



UEPB

**UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS I – CAMPINA GRANDE
CENTRO DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO
CURSO DE GRADUAÇÃO EM BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

ANTONIO LEOMAR FERREIRA SOARES

**MONITOR WATER: DISPOSITIVO IOT E APLICAÇÃO WEB PARA
MONITORAMENTO DO CONSUMO DE ÁGUA**

**CAMPINA GRANDE
2023**

ANTONIO LEOMAR FERREIRA SOARES

**MONITOR WATER: DISPOSITIVO IOT E APLICAÇÃO WEB PARA
MONITORAMENTO DO CONSUMO DE ÁGUA**

Trabalho de Conclusão de Curso de Graduação apresentado ao Curso de Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Fábio Luiz Leite Júnior

**CAMPINA GRANDE
2023**

É expressamente proibido a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano do trabalho.

S676m Soares, Antonio Leomar Ferreira.
Monitor water [manuscrito] : dispositivo IOT e aplicação web para monitoramento do consumo de água / Antonio Leomar Ferreira Soares. - 2023.
60 p. : il. colorido.

Digitado.

Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) - Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia, 2023.

"Orientação : Prof. Dr. Fábio Luiz Leite Júnior , Coordenação do Curso de Computação - CCT. "

1. Sistema web. 2. Internet das coisas. 3. Consumo de água. I. Título

21. ed. CDD 005.3

ANTONIO LEOMAR FERREIRA SOARES

**MONITOR WATER: DISPOSITIVO IOT E APLICAÇÃO WEB PARA
MONITORAMENTO DO CONSUMO DE ÁGUA**

Trabalho de Conclusão de Curso de Graduação apresentado ao Curso de Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de Bacharel em Ciência da Computação.

Aprovada em: 31 / Outubro / 2023.

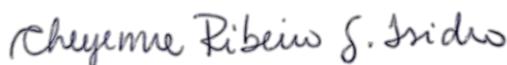
BANCA EXAMINADORA



Prof. Dr. Fábio Luiz Leite Júnior (CCT/UEPB)
Orientador(a)



Prof. Dr. Paulo Eduardo e Silva Barbosa (CCT/UEPB)
Examinador(a)



Profa. Me. Cheyenne Ribeiro Guedes Isidro (CCT/UEPB)
Examinador(a)

A Deus, pela oportunidade, força e coragem, DEDICO.

AGRADECIMENTOS

A Deus, pela presença, proteção, saúde e bençãos em minha vida.

A meus pais, Marcos e Fátima, pelo amor, inspiração, exemplos, ensinamentos e educação resultantes na minha formação como ser humano.

Aos meus irmãos, Letícia e Júnior, pelo apoio, companheirismo e momentos de alegria.

A minha noiva Marcella, pelo amor, presença ao meu lado, incentivo, paciência e compreensão nas ausências.

A toda minha família, pelo suporte, torcida e carinho.

Aos professores, em especial, meu orientador Prof. Fábio Leite, e colegas do Curso de Ciência da Computação da UEPB, que contribuíram ao longo dessa caminhada, por meio dos ensinamentos, discussões e apoio.

À UEPB, universidade pública e gratuita de excelente qualidade, pela oportunidade de desenvolvimento intelectual e formação profissional.

E por fim, a todos que, de alguma forma, contribuíram para a conclusão deste trabalho.

Muito obrigado!

“A melhor maneira de prever o futuro é inventá-lo.”
(Alan Kay, cientista da computação, em 1971)

