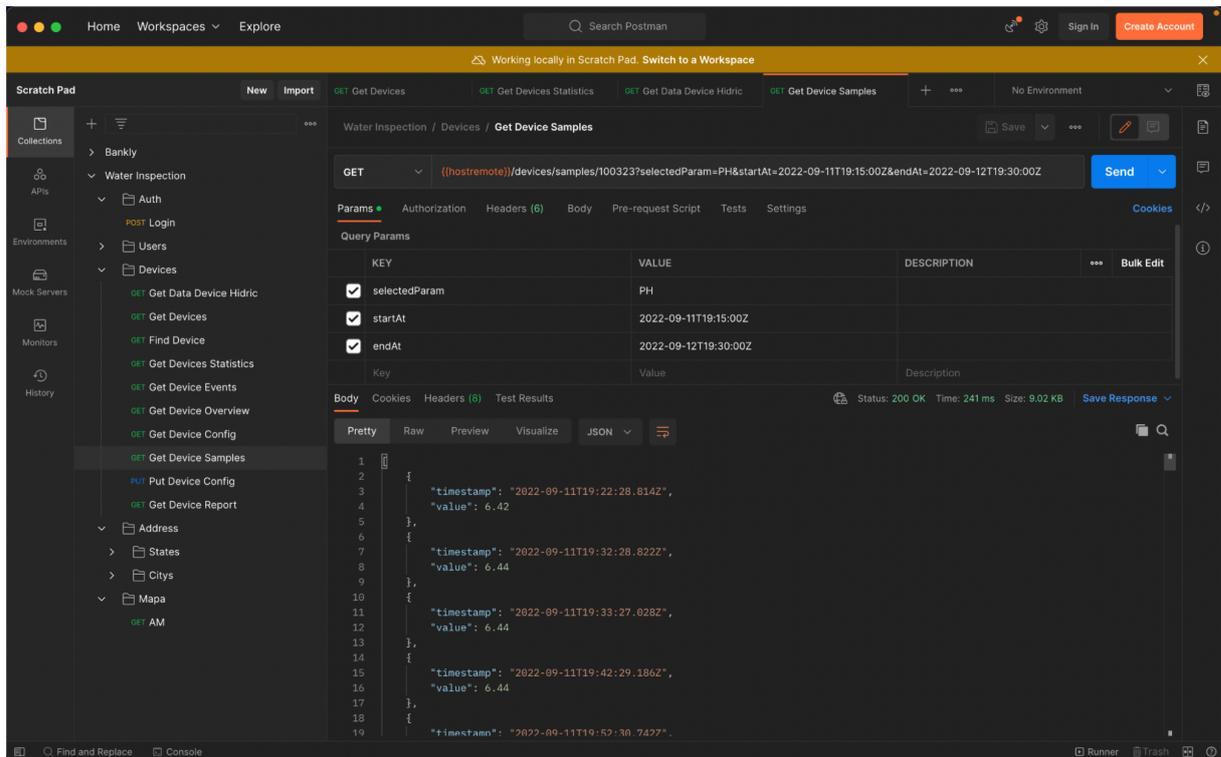


Fonte: (Elaborado pelo autor, 2022)

### 7.6.4 Postman

Ao realizar o desenvolvimento de um servidor *Web*, é necessário executar testes nas rotas criadas, para garantir a performance correta das funcionalidades antes de realizar a integração com o *frontend*, evitando prováveis perdas de tempo. Na figura 36 mostra o resultado da rota de obtenção de dados de parâmetros físico-químicos de um dispositivo.

Figura 36 – Teste de rotas com Postman

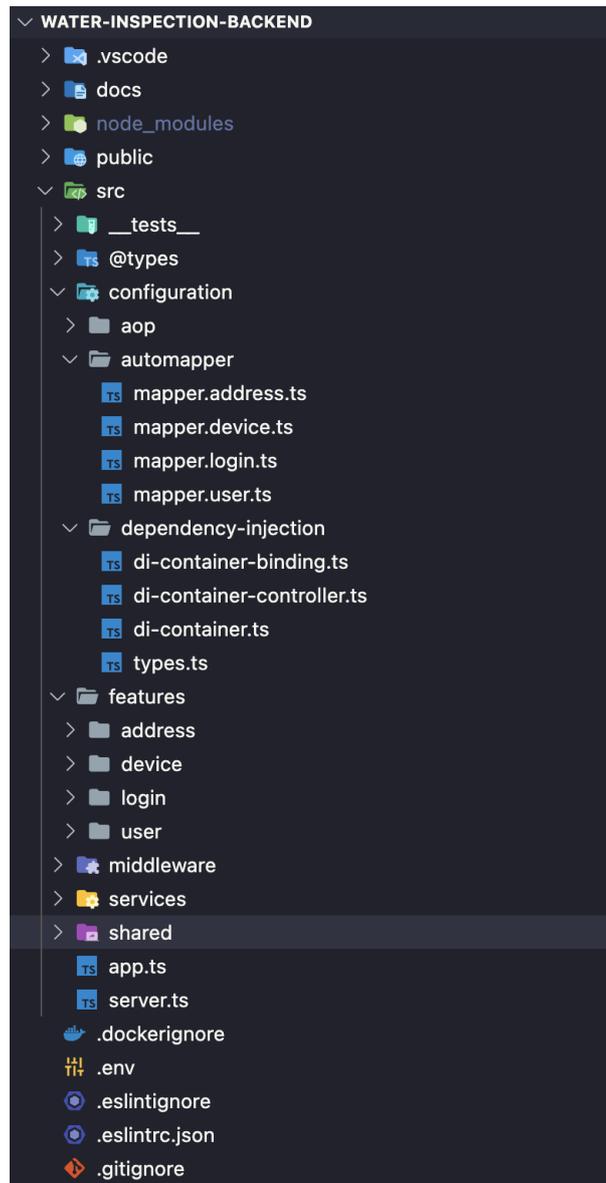


Fonte: (Elaborado pelo autor, 2022)

### 7.6.5 *TypeScript*

Foi utilizado o *TypeScript*, pois este entende a linguagem de programação *JavaScript* e usa inferência de tipos para fornecer ótimas ferramentas sem código adicional. Esta ferramenta é conhecida como superconjunto *JavaScript*, não sendo considerada uma linguagem de programação. Para o desenvolvimento deste trabalho, o *TypeScript* foi usado tanto do lado do cliente onde há a interação com o usuário em um navegador, quanto do lado do servidor onde está construída a regra de negócio e comunicações externas do projeto. Como é possível visualizar na figura 37 os arquivos de código do projeto possuem a extensão `.ts` que indica os arquivos escritos estão em *Typescript* e devem ser compilados de forma específica.

Figura 37 – Arquivos do backend em *TypeScript*



Fonte: (Elaborado pelo autor, 2022)

## 7.7 Estrutura e padrão de projeto utilizado

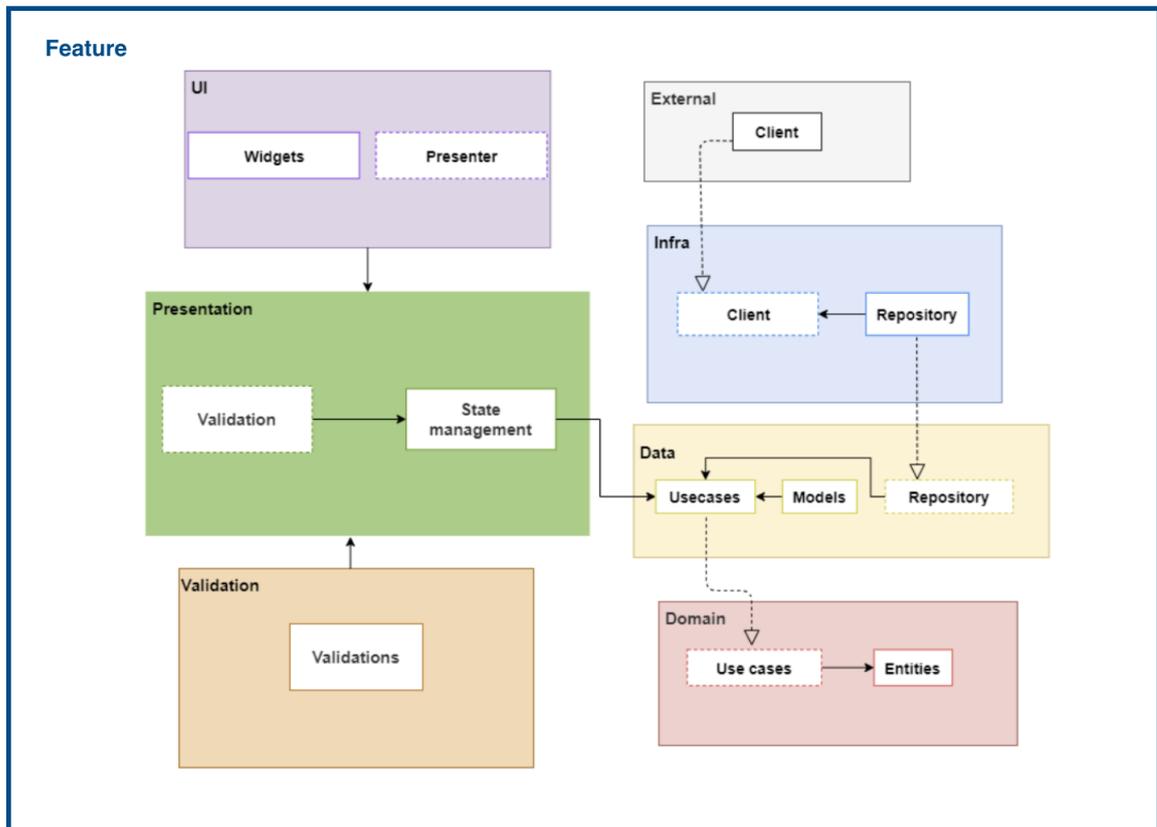
O projeto *Web* está dividido em duas partes principais: o *Backend*, onde consta a regra de negócio da aplicação, e o *Frontend*, onde é desenvolvida a comunicação com o usuário. Estes dois serão melhor descritos nas seções 7.7.1 e .

### 7.7.1 *Backend*

Para um projeto de *Software*, o *Backend* consiste na porção que associa o servidor com banco de dados, infraestrutura, gerenciamento de conteúdo e atualizações. No contexto da aplicação desenvolvida neste trabalho, cada sub-porção do *Backend* pôde ser colocada em módulos. Esta arquitetura é denominada de *Feature* que segue os princípios do SOLID em

conjunto com o *Clean Architecture*, como exposto na Figura 38. Uma *Feature* é composta por 6 camadas, sendo elas: *Domain*, *Data*, *Infra*, *UI*, *Presentation* e *Validation*.

Figura 38 – Arquitetura de serviço do *Backend*



Fonte: (Elaborado pelo autor, 2022)

Na figura 38 é possível observar as seis camadas de uma *Feature*. A camada *Domain* é onde foi definidas as entidades utilizadas pelos casos de uso e seus contratos. Esta camada está diretamente associada a camada de Entidades.

A camada de *Data* contém a implementação dos casos de uso e os adaptadores que transformam dados de formatos externos da aplicação para formatos compatíveis com as entidades. Além disso, esta camada teve a função de implementar interfaces e contratos para repositórios. Vale pôr em evidência que a camada *Data* se relaciona com a camada *Use Cases* (camada de casos de uso), ou seja, são regras de negócio da aplicação e adaptadores de interface.

Na camada de *Infra* foram implementados os contratos de repositórios definidos na camada *Data*. É onde bibliotecas externas foram utilizadas para realizar tarefas como requisições web e persistência de dados. Esta camada está relacionada com o último nível das camadas (*Frameworks* e *Drivers*).

Também é possível visualizar na figura 38, a camada UI abreviação para *User Interface*), cuja responsabilidade consistiu em implementar as rotas e também definir contratos dos *Presenters*. No modelo arquitetural *Clean Architecture* pode-se relacionar a UI à camada de *Frameworks* e *Drivers*.

A camada *Presentation* realizou a conexão entre a camada de dados (*Data*) e a camada de interface de usuário (UI). Os dados foram providos dos casos de uso para a interface e da interface para os casos de uso. Além disso, também realizou a definição de contratos para a validação de dados.

Por fim, a camada de *Validation* validou informações inseridas na camada de *Presentation*, como e-mails, formato de números de geolocalização, etc.

A figura 39 abaixo evidencia como a estrutura de pastas descrita fora implementada no projeto *Water Inspection*.

Figura 39 – Estrutura de pasta desenvolvida no *Backend*.

```

41 import { DeviceRepositoryImpl } from '@/features/device/infra/repository/device.repository.impl'
42 import { GetStatesUseCaseImpl } from '@/features/address/data/use-cases/get-states.use-case.impl'
43 import { GetCitiesByStateUseCaseImpl } from '@/features/address/data/use-cases/get-cities-by-state.use-case.impl'
44
45 export function bindRepositories(DIContainer: Container) {
46   DIContainer.bind<UserRepository>(TYPES.UserRepository).to(UserRepositoryImpl)
47   DIContainer.bind<AddressRepository>(TYPES.AddressRepository).to(AddressRepositoryImpl)
48 }
49
50 export function bindServices(DIContainer: Container) {
51   DIContainer.bind<LogService>(TYPES.LogService).to(LogServiceImpl)
52   DIContainer.bind<RepositoryService>(TYPES.RepositoryService).to(RepositoryServiceImpl)
53 }
54
55 export function bindUseCases(DIContainer: Container) {
56   DIContainer.bind<CreateUserUseCase>(TYPES.CreateUserUseCase).to(CreateUserUseCaseImpl)
57   DIContainer.bind<UpdateUserUseCase>(TYPES.UpdateUserUseCase).to(UpdateUserUseCaseImpl)
58   DIContainer.bind<DeleteUserUseCase>(TYPES.DeleteUserUseCase).to(DeleteUserUseCaseImpl)
59   DIContainer.bind<GetUserUseCase>(TYPES.GetUserUseCase).to(GetUserUseCaseImpl)
60   DIContainer.bind<LoginUseCase>(TYPES.LoginUseCase).to(LoginUseCaseImpl)
61   DIContainer.bind<GetDataDeviceHidricUseCase>(TYPES.GetDataDeviceHidricUseCase).to(GetDataDeviceHidricUseCaseImpl)
62   DIContainer.bind<GetDevicesUseCase>(TYPES.GetDevicesUseCase).to(GetDevicesUseCaseImpl)
63   DIContainer.bind<FindDeviceUseCase>(TYPES.FindDeviceUseCase).to(FindDeviceUseCaseImpl)
64   DIContainer.bind<DeviceRepository>(TYPES.DeviceRepository).to(DeviceRepositoryImpl)
65   DIContainer.bind<GetDevicesStatisticsUseCase>(TYPES.GetDevicesStatisticsUseCase).to(GetDevicesStatisticsUseCaseImpl)
66   DIContainer.bind<GetDeviceEventsUseCase>(TYPES.GetDeviceEventsUseCase).to(GetDeviceEventsUseCaseImpl)
67   DIContainer.bind<GetDeviceOverviewUseCase>(TYPES.GetDeviceOverviewUseCase).to(GetDeviceOverviewUseCaseImpl)
68   DIContainer.bind<GetDeviceConfigUseCase>(TYPES.GetDeviceConfigUseCase).to(GetDeviceConfigUseCaseImpl)
69   DIContainer.bind<GetDeviceSamplesUseCase>(TYPES.GetDeviceSamplesUseCase).to(GetDeviceSamplesUseCaseImpl)
70   DIContainer.bind<GetStatesUseCase>(TYPES.GetStatesUseCase).to(GetStatesUseCaseImpl)
71   DIContainer.bind<GetCitiesByStateUseCase>(TYPES.GetCitiesByStateUseCase).to(GetCitiesByStateUseCaseImpl)
72 }
73
74 export function bindMiddleware(DIContainer: Container) {
75   DIContainer.bind<AuthMiddleware>(TYPES.AuthMiddleware).to(AuthMiddleware)
76 }
77

```

Fonte: (Elaborado pelo autor, 2022)

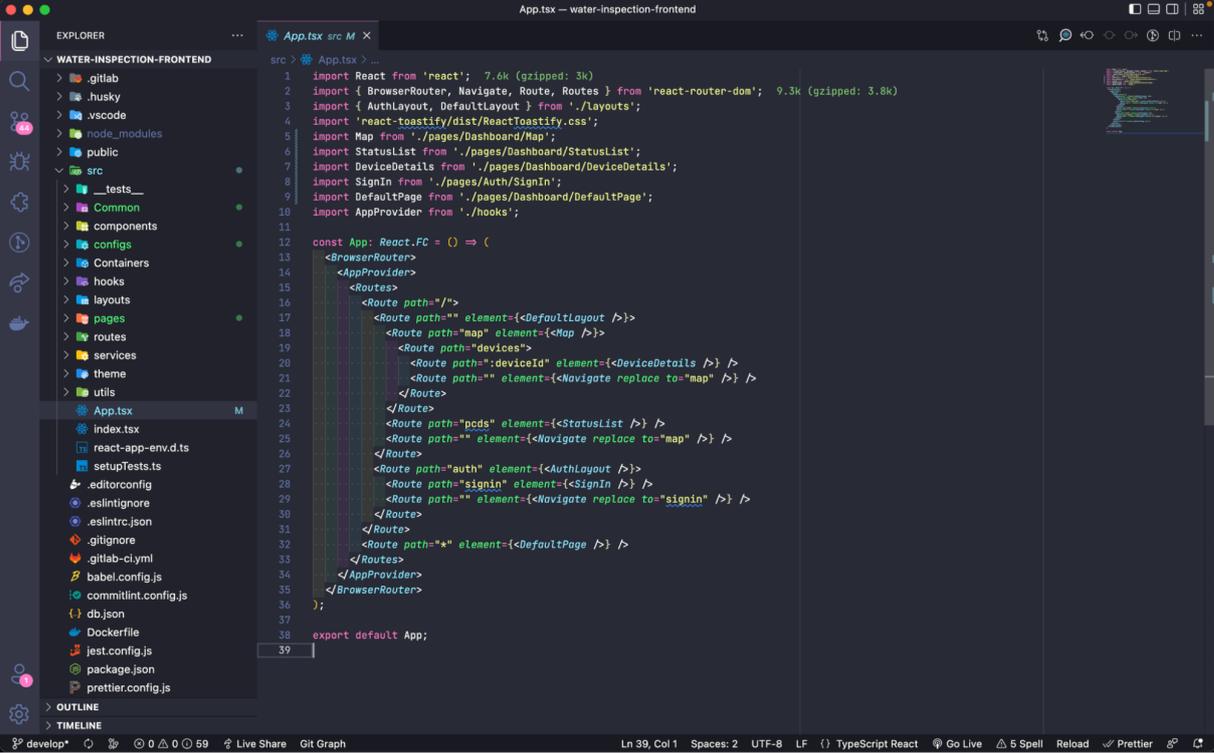
### 7.7.2 Frontend

Como em todo projeto de *Software*, o *Frontend* consistiu na porção que implementa a interface gráfica do sistema. No desenvolvimento deste trabalho a arquitetura organizacional de pastas do projeto do *Frontend* foi pensada para proporcionar escalabilidade

e boa compreensão para possíveis manutenções no código. Estes diretórios são separados em: *Common*, *Components*, *Configs*, *Containers*, *Hooks*, *Pages*, *Routes* e *Services*.

- ***Common***: contém todos os assets do projeto os quais são utilizados pelo aplicativo, como estilos globais, imagens, fontes, mocks, stories e funções reutilizáveis.
- ***Components***: responsável por todos os componentes utilizados de forma global na aplicação desenvolvida. Os componentes utilizados apenas em uma página ficam em uma pasta *components* dentro do diretório de cada página em *Pages*.
- ***Configs***: abrange todos os arquivos de configuração utilizados de forma global pela aplicação.
- ***Containers***: responsável por desacoplar a aplicação de bibliotecas, possibilitando alterá-las sem precisar realizar várias mudanças em lugares diferentes.
- ***Hooks***: é onde ficam todos os *hooks* customizáveis da aplicação, como o *hook* que cuida da sessão de usuário.
- ***Pages***: composto por pastas contendo os arquivos de uma página da aplicação. Dentro da pasta de uma página há uma outra chamada *components* contendo todos os componentes exclusivos da página.
- ***Route***: encontra-se os arquivos de rota da aplicação.
- ***Services***: contém os arquivos de serviços externos da aplicação.

É possível visualizar a implementação destes diretórios no desenvolvimento do *Frontend* na figura ??.

Figura 40 – Estrutura de pasta desenvolvida no *Frontend*.