RESUMO

O atendimento às demandas de água para abastecimento nas cidades está se tornando cada dia mais desafiador, em virtude da escassez e distância dos mananciais. Além disso, percebe-se no cotidiano situações de altas taxas de perdas e desperdícios de água, seja por gestão deficiente do sistema de abastecimento e/ou mau uso do usuário. Desta forma, o Gerenciamento da Demanda de Água (GDA) apresenta-se como alternativa viável para solução do problema, pois tem como foco o uso eficiente da água, tendo as medidas estruturais tecnológicas como eixo de ação aplicado neste trabalho. Neste sentido, o objetivo desse estudo contempla o desenvolvimento de um sistema web e um protótipo de hidrômetro com tecnologia IoT para realizar o monitoramento remoto do consumo de água, promovendo assim um acompanhamento mais preciso, rápido e em tempo real do perfil de consumo do ponto monitorado, proporcionando a descoberta de anomalias, tais como: vazamentos de água, consumos excessivos etc. Como resultado desta prova de conceito, foi produzida uma versão inicial do sistema proposto, denominado “Monitor Water”, implementado a parte *frontend* com a tecnologia React e o *backend* com o framework Spring Boot. Para o protótipo de hidrômetro IoT foi utilizado microcontrolador NodeMCU – ESP32 com capacidade de comunicação sem fio e sensor de fluxo para captação da vazão consumida. Por fim, conclui-se que ferramentas tecnológicas da área da Ciência da Computação propiciam meios eficientes para a melhoria e otimização na gestão de processos gerais, que no caso concreto demonstrado, visou o uso eficiente de água.

Palavras-Chave: gerenciamento da demanda de água; sistema web; internet das coisas.

ABSTRACT

Meeting water supply demands in cities is becoming more complicated every day, due to the scarcity and distance of springs. In addition, situations of high rates of loss and waste of water can be seen in everyday life, either due to poor management of the supply system and/or misuse by the user. In this way, water demand management (WDM) presents itself as a viable alternative for solving the problem, as it focuses on the efficient use of water, with the technological measures developed as the axis of action applied in this work. In this sense, the objective of this study includes the development of a web system and a water meter prototype with IoT technology to carry out remote monitoring of water consumption, thus promoting a more accurate, fast and real-time monitoring of the point's consumption profile. monitored, providing the discovery of anomalies, such as: water leaks, excessive consumption, etc. As a result of this proof of concept, an initial version of the standard system was produced, called "monitor water", implementing the frontend part with React technology and the backend with the Spring Boot framework. For the IoT water meter prototype, the NodeMCU - ESP32 microcontroller with wireless communication capability and a flow sensor to generate consumed water was used. Finally, it is concluded that technological tools in the area of computer science provide efficient means for the improvement and optimization in the management of general processes, which in the concrete case demonstrated, aimed at the efficient use of water.

Keywords: water demand management; web system; internet of things.

LISTA DE ILUSTRAÇÕES

| | |
|---|----|
| Figura 1 - Etapas metodológicas..... | 24 |
| Figura 2 - Tela principal do <i>software</i> draw.io..... | 26 |
| Figura 3 - Diagrama do Model-View-Controller | 27 |
| Figura 4 - Logo do visual <i>studio code</i> | 28 |
| Figura 5 - Logo do Git e Github | 29 |
| Figura 6 - Logo do React..... | 30 |
| Figura 7 - Logo do <i>Spring Boot</i> | 31 |
| Figura 8 - Logo do <i>Postman</i> | 32 |
| Figura 9 - Logo do H2 <i>database</i> | 32 |
| Figura 10 - NodeMCU Esp-8266..... | 34 |
| Figura 11 - <i>Protoboard</i> | 35 |
| Figura 12 - Partes componentes do sensor de fluxo | 36 |
| Figura 13 - Sensor de fluxo | 37 |
| Figura 14 - Tabela e gráfico para conversão da frequência em vazão..... | 37 |
| Figura 15 - <i>Jumpers</i> | 38 |
| Figura 16 - Caixa de proteção..... | 38 |
| Figura 17 - Carregador com cabo micro USB | 39 |
| Figura 18 – Ambiente de programação IDE Arduino..... | 40 |
| Figura 19 - Diagrama de casos de uso | 43 |
| Figura 20 - Diagrama de entidade relacionamento | 44 |
| Figura 21 - Tela inicial | 45 |
| Figura 22 - Tela de inserção de dados..... | 45 |
| Figura 23 - Tela inicial com ponto de monitoramento cadastrado..... | 46 |
| Figura 24 - Tela de confirmação de exclusão..... | 47 |
| Figura 25 - Tela de edição de dados..... | 47 |
| Figura 26 - Aba “Info Gerais”..... | 48 |
| Figura 27 - Aba “Gráfico”..... | 49 |
| Figura 28 - Caixa de alerta para seleção | 49 |
| Figura 29 - Gráfico de consumo de água | 50 |
| Figura 30 - Tabela de consumo de água..... | 51 |
| Figura 31 - Trecho do código da camada <i>domain</i> | 52 |
| Figura 32 - Trecho do código da camada <i>controller</i> | 52 |