The image shows a VS Code editor window with the Explorer sidebar on the left and the main editor area on the right. The Explorer sidebar shows the file structure of a project named 'WATER-INSPECTION-FRONTEND'. The main editor area displays the content of the 'App.tsx' file, which is a React application component. The code includes imports for various libraries and components, and a main function that renders the application using a Router and a Provider. The code is as follows:

```
1 import React from 'react'; // 7.4k (gzipped: 3k)
2 import { BrowserRouter, Navigate, Route, Routes } from 'react-router-dom'; // 9.3k (gzipped: 3.8k)
3 import { AuthLayout, DefaultLayout } from './layouts';
4 import 'react-toastify/dist/ReactToastify.css';
5 import Map from './pages/Dashboard/Map';
6 import StatusList from './pages/Dashboard/StatusList';
7 import DeviceDetails from './pages/Dashboard/DeviceDetails';
8 import SignIn from './pages/Auth/SignIn';
9 import DefaultPage from './pages/Dashboard/DefaultPage';
10 import AppProvider from './hooks';
11
12 const App: React.FC = () => {
13   <BrowserRouter>
14     <AppProvider>
15       <Routes>
16         <Route path="/">
17           <Route path="" element={<DefaultLayout />}>
18             <Route path="map" element={<Map />}>
19               <Route path="devices">
20                 <Route path="deviceId" element={<DeviceDetails />} />
21                 <Route path="" element={<Navigate replace to="map" />} />
22               </Route>
23             </Route>
24           <Route path="peds" element={<StatusList />} />
25           <Route path="" element={<Navigate replace to="map" />} />
26         </Route>
27         <Route path="auth" element={<AuthLayout />}>
28           <Route path="signin" element={<SignIn />} />
29           <Route path="" element={<Navigate replace to="signin" />} />
30         </Route>
31         <Route path="*" element={<DefaultPage />} />
32       </Routes>
33     </AppProvider>
34   </BrowserRouter>
35 );
36
37 export default App;
```

Fonte: (Elaborado pelo autor, 2022)

## 8 RESULTADOS

Em decorrência do capítulo anterior serão expostos os resultados obtidos a partir do *Backend* e *Frontend*, bem como suas respectivas validações. Existem algumas formas de verificar as implementações realizadas no *backend*, uma delas é consumindo os *endpoints* via *HTTP Client* (Postman). Outra maneira consiste em acessar as páginas Web da aplicação desenvolvida e analisar a forma como os dados estão retornando da API.

### 8.1 Validação da plataforma desenvolvida com um dispositivo para coleta de dados de parâmetros físico-químicos da água.

Para a validação do sistema desenvolvido, conforme a figura 41, utilizou-se um dispositivo ligado aos sensores para o envio dos parâmetros físico-químicos da água via MQTT. O teste foi feito no Laboratório de Sistemas Embarcados (LSE) do Hub - Instituto de Tecnologia e Inovação localizado na Escola Superior de Tecnologia da UEA. A amostra de água utilizada foi coletada em uma torneira abastecida pelo sistema hídrico da instituição.

Figura 41 – Recebimento dos dados de parâmetros físico-químicos da água através de um dispositivo.



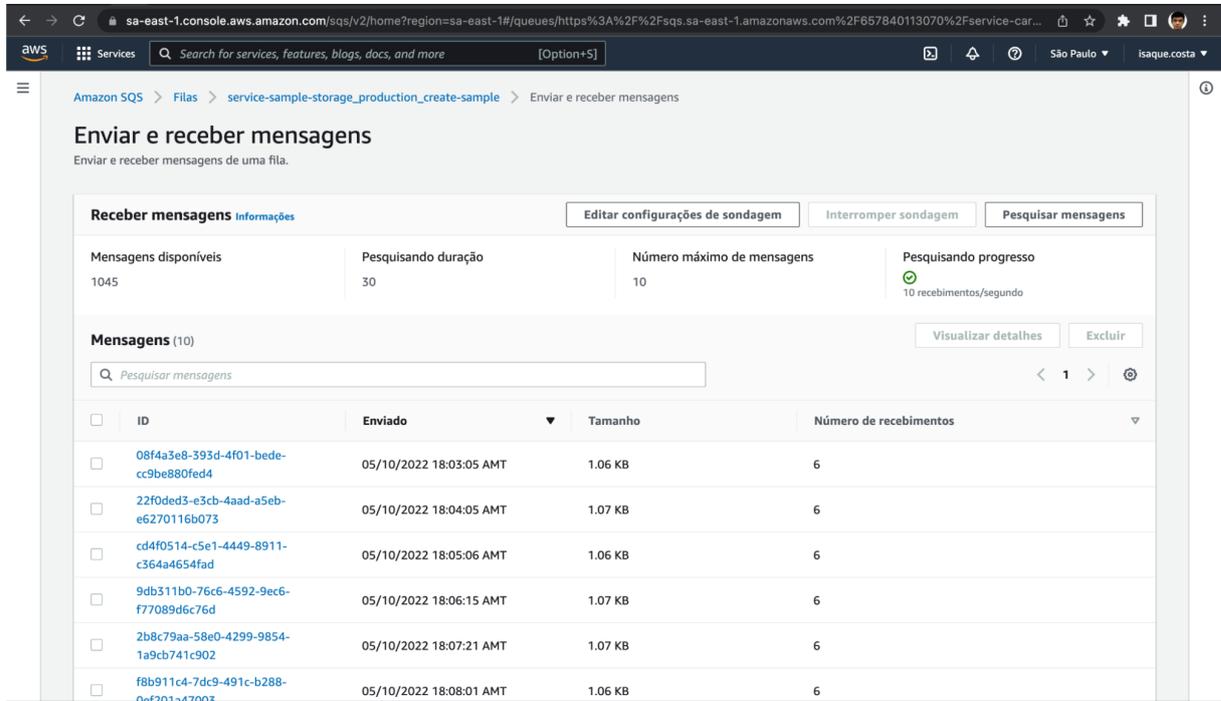
Fonte: (Elaborado pelo autor, 2022)

Vale pôr em evidência que o sistema desenvolvido neste trabalho suportaria o envio de dados de qualquer outro dispositivo, desde que fosse implementado um *script* de envio das amostras ao tópico SNS da AWS. O dispositivo utilizado para a validação do sistema foi desenvolvido no LSE para um outro projeto de pesquisa. O experimento feito teve a finalidade de testar a plataforma desenvolvida com o envio de dados reais.

Conforme a 42, mensagens com dados de parâmetros físico-químicos da água foram publicadas no tópico SNS e recebidos na fila SQS da AWS para processamento. Nessa

figura, podemos verificar que temos 1045 amostras disponíveis para serem processadas pelo serviço de armazenamento do *backend*.

Figura 42 – Mensagens com amostras de dados na fila SQS da AWS

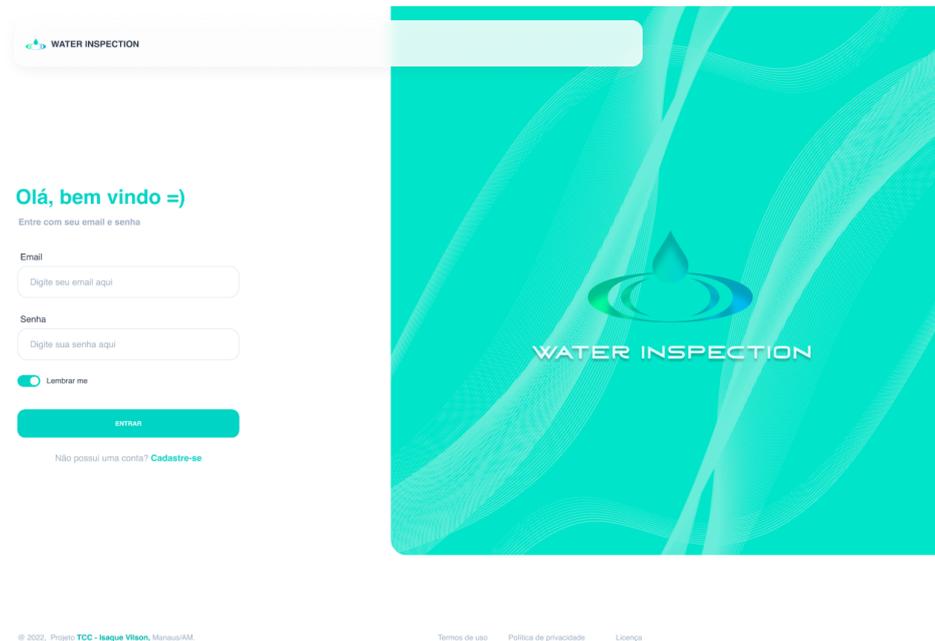


Fonte: (Elaborado pelo autor, 2022)

Nas próximas seções serão apresentados os resultados obtidos no software supervisorio desenvolvido.

## 8.2 Página de autenticação

A figura 43 exibe a funcionalidade de *Login* do projeto. O usuário, analista ou administrador, insere as informações previamente cadastradas de email e senha. A partir desta ação, o sistema identifica as permissões do usuário possibilitando que este acesse o sistema apropriadamente.

Figura 43 – Página de login da plataforma *Water Inspection*

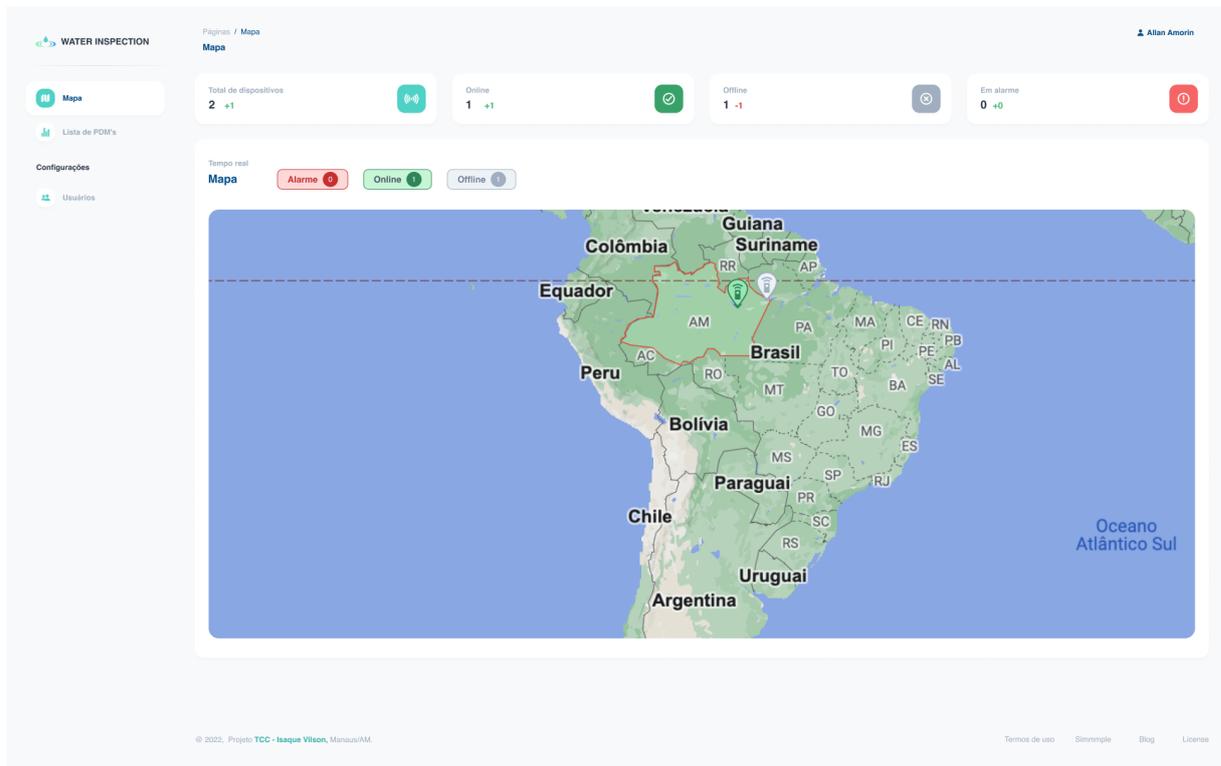
Fonte: (Elaborado pelo autor, 2022)