| | |
|--|----|
| Figura 33 - Trecho do código da camada <i>Repository</i> | 53 |
| Figura 34 - Protótipo do hidrômetro IoT | 54 |
| Figura 35 - Formato de envio da informação..... | 55 |

LISTA DE QUADROS

| | |
|--|----|
| Quadro 1 – Requisitos Funcionais | 42 |
| Quadro 2 – Requisitos Não Funcionais | 43 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|--------|------------------------------------|
| API | Application Programming Interface |
| CRUD | Create, Read, Update, Delete |
| CSV | comma-separated-values |
| GDA | Gerenciamento da demanda de água |
| GND | Ground |
| HTTP | HyperText Transfer Protocol |
| IA | inteligência artificial |
| ID | Identificador |
| IDE | Integrated Development Environment |
| IoT | Internet of Things |
| JPA | Java Persistence API |
| JSON | JavaScript Object Notation |
| Mpa | Mega Pascal |
| MVC | Model-View-Controller |
| ONU | Organização das Nações Unidas |
| PNG | Portable Network Graphics |
| PVC | Policloreto de Vinila |
| PWM | Pulse Width Modulation |
| REST | Representational State Transfer |
| RF | Requisitos Funcionais |
| RNF | Requisitos não funcionais |
| SVG | Scalable Vector Graphics |
| URI | Uniform Resource Identifier |
| URL | Uniform Resource Locator |
| USB | Universal Serial Bus |
| VsCode | Visual Studio Code |
| XML | eXtensible Markup Language |

SUMÁRIO

| | | |
|----------------|---|-----------|
| 1 | INTRODUÇÃO | 14 |
| 1.1 | Objetivos | 15 |
| 1.1.1 | Objetivo Geral | 15 |
| 1.1.2 | Objetivos Específicos | 15 |
| 2 | REFERENCIAL TEÓRICO | 17 |
| 2.1 | Desenvolvimento sustentável | 17 |
| 2.2 | Gestão da Demanda de Água (GDA) | 18 |
| 2.3 | Quarta revolução industrial ou indústria 4.0 | 19 |
| 2.4 | Internet das coisas (IOT) | 20 |
| 2.5 | Desenvolvimento WEB | 21 |
| 2.6 | Sistemas inteligentes de medição | 22 |
| 3 | METODOLOGIA | 24 |
| 3.1 | Etapa I - Desenvolvimento do sistema web | 25 |
| 3.1.1 | Engenharia de Requisitos | 25 |
| 3.1.2 | Diagrama de Caso de Uso | 26 |
| 3.1.3 | Diagrama Entidade Relacionamento | 26 |
| 3.1.4 | Arquitetura MVC | 27 |
| 3.1.5 | Tecnologias de uso comum do frontend e backend | 28 |
| 3.1.5.1 | <i>Visual Studio Code</i> | 28 |
| 3.1.5.2 | <i>Git e Github</i> | 29 |
| 3.1.6 | Tecnologias usadas no desenvolvimento frontend | 29 |
| 3.1.7 | Tecnologias usadas no desenvolvimento backend | 30 |
| 3.1.7.1 | <i>Spring boot</i> | 30 |
| 3.1.7.2 | <i>Postman</i> | 31 |
| 3.1.7.3 | <i>H2 Database</i> | 32 |
| 3.1.7.4 | <i>API REST</i> | 32 |
| 3.2 | Etapa II - Desenvolvimento do hidrômetro inteligente IoT | 34 |
| 3.2.1 | Hardware | 34 |
| 3.2.1.1 | <i>Microcontrolador</i> | 34 |
| 3.2.1.2 | <i>Placa protoboard</i> | 35 |
| 3.2.1.3 | <i>Sensor de fluxo</i> | 35 |

| | | |
|--------------|--|----|
| 3.2.1.4 | <i>Jumpers</i> | 37 |
| 3.2.1.5 | <i>Caixa protetora</i> | 38 |
| 3.2.1.6 | <i>Carregador com cabo micro USB</i> | 39 |
| 3.2.2 | Software | 39 |
| 4 | RESULTADOS E DISCUSSÕES | 41 |
| 4.1 | Desenvolvimento do sistema web | 41 |
| 4.1.1 | Engenharia de Requisitos | 41 |
| 4.1.2 | Diagrama de Casos de Uso | 43 |
| 4.1.3 | Diagrama de Entidade Relacionamento | 43 |
| 4.1.4 | Desenvolvimento frontend | 45 |
| 4.1.5 | Desenvolvimento backend | 51 |
| 4.2 | Desenvolvimento do hidrômetro IoT | 53 |
| 4.2.1 | Hardware | 53 |
| 4.2.2 | Software | 55 |
| 5 | CONCLUSÃO | 56 |
| | REFERÊNCIAS | 57 |

1 INTRODUÇÃO

A disponibilidade de água para abastecimento das populações está se tornando cada vez mais escassa e de difícil resolução, gerando um grande desafio para a humanidade, em virtude de diversos fatores, tais como: mudanças climáticas, aumento da população, poluição, falta de investimentos, mau uso do recurso etc.

No Brasil, país que detém uma das maiores bacias hidrográficas do mundo, ainda assim sofre com problemas de escassez de água devido à má distribuição do recurso (TOMAZ, 2001 apud MARINOSK et al. 2008). Essa situação é ainda mais crítica em regiões semiáridas, tal como o Nordeste brasileiro, onde possuem pluviosidade irregular, secas sucessivas e consequente disponibilidade hídrica limitada.

Contudo, para agravar ainda mais este cenário, percebe-se que a maioria dos municípios possuem sistemas de abastecimento com altos índices de perdas na distribuição de água, com média de 46,3%, relacionado potencialmente a falhas de gestão, inexistência ou ineficiência na medição (índice de macromedição de 49,42%), entre outros fatores que refletem tanto em perdas físicas, quanto de faturamento (SNIS, 2021 apud MEDEIROS, 2022).

Diante dessa problemática, escassez de água e uso ineficiente, as companhias de abastecimento de água têm como um dos principais desafios a aplicação de novas tecnologias e ações que garantam uma maior racionalização dos recursos hídricos e controle de perdas (GONÇALVES, 2018).

O Gerenciamento da Demanda de Água (GDA) apresenta-se como alternativa viável para proporcionar um uso eficiente de água e assim atender às necessidades futuras por abastecimento, tendo sua eficácia comprovada na literatura, seja por meios tecnológicos, campanhas educativas e atribuição de preço à água (FRIEDMAN et al, 2011).

Em tempos de indústria 4.0, cada vez mais presente no país, o uso de tecnologias, análises de cenários, riscos e materialidade, monitoramento e medidas preventivas se mostram ferramentas essenciais na gestão de perdas (SNIS, 2022).

Nesse contexto, o desenvolvimento de dispositivos automatizados de monitoramento e controle em redes de abastecimento de água pode auxiliar o processo de gestão das empresas e dos usuários, uma vez que permite ganhos

econômicos, sociais e ambientais, principalmente em áreas vulneráveis nas quais a disponibilidade de água é limitada (MEDEIROS, 2022).

Os sistemas de coleta de dados totalmente automatizados propiciam redução de custos operacionais, enquanto os consumidores têm a oportunidade de receberem informações contínuas sobre seu consumo, e assim, adaptar e mudar o comportamento (KAUFMANN, KÜNZEL e LOOCK, 2013).