### 8.3 Página de mapa com os postos de monitoramento

A página de mapa com os postos de monitoramento na figura 44 executa o que foi previsto no caso de uso da tabela 17. Nesta tela mostra o funcionamento da identificação do Posto de Monitoramento e sua localidade no mapa, mais especificamente na unidade da Escola Superior de Tecnologia da UEA. Como é possível perceber, o Posto de Monitoramento (PDM) está online e não foi gerado nenhum alarme até o momento do recolhimento de dados para a escrita deste trabalho, podendo concluir que este PDM está funcionando adequadamente.

Também é possível perceber que existe outro PDM cadastrado nesta plataforma, no entanto, está offline. Este cadastro foi feito apenas com a finalidade de demonstrar que há a possibilidade de existir mais de um PDM na plataforma.

Figura 44 – Página de mapa com os postos de monitoramento em tempo real



Fonte: (Elaborado pelo autor, 2022)

#### 8.4 Página de listagem de postos de monitoramento

A página de listagem de PDM's, apresentada na figura 45, possui a mesma função descrita na Página de mapa com postos de monitoramento, no entanto esta é uma alternativa para visualizar os dados caso não seja possível carregar o mapa, ou ainda para uma visualização mais concisa dos dados.

No caso demonstrado abaixo há dois PDM cadastrados, um em Parintins e outro em Manaus. O posto de monitoramento de Parintins foi cadastrado apenas como caso demonstrativo e não estava online no momento da captura de dados para a elaboração deste trabalho.

Já o posto de monitoramento localizado em Manaus foi utilizado para os testes na Escola Superior de Tecnologia da UEA. Também é possível visualizar que há o indicativo de contagem dos dispositivos que emitiram alarme, online e offline. Também é importante ressaltar que esta funcionalidade recebe dados em tempo real do servidor para a precisão dos dados a serem analisados.

Figura 45 – Página de listagem de posto de monitoramento

WATER INSPECTION

Página / Lista de PDM's

Lista de status de PDM

Tempo real

PDMs

Alarma 0

Online 44

Offline 0

(+30) há uma hora atrás

Código	Município / UF	Bacia	Rio	Ponto de instalação	Status
AA23344KJ	Parintins / AM	Amazonas	Amazonas	Porto de Parintins	Offline
AA23344KJ	Manaus / AM	Amazonas	Rio Negro	Comunidade Nossa senhora de fátima	Online

1 >

© 2022, Projeto TCC - Isaque Vilson, Manaus/AM

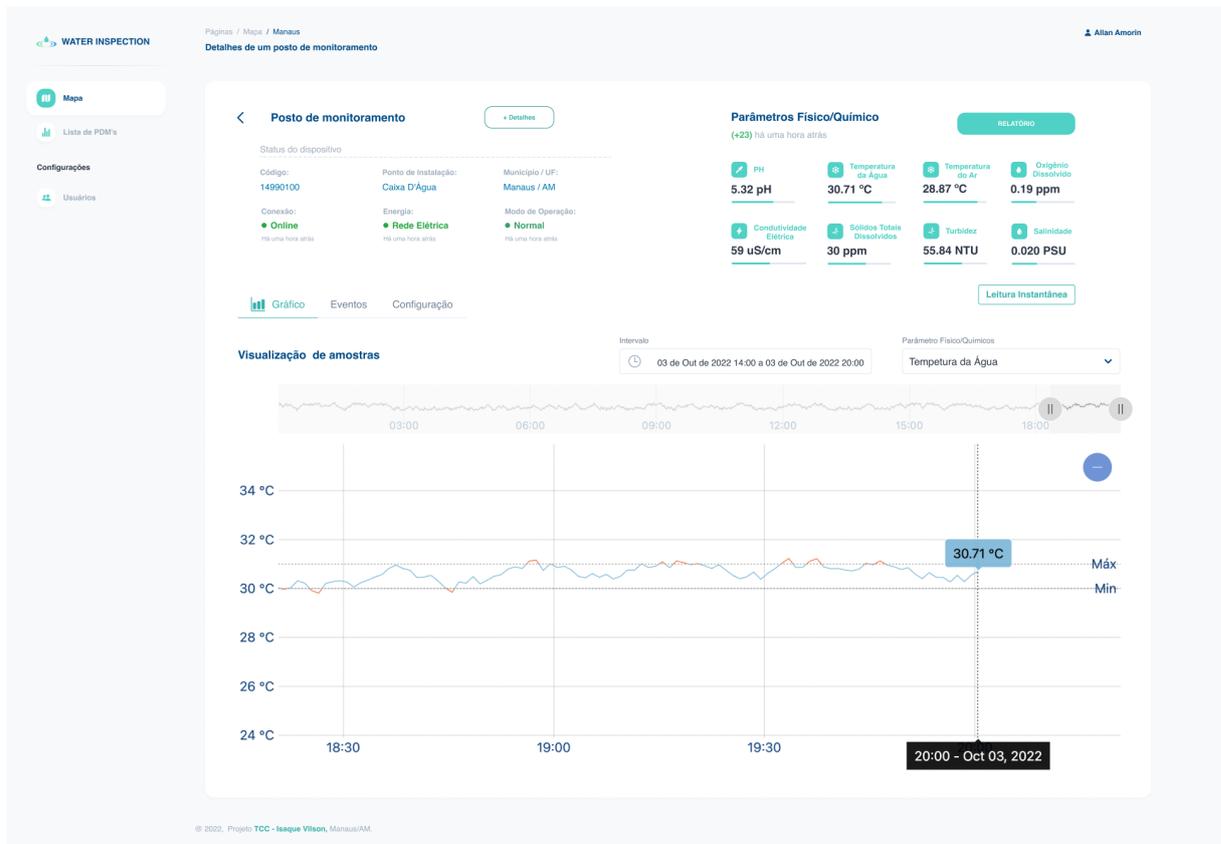
Termos de uso | Sobre | Blog | License

Fonte: (Elaborado pelo autor, 2022)

## 8.5 Página de detalhes de um posto de monitoramento

A figura 47 apresenta a página de detalhes de um posto de monitoramento. Nela é possível visualizar o histórico de eventos ocorridos na aplicação, o gráfico de amostras coletadas dos parâmetros em um determinado espaço de tempo, ir para as configurações de um PDM, e visualizar os parâmetros coletados mais atualizados.

Figura 46 – Página de detalhes de um posto de monitoramento



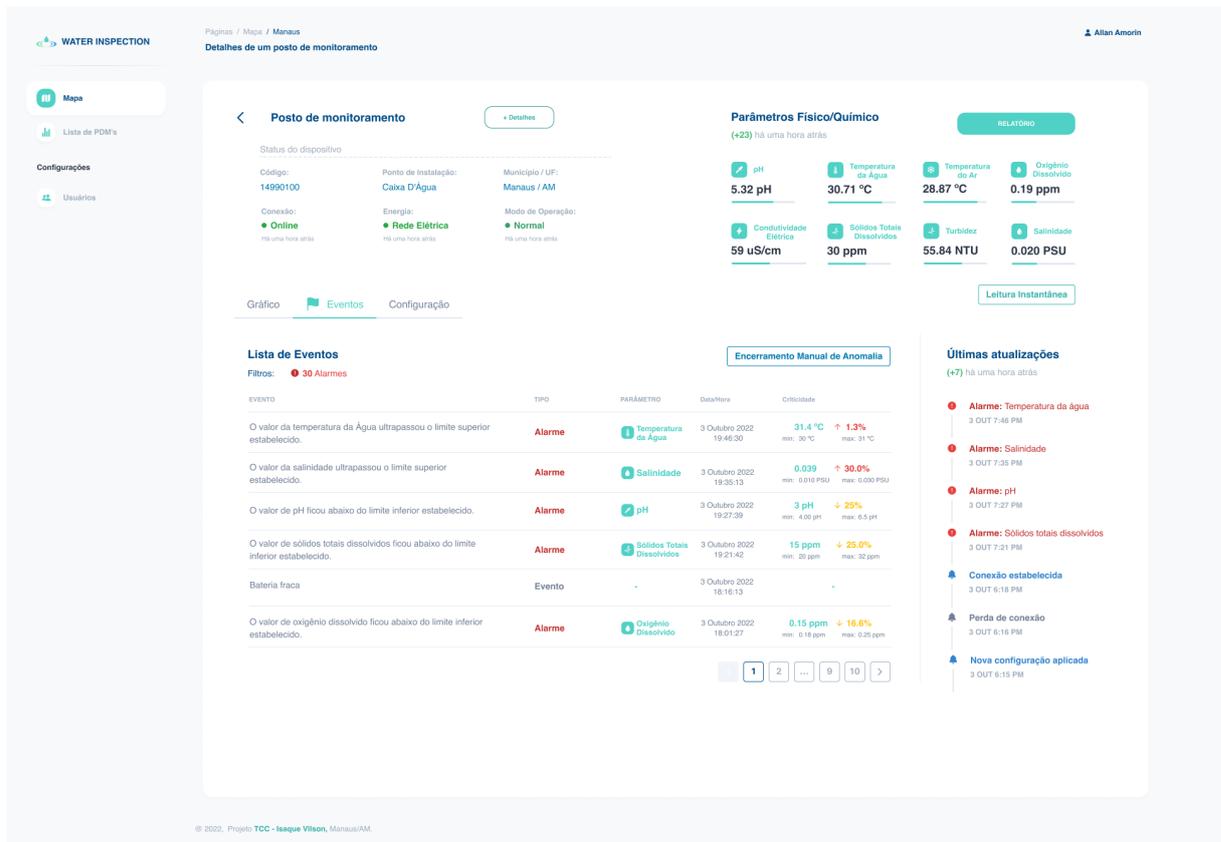
Fonte: (Elaborado pelo autor, 2022)

## 8.6 Página de eventos e alarmes do posto de monitoramento

Na página de eventos e alarmes, demonstrada na figura 47, tem a funcionalidade de exibir ao usuário informações do posto de monitoramento em tempo real. Os alarmes são gerados quando há variância fora dos limites máximo e mínimo previamente estabelecidos no sistema. Já os eventos são emitidos pelo dispositivo do posto de monitoramento e indica a ocorrência de uma mudança, podendo ser bateria fraca, sensor desconectado, erro na leitura de um sensor, perda de conexão, entre outros.