As ferramentas tecnológicas advindas da Ciência da Computação, tais como desenvolvimento de programas e sistemas de informática, aplicação da tecnologia de Internet das Coisas (IoT, do inglês *Internet of Things*), armazenamento de dados, conectividade, automatização etc., podem servir de meios para ações voltadas ao combate às perdas e desperdícios, facilitando diagnósticos, monitoramento e tomada de decisão por parte dos agentes.

Neste sentido, visando a otimização dos serviços, a medição inteligente vem sendo utilizada como uma tecnologia em desenvolvimento para essa finalidade. Os medidores inteligentes, ou “*smart meters*” facilitam a disponibilidade de dados detalhados sobre os padrões de consumo dos usuários, tanto de água quanto de energia elétrica e gás (OTAKI; HONDA; UEDA, 2020).

Assim, analisando as possibilidades das ações estruturais de Gerenciamento da Demanda de Água com emprego de tecnologias (*smart meters*) como parte integrante da solução do problema, este trabalho apresenta uma proposta de desenvolvimento de um sistema web e de um dispositivo IoT para realizar a medição e monitoramento do perfil de consumo de água, visando identificar possíveis anomalias do ponto monitorado.

1.1 Objetivos

1.1.1 – Objetivo Geral

Auxiliar o monitoramento remoto do consumo e gestão de perdas e desperdícios de água, através do desenvolvimento de um sistema web e um protótipo de hidrômetro, utilizando tecnologia IoT.

1.1.2 – Objetivos Específicos

- Identificar os requisitos funcionais e não funcionais do sistema;
- Desenvolver o *frontend* da aplicação, através da tecnologia *React*;

- Desenvolver o *backend* da aplicação, através da tecnologia *Spring Boot*;
- Desenvolver um protótipo IoT para medição de vazões, com uso de sensores e microcontrolador.

2 REFERENCIAL TEÓRICO

2.1 Desenvolvimento sustentável

O desenvolvimento sustentável é entendido através do relatório de Bruntland como o desenvolvimento que atende às necessidades do presente, sem comprometer a possibilidade de as gerações futuras atenderem as suas próprias necessidades.

Siche et al. (2007) consideram sustentabilidade um conceito complexo e que possui distintos significados, mas em todos está intrínseco o conceito de equilíbrio da biosfera e do bem-estar da humanidade. Se o desenvolvimento atual não é sustentável, é porque se degradam serviços ambientais indispensáveis ao nosso bem-estar e que não podem ser substituídos pelo capital humano.

Assim, as ações humanas devem ser pautadas sob a ótica da preservação dos recursos naturais disponíveis, buscando o equilíbrio entre o uso e a capacidade de renovação do insumo, utilizando de técnicas adequadas e evoluções de conhecimento para o cumprimento desta meta.

Neste sentido, como parte de sua contribuição ao desenvolvimento econômico e social, deve-se utilizar a ciência e a tecnologia para descobrir, evitar e combater os riscos que ameaçam o meio ambiente, para solucionar os problemas ambientais e para o bem comum da humanidade (ONU, 1972).

Ainda, de acordo com a ONU (1972), em larga e tortuosa evolução da raça humana neste planeta, chegou-se a uma etapa em que, graças à rápida aceleração da ciência e da tecnologia, o homem adquiriu o poder de transformar de inúmeras maneiras, em uma escala sem precedentes, tudo que o cerca.

Logo, é imprescindível que a humanidade utilize essa capacidade de transformação propiciada pelas tecnologias em prol do desenvolvimento sustentável, agregando eficiência às operações e no consumo dos recursos, visando a redução do impacto ao meio ambiente.

Dentre os cenários mais preocupantes, sob o aspecto da degradação dos recursos naturais, encontra-se a água, pois em virtude da crescente demanda e escassez deste recurso, a vida da humanidade está sob risco.

Segundo Braga (2001), o abastecimento de água torna-se a cada dia mais problemático devido ao acentuado crescimento populacional, expansão das

atividades econômicas, elevação da demanda de água, oferta de água cada vez mais escassa e cara, e uso ineficiente de água.

A necessidade do uso sustentável dos recursos hídricos é um dos pontos existenciais da Agenda 21, em que possui como objetivo assegurar que se mantenha uma oferta adequada de água de boa qualidade para toda a população do planeta, utilizando-se de tecnologias inovadoras, inclusive o aperfeiçoamento de tecnologias nativas, necessárias para aproveitar plenamente os recursos hídricos limitados e protegê-los da poluição.

Diante deste contexto, observa-se a necessidade e importância de programas e uso de tecnologias para o gerenciamento dos recursos hídricos, de modo a garantir a proteção e uso eficiente deste bem tão valioso e vital para o futuro da humanidade.

2.2 Gestão da Demanda de Água (GDA)

A escassez de água já é uma realidade em várias regiões do mundo, não apenas nas semiáridas e, infelizmente, este processo vem se tornando mais crítico em virtude de alguns fatores, como: crescimento populacional, intensificação das atividades industriais, uso irracional dos recursos, secas, erosões do solo, desertificação, dentre outros, o que tem gerado problemas relacionados à falta de água para o atendimento das necessidades mais elementares da população (SOUZA; VIEIRA, 2004).

Para o enfrentamento desta problemática, normalmente a solução utilizada pelas políticas públicas é a busca pelo suprimento da quantidade de água demandada, dando prioridade à expansão da oferta, com a construção de obras hidráulicas (barragens, canais, adutoras etc.), alternativa que gera elevados custos ambientais, econômicos e sociais (SOARES, 2012).

Contudo, o atendimento da demanda pode ser alcançado pela conservação e uso eficiente da água, especialmente em situações que a construção e obras se tornam inconcebíveis sob o aspecto econômico e ambiental (STUDART; CAMPOS, 2001).

O Gerenciamento da Demanda de Água vem como alternativa viável para proporcionar um uso eficiente de água e assim atender às necessidades futuras por abastecimento, tendo sua eficácia comprovada na literatura, seja por meios

tecnológicos, campanhas educativas e atribuição de preço à água (FRIEDMAN et al, 2011).

Deste modo, Savenije e Van Der Zagg (2002) ressaltam que o gerenciamento da demanda é aplicado através das seguintes medidas: i) medidas estruturais, cuja redução de consumo de água é proporcionada pela adoção de alternativas tecnológicas; ii) medidas não-estruturais, embasadas em incentivos econômicos e legais à mudança de comportamento dos usuários etc.

Albuquerque (2004) pormenoriza as várias alternativas para implantação da Gestão da Demanda, em que destacamos as ações tecnológicas, que compreendem ações de medição individualizada em edifícios, instalações prediais que reduzam o consumo (aparelhos poupadores), sistemas individuais ou comunitários de captação de água da chuva, reuso de água, micro e macromedição na rede, sistemas automatizados de monitoramento e controle da rede de distribuição, entre outros.

A falta de água vem impactando no desenvolvimento e sobrevivência das populações em várias partes do mundo. Neste contexto, observa-se a urgência em promover meios de uso eficiente, pois a pressão por esse recurso torna-se cada dia mais relevante. As ações realizadas através da aplicação dos conceitos de Gerenciamento da Demanda de Água apresentam-se como um importante meio na busca pela eficiência do consumo final do usuário.

2.3 Quarta revolução industrial ou indústria 4.0

As novas tecnologias podem contribuir para tornar os processos mais eficientes e otimizados, com redução do uso de recursos naturais, de geração de resíduos e consumo de energia (MAGALHÃES; VENDRAMINI, 2018).

Em cada espaço temporal, a tecnologia vigente, resultado das descobertas e conhecimento humano, é o lastro para promover transformações em vários aspectos da sociedade.

A descoberta da energia a vapor, no final do século XVIII, e a invenção da energia elétrica, na segunda metade do século XIX, marcam, respectivamente, a primeira e a segunda revolução industrial. Mais tarde, o desenvolvimento da computação e da Internet caracteriza a Terceira Revolução Industrial, também conhecida como revolução digital (SCHNEIDER, 2019).

Na linha de evolução, Schwab (2016) menciona que o mundo está experimentando uma nova revolução tecnológica, cujas novidades envolvem inteligência artificial (IA), robótica, Internet das Coisas (IoT), veículos autônomos, impressão em 3D, nanotecnologia, biotecnologia, ciência dos materiais, armazenamento de energia e computação quântica.