É importante ressaltar que os eventos são apenas mensagens captadas pelo sistema, não sendo previamente estabelecida a descrição de todos os eventos possíveis.

Figura 47 – Página de eventos e alarmes de um posto de monitoramento

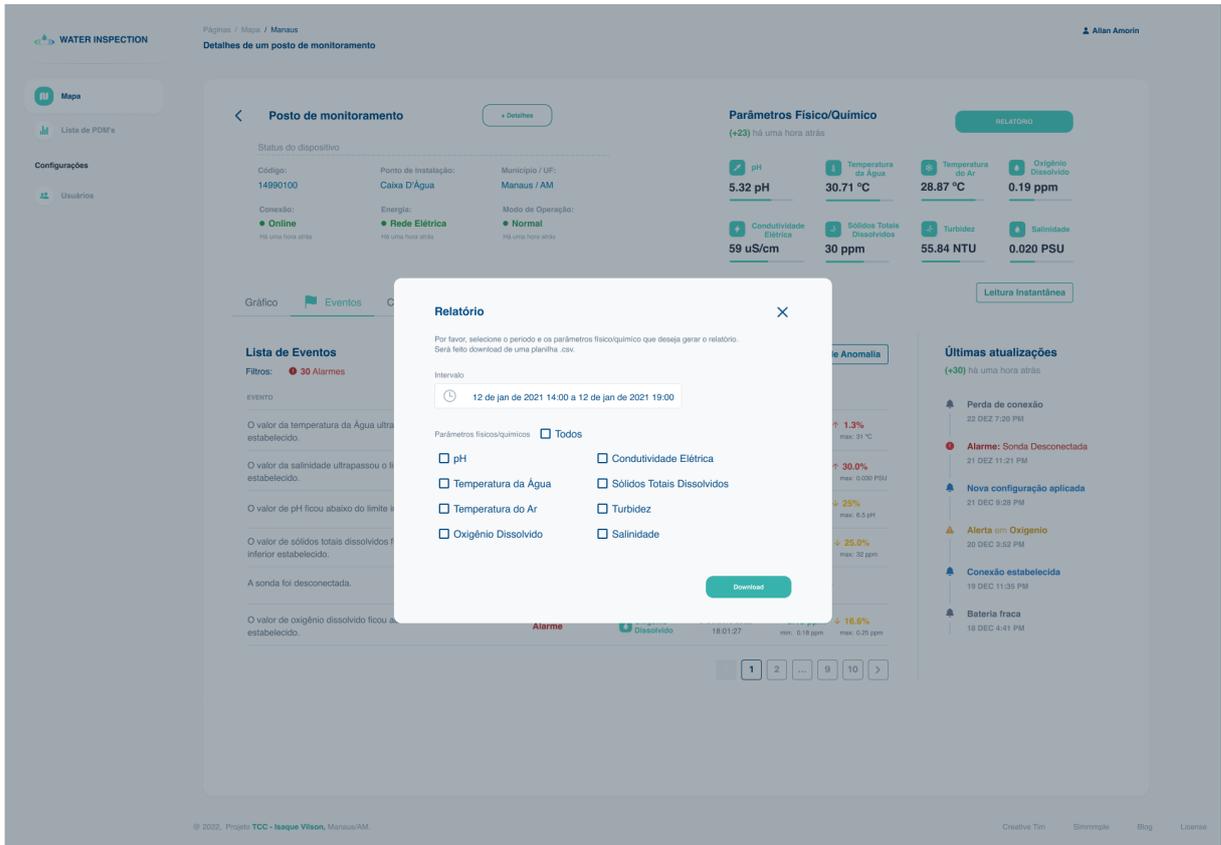


Fonte: (Elaborado pelo autor, 2022)

## 8.7 Página de geração de relatório

A figura 48 apresenta a página de geração de relatório para atender a necessidade do usuário de gerar uma planilha em formato .csv com todos os dados de parâmetros físico-químicos coletados. O relatório é gerado com os parâmetros selecionados na interface e com a possibilidade de delimitar um período de tempo. É possível visualizar a planilha gerada na figura 49.

Figura 48 – Página de geração de relatório



Fonte: (Elaborado pelo autor, 2022)

Figura 49 – Planilha gerado pela plataforma

	A	B	C	D	E	F	G	H
1	Data e Hora	PH (Ph)	Temperatura da Água (°C)	Temperatura do Ar (°C)	Oxigênio Dissolvido (ppm)	Condutividade Elétrica (uS/cm)	Sólidos Totais Dissolvidos (ppm)	Turbidez (N)
2	05/10/2022 00:08	6,6	31,3	28,77	0,19	55,06		27,53
3	05/10/2022 00:18	6,59	31,3	28,54	0,18	55,06		27,53
4	05/10/2022 00:28	6,58	31,3	28,48	0,19	55,09		27,545
5	05/10/2022 00:37	6,58	31,3	27,98	0,19	55,09		27,545
6	05/10/2022 00:49	6,57	31,3	27,66	0,18	55,09		27,545
7	05/10/2022 00:59	6,58	31,3	27,25	0,18	55,17		27,585
8	05/10/2022 01:07	6,59	31,3	27,13	0,19	55,14		27,57
9	05/10/2022 01:18	6,56	31,3	27,25	0,19	55,2		27,6
10	05/10/2022 01:29	6,55	31,3	27,28	0,18	55,28		27,64
11	05/10/2022 01:37	6,58	31,3	27,07	0,18	55,19		27,595
12	05/10/2022 01:47	6,58	31,3	27,16	0,19	55,27		27,635
13	05/10/2022 01:57	6,56	31,3	27,07	0,19	55,3		27,65
14	05/10/2022 02:09	6,55	31,3	26,9	0,18	55,32		27,66
15	05/10/2022 02:11	6,55	31,3	26,9	0,18	55,32		27,66
16	05/10/2022 02:20	6,55	31,3	26,9	0,18	55,36		27,68
17	05/10/2022 02:28	6,55	31,3	26,93	0,19	55,31		27,655
18	05/10/2022 02:38	6,55	31,3	26,81	0,19	55,33		27,665
19	05/10/2022 02:40	6,55	31,3	26,81	0,18	55,33		27,665
20	05/10/2022 02:47	6,54	31,3	26,72	0,18	55,31		27,655
21	05/10/2022 02:57	6,55	31,3	26,52	0,19	55,4		27,7
22	05/10/2022 03:07	6,55	31,3	26,4	0,19	55,33		27,665
23	05/10/2022 03:20	6,56	31,3	26,31	0,18	55,44		27,72
24	05/10/2022 03:30	6,57	31,3	26,22	0,18	55,35		27,675
25	05/10/2022 03:40	6,56	31,3	26,13	0,18	55,4		27,7
26	05/10/2022 03:48	6,56	31,3	25,93	0,19	55,38		27,69
27	05/10/2022 03:57	6,56	31,3	25,87	0,19	55,28		27,64
28	05/10/2022 03:59	6,56	31,3	25,87	0,18	55,28		27,64
29	05/10/2022 04:07	6,57	31,3	25,87	0,19	55,29		27,645

Fonte: (Elaborado pelo autor, 2022)

## 8.8 Página de configuração de um posto de monitoramento

O administrador pode realizar a edição de um PDM com as informações obrigatórias, estas estão indicadas por um asterisco vermelho ao lado do campo como exposto na figura 50.

Figura 50 – Página de edição de informações de um PDM

The screenshot displays the 'Configuração da remota' (Remote Configuration) page. It is divided into several sections:

- Header:** 'WATER INSPECTION' logo, navigation links, and user profile 'Allan Amorim'.
- Left Sidebar:** 'Mapa', 'Lista de PDM's', 'Configurações', and 'Usuários'.
- Main Content Area:**
  - Posto de monitoramento:** Overview with fields for 'Código' (14990100), 'Ponto de Instalação' (Caixa D'Água), 'Município / UF' (Manaus / AM), 'Energia' (Online), 'Rede Elétrica', and 'Modo de Operação' (Normal).
  - Parâmetros Físico/Químico:** Real-time data for pH (5.32), Temperature of Water (30.71 °C), Temperature of Air (28.87 °C), Dissolved Oxygen (0.19 ppm), Electrical Conductivity (59 uS/cm), Total Dissolved Solids (30 ppm), Turbidity (55.84 NTU), and Salinity (0.020 PSU).
  - Configuração da remota:** A grid of form fields for:
    - Dados da remota:** Código\*, Modelo\*, Tipo de energia\*.
    - Localização:** Bacia\*, Rio\*, Ponto de instalação\*, UF\*, Município\*, Latitude\*, Longitude\*.
    - +Informação:** Vazão\*, Cor da Água\*, Tipo do sistema aquático\*.
    - Período de certificação\*** and **Classificação do curso da água\***.

A 'SALVAR' (Save) button is located at the bottom right of the configuration area.

Fonte: (Elaborado pelo autor, 2022)

Na seção de Modos de Operação, o administrador pode realizar a configuração do modo de operação descrito no *Caso de Uso UC010* da tabela 28. Para o auxílio na usabilidade do sistema, foi desenvolvido os ícones de *toggle* que contêm instruções de uso. Para ativá-los, basta clicar em cima do ícone ao lado de cada campo, como exibido na figura 51.

Figura 51 – Página de configuração do modo de operação de um posto de monitoramento

The screenshot displays the 'WATER INSPECTION' interface. At the top, it shows the user 'Allan Amorin' and the page title 'Detalhes de um posto de monitoramento'. The main content is divided into several sections:

- Status do dispositivo:**
  - Código: 14990100
  - Ponto de Instalação: Caixa D'Água
  - Município / UF: Manaus / AM
  - Energia: Rede Elétrica
  - Modo de Operação: Normal
  - Status: Online (há uma hora atrás)
- Parâmetros Físico/Químico (+23) há uma hora atrás:**
  - pH: 5.32
  - Temperatura da Água: 30.71 °C
  - Temperatura do Ar: 28.87 °C
  - Orignio Dissolvido: 0.19 ppm
  - Condutividade Elétrica: 59 uS/cm
  - Sólidos Totais Dissolvidos: 30 ppm
  - Turbidez: 55.84 NTU
  - Salinidade: 0.020 PSU
- Configuração da remota:**
  - Informações
  - Modos de operação** (highlighted)
  - Alertas
- Configuração de períodos:**
  - EM CONDIÇÃO NORMAL:**

	Com Energia concessionária	Perda de Energia (c/ concessionária)	Energia Solar (s/ concessionária)
Amostragem	00:00	00:00	00:00
  - EM CONDIÇÃO DE ANOMALIAS DE PARÂMETROS COLETADOS:**

	Com Energia concessionária	Perda de Energia (c/ concessionária)	Energia Solar (s/ concessionária)
Amostragem	00:00	00:00	00:00
Evento de anomalia	00:00	00:00	00:00

Annotations in the image explain that the time for collection and sending of samples is 00:00 and that the anomaly state occurs when parameters are collected outside established limits.

Fonte: (Elaborado pelo autor, 2022)

Por fim, conforme a figura 52, o administrador também poderá configurar os valores de máximo e mínimo dos parâmetros físico-químicos da água (*Caso de Uso UC009* da Tabela 27). Estes serão os valores base para a geração de alarmes no sistema.

Figura 52 – Página de configuração de alertas de um posto de monitoramento

The screenshot displays the 'Configuração de alertas' (Alert Configuration) page for a monitoring station. The interface includes a sidebar with navigation options like 'Mapa', 'Lista de PDM's', 'Configurações', and 'Usuários'. The main content area is divided into several sections:

- Posto de monitoramento:** Displays station details such as 'Código: 14990100', 'Ponto de Instalação: Caixa D'Água', and 'Município / UF: Manaus / AM'. It also shows the status as 'Online' and power source as 'Rede Elétrica'.
- Parâmetros Físico/Químico:** A dashboard showing real-time values for pH (5.32), Temperatura da Água (30.71 °C), Temperatura do Ar (28.87 °C), Origênio Dissolvido (0.19 ppm), Condutividade Elétrica (59 uS/cm), Sólidos Totais Dissolvidos (30 ppm), Turbidez (55.84 NTU), and Salinidade (0.020 PSU).
- Configuração da remota:** A sidebar with buttons for 'Informações', 'Modos de operação', and 'Alertas'.
- Parâmetros de Coleta:** A table defining alert thresholds for various parameters. The 'Oxigênio Dissolvido (ppm)' row shows a minimum of 10.45 and a maximum of 00.00, with a red box around the 00.00 value and the label 'Obrigatório' (Mandatory).

Parâmetros de Coleta	Mínimo	Máximo
Oxigênio Dissolvido (ppm)	10,45	00,00 Obrigatório
Condutividade Elétrica (uS/cm)	00,00	00,00
PH (pH)	00,00	00,00
Turbidez (NTU)	00,00	00,00
Temperatura da Água (°C)	00,00	00,00
Temperatura do Ar (°C)	00,00	00,00
Sólidos Totais Dissolvidos (ppm)	00,00	00,00
Salinidade (PSU)	00,00	00,00

© 2022, Projeto TCC - Isaque Wilson, Manaus/AM

Fonte: (Elaborado pelo autor, 2022)

## 8.9 Página de listagem de usuários do sistema

Esta página possui a listagem contendo todos os usuários cadastrados na plataforma. Para fins de testes, foram cadastrados 47 usuários, sendo 44 ativos e 3 classificados como inativos. O usuário administrador pode clicar no ícone de lápis à direita de cada usuário cadastrado para editá-lo e clicar no ícone de lixeira para excluí-lo. O administrador do sistema pode clicar nos componentes de caixa de seleção à esquerda do *ID* de cada usuário para selecioná-los e editar vários de uma vez ou excluir vários de uma vez. Estes elementos podem ser visualizados na figura 53.

A partir da tela de listagem, também é possível ir para a criação de um novo analista do sistema. O botão de *Novo Usuário*, que está no canto superior direito da tela, direciona o usuário administrador à página de cadastro/edição de um usuário descrita na próxima seção.

Figura 53 – Página de listagem de usuários cadastrados na plataforma

WATER INSPECTION

Páginas / Configurações

Usuários

Usuários

Ativos (3) Inativos (44)

(47) Usuários

NOVO USUÁRIO

ID	Nome	Email	Localidade	Função	Status	
1233324	marcos andré	marcos@gmail.com	Parintins, AM	Analista	Ativo	
5454545	Julio Feltosa	jf@gmail.com	Manaus, AM	Administrador	Inativo	
5433566	Allan Amorin	allan@gmail.com	Manacapuru, AM	Administrador	Inativo	
5544444	Jelvis Rodrigues	allan@gmail.com	Belém, PA	Analista	Ativo	
7777775	Macon Lopes	maicon@gmail.com	Presidente Figueiredo, AM	Analista	Ativo	
4454545	Alessandra Martins	Ales@gmail.com	Itanduba, AM	Analista	Ativo	

1 2 ... 9 10 >

© 2022, Projeto TCC - Isaque Wilson, Manaus/AM

Termos de uso Simples Blog License

Fonte: (Elaborado pelo autor, 2022)

## 8.10 Página de cadastro/edição de um usuário

A figura 54 representa a tela do sistema na funcionalidade de cadastro de usuário, quando o administrador necessita adicionar um novo analista ao sistema. Os campos de nome, sobrenome, email, cidade, estado são as informações básicas de identificação. Já os campos de função e status são para a diferenciação de tipo de usuário (administrador ou analista) e para sua situação no sistema (usuário ativo ou inativo) respectivamente. Uma vez criado o usuário, uma senha temporária é enviada ao email cadastrado, onde é redefinida no primeiro acesso.

Figura 54 – Página de cadastro de novo usuário na plataforma.

WATER INSPECTION

Páginas / Configurações

Usuários

Mapa

Lista de PCD's

Configurações

Usuários

Cadastro

**Novo usuário**

**Nome** \*  
Seu primeiro nome

**Sobrenome** \*  
Seu segundo nome

**Email** \*  
Seu melhor email

**Cidade** \*  
Sua cidade

**Estado** \*  
Seu estado

**Função** \*  
Função do usuário

**Status** \*  
Ativar/desativar usuário

Uma senha temporária do usuário será enviado para o email informado.

SALVAR

© 2022, Projeto TCC - Isaque Wilson, Manaus/AM.

Termos de uso | Simples | Blog | License

Fonte: (Elaborado pelo autor, 2022)

Já a tela representada na figura 55 abaixo é utilizada quando o usuário administrador necessita editar um usuário já existente. As informações do usuário a ser editado são carregadas em seus devidos campos. O administrador basta inserir a alteração no campo desejado e clicar em salvar. As informações atualizadas são salvas no banco de dados.

Figura 55 – Página de edição de usuário na plataforma.

WATER INSPECTION

Páginas / Configurações

Usuários

Mapa

Lista de PCD's

Configurações

Usuários

Edição

**Editar usuário**

**Nome** \*  
Marcos

**Sobrenome** \*  
André

**Email** \*  
marcos@gmail.com

**Cidade** \*  
Parintins

**Estado** \*  
Amazonas

**Função** \*  
Analista

**Status** \*  
Ativo

SALVAR

© 2022, Projeto TCC - Isaque Wilson, Manaus/AM.

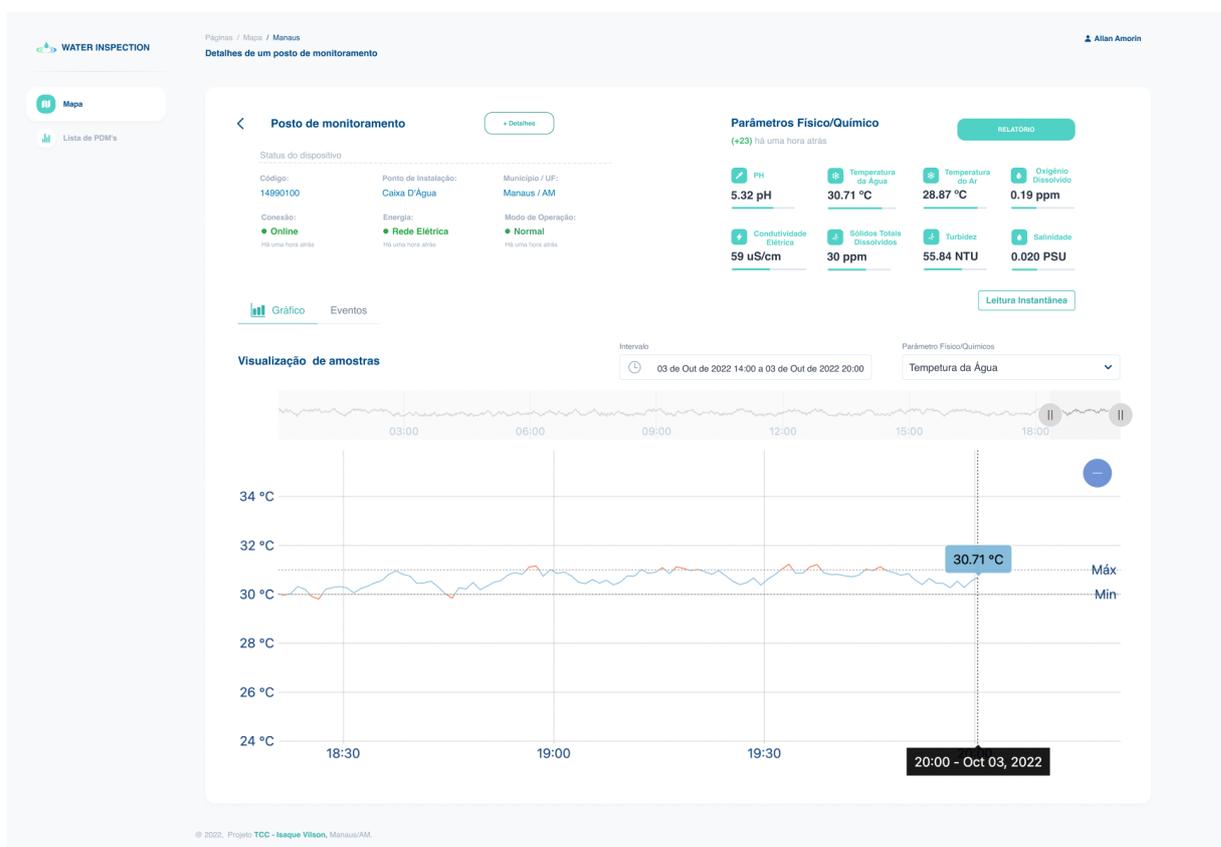
Termos de uso | Simples | Blog | License

Fonte: (Elaborado pelo autor, 2022)

## 8.11 Diferenças na página entre um usuário analista e um administrador

Também foi estabelecido permissões dentro da plataforma. Como exibido na figura 56, um usuário analista não terá acesso à seção de configuração de usuário descrita anteriormente. Além disso, na tela de detalhe dos Postos de Monitoramentos, é possível perceber que existem as sub-seções de Gráfico e Evento. No entanto, a sub-seção de Configuração não está disponível ao usuário analista, pois nela há as funcionalidades restritas apenas ao administrador.

Figura 56 – Página de detalhes de um posto de monitoramento para um usuário com perfil de analista.



Fonte: (Elaborado pelo autor, 2022)

## 9 CONCLUSÃO

Este trabalho expôs a implementação de uma solução viável para a coleta de parâmetros físico-químicos da água por intermédio de uma aplicação web que atendesse aos mais diversos tipos de sensores.

Durante o desenvolvimento desta pesquisa, foram verificadas as principais necessidades ao realizar a captura dos parâmetros da água. Para isso, foram verificados os diferentes tipos de sensores capazes de atender a este propósito e como exibir os dados coletados. Além disso, uma pesquisa dentre as arquiteturas mais apropriadas para prover o melhor suporte às tecnologias indicadas para a comunicação dos parâmetros coletados foi desenvolvida. Para a realização das análises e experimentos, foi necessário o conhecimento de diversas áreas cursadas durante a graduação e as evidências destes experimentos e da implementação da solução proposta foram apresentadas neste trabalho. Acerca da hipótese e dos resultados obtidos, foi possível concluir que o sistema desenvolvido é capaz de atender as demandas de captura de parâmetros da água, independente do protocolo de comunicação ou sensor escolhido pelo usuário. Estes parâmetros podem ser visualizados na aplicação *web Water Inspection*.

Vale pôr em evidência que o trabalho desenvolvido pode ser aplicado à UEA como forma de coletar e monitorar dados da água melhorando a nota da instituição no ranking do *GREENMETRIC*. Além disso, apesar de a compatibilidade do sistema ser ampla, ainda é necessário conhecimentos técnicos na área para integrar diferentes tipos de dispositivos ao software.

### 9.1 Sugestão de trabalhos futuros

Para a realização de trabalhos futuros, recomenda-se a implementação da funcionalidade de cadastrar novos dispositivos pela plataforma web, já que no sistema implementado somente dispositivos inseridos no momento da instalação do software são monitorados.

Além disso, com uso da plataforma desenvolvida nesse projeto de pesquisa, um estudo mais aprimorado pode ser realizado sobre a qualidade da água na Escola Superior de Tecnologia (EST) coletando mais amostras e configurando, para cada parâmetro físico-químico, os alertas com limite inferior e superior mais apropriados para uma classificação de boa qualidade hídrica. Com isso, será possível efetuar uma submissão de medidas práticas a favor da água para o ranking do *GREENMETRIC*, e assim, elevar a pontuação da universidade quanto a conservação e uso da água.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ALBUQUERQUE, D. T. P. U. B. D. *Sensores industriais: fundamentos e aplicações*. [S.l.: s.n.], 2020. <<https://books.google.com.br/books?hl=pt-BR&lr=&id=1qgPEAAAQBAJ&oi=fnd&pg=PP15&dq=sensores&ots=RJ5C-InFUG&sig=CkJLkZMSWIOTp0sNN4Wu8FyJx1Y#v=onepage&q=sensores&f=false>>.
- AMBIENTE, M. *Preservação Dos Recursos Naturais*. 2022. Acesso em: 08 out. 2022. Disponível em: <<https://meioambiente.culturamix.com/recursos-naturais/preservacao-dos-recursos-naturais>>.
- ARAUJO, D. *SOLID: guia completo sobre os 5 princípios da POO!* 2021. Acesso em: 08 out. 2022. Disponível em: <<https://blog.betrybe.com/linguagem-de-programacao/solid-cinco-principios-poo/>>.
- AWS. *Amazon SQS - Serviço de enfileiramento de mensagens*. 2022. Acesso em: 08 out. 2022. Disponível em: <<https://aws.amazon.com/pt/sqs/>>.
- AWS. *AWS SNS - Amazon Simple Notification Service*. 2022. Acesso em: 08 out. 2022. Disponível em: <<https://aws.amazon.com/pt/sns/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc>>.
- AWS. *O que é AWS? Como funciona Amazon Web Services*. 2022. Acesso em: 08 out. 2022. Disponível em: <[https://aws.amazon.com/pt/what-is-aws/?nc1=f\\_cc](https://aws.amazon.com/pt/what-is-aws/?nc1=f_cc)>.
- AWS. *Simplify Your Pub/Sub Messaging with Amazon SNS Message Filtering*. 2022. Acesso em: 08 out. 2022. Disponível em: <<https://aws.amazon.com/blogs/compute/simplify-pubsub-messaging-with-amazon-sns-message-filtering/>>.
- BANKER, K. *MongoDB in Action: Covers MongoDB version 3.0*. 2016. Acesso em: 25 nov. 2021. Disponível em: <<https://books.google.com.br/books?hl=pt-BR&lr=&id=kzkeEAAAQBAJ&oi=fnd&pg=PT21&dq=what+is+mongodb&ots=8U3-nX3-9a&sig=dmo1ls3zk8N-c8CQrgY0aT4EjFA#v=onepage&q=what%20is%20mongodb&f=true>>.
- BRIEF, I. L. *Internet of things*. 2019. Acesso em: 22 nov. 2021. Disponível em: <[https://elk.adalidda.com/2019/09/IRENA\\_Internet\\_Of\\_Things\\_2019.pdf](https://elk.adalidda.com/2019/09/IRENA_Internet_Of_Things_2019.pdf)>.
- CARRARO, M. *Ferramenta para Geração de Telas de Supervisão*. 2017. 14 p. Acesso em: 08 out. 2022. Disponível em: <[https://repositorio.ufsc.br/bitstream/handle/123456789/199990/PFC%20Murilo%20Ramos%20Carraro\\_2016-2.pdf?sequence=1&isAllowed=y](https://repositorio.ufsc.br/bitstream/handle/123456789/199990/PFC%20Murilo%20Ramos%20Carraro_2016-2.pdf?sequence=1&isAllowed=y)>.

- DEV MEDIA. *Princípios da Engenharia de Software*. 2013. Acesso em: 08 out. 2022. Disponível em: <https://www.devmedia.com.br/principios-da-engenharia-de-software/29630>.
- EVANS, N. A. N. A. R. A systematic mapping study in microservice architecture. *2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA)*, p. 44–51, 2021. Acesso em: 08 out. 2022. Disponível em: <https://sci-hub.shhttps://ieeexplore.ieee.org/abstract/document/7796008>.
- FABRI, J. A. *A aplicabilidade do modelo cascata na engenharia de software*. 2013. Acesso em: 08 out. 2022. Disponível em: <https://engenhariasoftware.wordpress.com/2013/01/24/a-aplicabilidade-do-modelo-cascata-na-engenharia-de-software/>.
- FERNANDES, A. B. C. V. Uma análise dos protocolos de comunicação para internet das coisas. *BIBLIOTECA DIGITAL DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO*, 2020. Acesso em: 07 nov. 2021. Disponível em: <https://sol.sbc.org.br/index.php/courb/article/view/7475/7357>.
- FREIRE, H. *React, Vue, Angular, conheça suas vantagens e desvantagens e qual é melhor para seus projetos*. 2019. Acesso em: 08 out. 2022. Disponível em: <https://henrique-freire.medium.com/react-vue-angular-conhe%C3%A7a-suas-vantagens-e-desvantagens-e-qual-%C3%A9-melhor-para-seus-projetos-53734bb3d37f>.
- GREENMETRIC, U. *UI GREENMETRIC WORLD UNIVERSITY RANKING*. 2021. <https://greenmetric.ui.ac.id/about/welcome>.
- HILLAR, G. *MQTT Essentials-A lightweight IoT protocol*. [S.l.: s.n.], 2017. <https://books.google.com.br/books?hl=pt-BR&lr=&id=40EwDwAAQBAJ&oi=fnd&pg=PP1&dq=mqtt+book&ots=fLGAIND5nG&sig=ghyYcfga4KavAn8R01fPxcTunu4#v=onepage&q=mqtt%20book&f=false>.
- KILIÇ, Z. International journal of hydrology. *The importance of water and conscious use of water*, v. 4, n. 5, p. 239–241, 2020. Acesso em: 07 nov. 2021. Disponível em: [https://www.researchgate.net/profile/Zeyneb-Kilic/publication/350210868\\_The\\_importance\\_of\\_water\\_and\\_conscious\\_use\\_of\\_water/links/6093d46a92851c490fbc82c6/The-importance-of-water-and-conscious-use-of-water.pdf](https://www.researchgate.net/profile/Zeyneb-Kilic/publication/350210868_The_importance_of_water_and_conscious_use_of_water/links/6093d46a92851c490fbc82c6/The-importance-of-water-and-conscious-use-of-water.pdf).
- L, S. *Why You Should Care If Your University Is Sustainable*. [S.l.]: QS Top Universities, 2021. <https://www.topuniversities.com/courses/sustainable-development/why-you-should-care-if-your-university-sustainable>.
- MARTIN, R. *Arquitetura Limpa: O guia do artesão para estrutura e design de software*. 2019. Acesso em: 08 out. 2022. Disponível em: [http://www.iaeng.org/IJCS/issues\\_v49/issue\\_1/IJCS\\_49\\_1\\_28.pdf](http://www.iaeng.org/IJCS/issues_v49/issue_1/IJCS_49_1_28.pdf).
- MELO thays. *Avaliação de Desempenho de Protocolos de Comunicação para Aplicações IoT*. 2022. Acesso em: 08 out. 2022. Disponível em: <https://repositorio.ufpe.br/bitstream/123456789/35370/1/DISSERTA%C3%87%C3%83O%20Thays%20Melo%20de%20Moraes.pdf>.
- MONGODB. *MongoDB: The Developer Data Platform*. 2022. Acesso em: 03 out. 2022. Disponível em: <https://www.mongodb.com/>.
- MORAES, W. *Construindo aplicações com NodeJS*. [S.l.: s.n.], 2021.

PALARMINI, L. F. *Conheça os principais protocolos para IoT*. 2020. Acesso em: 22 nov. 2021. Disponível em: <https://www.filipeflop.com/blog/conheca-os-principais-protocolos-para-iot/>.

POSTGRESQL. *About PostgreSQL*. 2022. Acesso em: 03 out. 2022. Disponível em: <https://www.postgresql.org/about/>.

REDDY, A. *Why is React so Popular?* 2020. Acesso em: 03 out. 2022. Disponível em: <https://dev.to/nrabhiram/why-is-react-so-popular-1oa1>.

REDHAT. *O que são microsserviços?* 2021. Acesso em: 03 out. 2022. Disponível em: <https://www.redhat.com/pt-br/topics/microservices/what-are-microservices>.

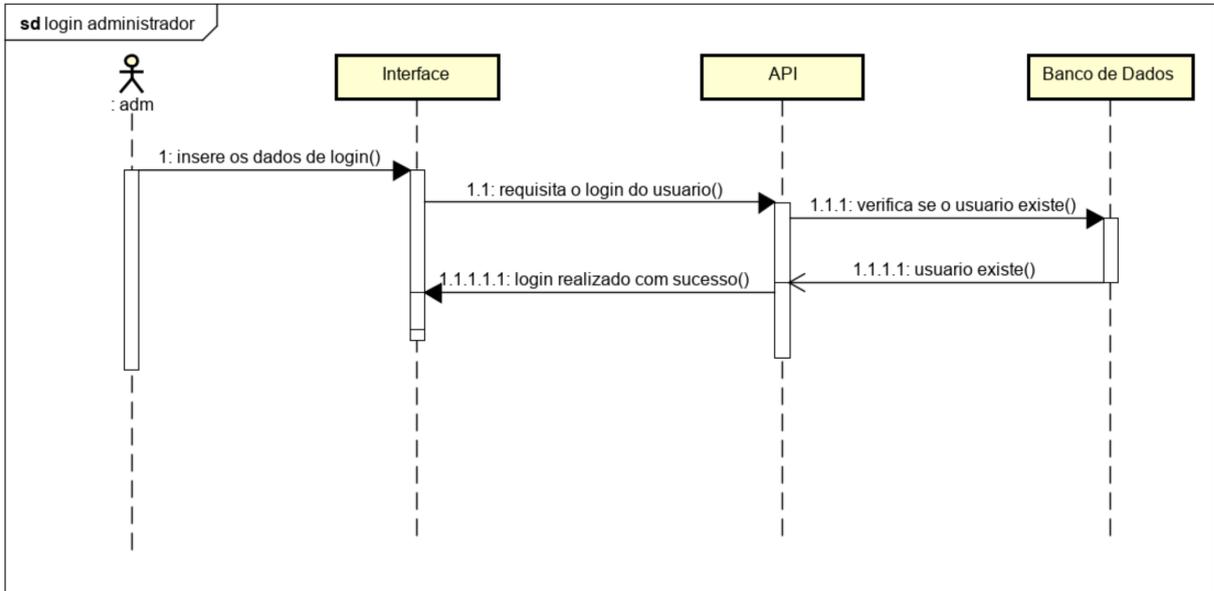
ROWLAND E.R.; CROSS, M. H. H. *Considering Multiple Futures: Scenario Planning To Address Uncertainty in Natural Resource Conservation*. [S.l.: s.n.], 2014. <https://www.fws.gov/home/climatechange/pdf/Scenario-Planning-Report.pdf>.

SENADO, A. *Em busca de um novo modelo de gestão para o uso da água*. 2022. Acesso em: 08 out. 2022. Disponível em: <https://www12.senado.leg.br/noticias/especiais/especial-cidadania/em-busca-de-um-novo-modelo-de-gestao-para-o-uso-da-agua>.

## A APÊNDICE - DIAGRAMA DE SEQUÊNCIA - UML

### A.1 Diagrama de sequência para *login* do Administrador

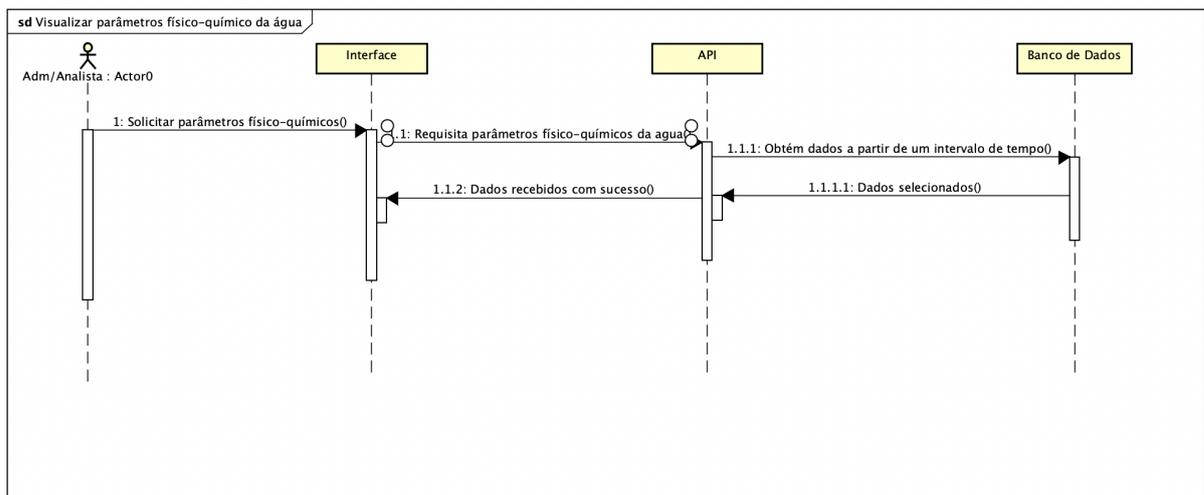
Diagrama de sequência para *login* do Administrador



Fonte: (Elaborado pelo autor, 2022)

### A.2 Diagrama de sequência para obter parâmetros físico-químico da água

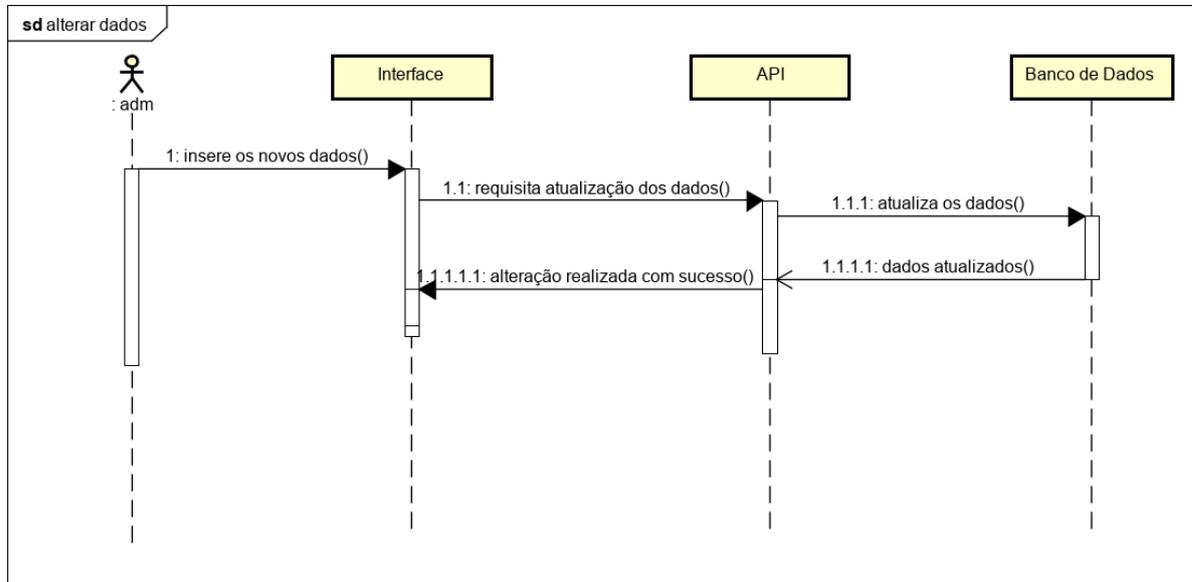
Diagrama de sequência para obter parâmetros físico-químico da água



Fonte: (Elaborado pelo autor, 2022)

### A.3 Diagrama de Sequência para editar dados do usuário

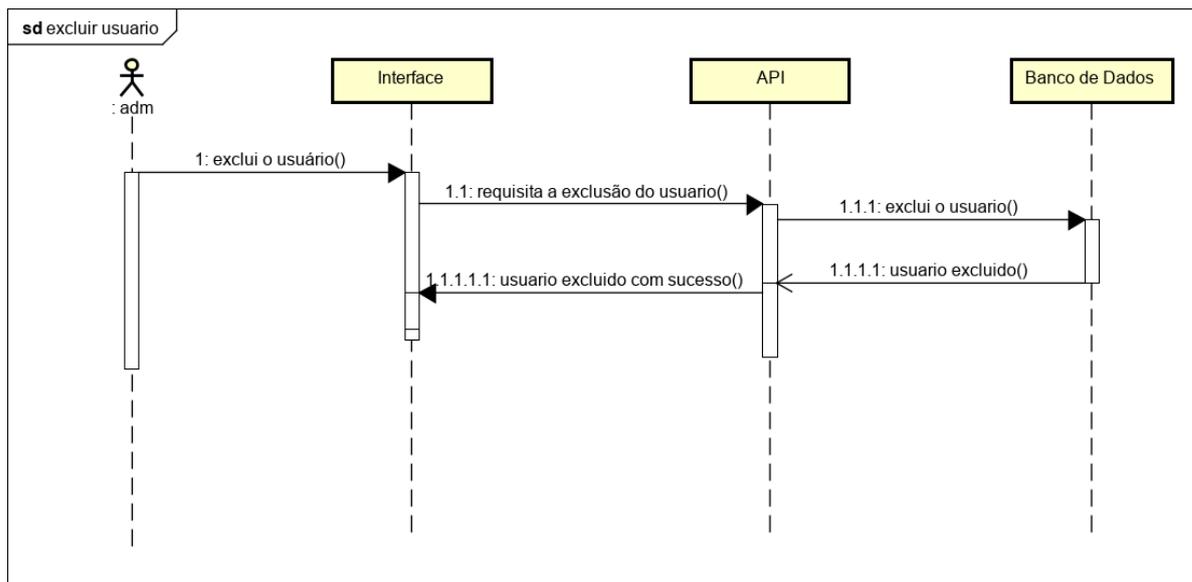
Diagrama de Sequência para editar dados do usuário



Fonte: (Elaborado pelo autor, 2022)

### A.4 Diagrama de Sequência para deletar dados do usuário

Diagrama de Sequência para deletar dados do usuário



Fonte: (Elaborado pelo autor, 2022)