Ainda segundo Schwab (2016), os sensores e vários outros meios de conectar as coisas do mundo físico às redes virtuais estão se proliferando em um ritmo impressionante, na medida em que sensores menores, mais baratos e inteligentes estão sendo instalados em casas, roupas e acessórios, cidades, redes de transportes e energia. Atualmente, bilhões de dispositivos em todo o mundo estão conectados à internet e a expectativa é que o número desses dispositivos aumente ainda mais nos próximos anos.

Gaia (2016) ressalta que o atual momento das empresas neste processo de quarta revolução industrial, também chamado de Indústria 4.0, apresenta uma capacidade enorme de aperfeiçoar a gestão dos processos produtivos do setor industrial, potencializado pela Internet das Coisas, cuja possibilidade de tudo se conectar a partir do acionamento de um sistema está promovendo alterações significativas nas indústrias.

2.4 Internet das coisas (IOT)

A Internet das Coisas (IoT - *Internet of Things*) é um novo paradigma que surgiu e está rapidamente se expandindo no cenário das telecomunicações. O conceito básico é a possibilidade de interação, por meio de uma conexão com a Internet, entre diversos dispositivos chamados de Coisas - como sensores, celulares, etiquetas, lâmpadas etc. - para alcançar objetivos comuns (ATZORI et al., 2010).

Neste mesmo sentido, Elsisí *et al.* (2021) comenta que IoT é uma tecnologia que faz a integração entre diversos componentes físicos e eletrônicos com a internet, tornando-se uma peça ativa no processamento de negócios ao compartilhar informações em tempo real.

Com alta capacidade de comunicação, os dispositivos IoT propiciam possíveis tomadas de decisões inteligentes em relação ao contexto em que os aparelhos estão inseridos, proporcionando o surgimento de ambientes inteligentes, com o

propósito de melhorar e agilizar atividades do dia a dia das pessoas em diversas áreas (VERMESAN et al., 2013).

De acordo com Gubbi *et al.* (2013), a Internet das Coisas é composta por três elementos principais: i) *Hardware*, responsável pelo processamento e armazenamento das informações, através dos sensores, atuadores etc.; ii) *Middleware*, pelo armazenamento e análise de dados e; iii) *Software* de Apresentação, ferramentas para entender, visualizar e interpretar os dados de diversas plataformas projetadas para diferentes aplicações.

Em relação aos tipos de tecnologias para conexão dos dispositivos à internet para o envio das informações coletadas, pode-se citar algumas: Bluetooth, Zigbee, Wi-Fi e LoRaWAN (ELISIARIO et al., 2022). Abaixo segue uma breve descrição de cada tecnologia.

O Bluetooth fornece uma conexão sem fio desenvolvida entre aparelhos em curta distância, operando na banda de 2.4 GHz e atualmente é o padrão aderido pela indústria de aparelhos móveis. O ZigBee utiliza o padrão LPWAN, compara-se ao Bluetooth porque ambas são comunicações de curto alcance, já que o ZigBee transmite até 100m e não é recomendado para transmissão de dados em altas quantidades. O Wi-Fi é uma tecnologia do padrão IEEE 802.11 com faixa de 2,4 e 5GHz. Necessita de dois elementos para estabelecer uma rede central, sendo um ponto de acesso e um adaptador respectivamente. A rede sem fio LoRaWAN altamente popular entre a LPWAN (Low Power Wide Area Network) utiliza uma camada física responsável pela comunicação de longa distância (SIKIMICÉ et al., 2020). o protocolo LoRa é útil para sistemas que utilizam baterias e necessitam enviar pequena quantidade de dados em intervalos curtos e em longas distâncias (DEVALAL et al., 2018).

2.5 Desenvolvimento WEB

O desenvolvimento Web é uma área da Ciência da Computação responsável pelo desenvolvimento de websites ou aplicações Web, em que o usuário acessa as informações via web browser, também chamados de navegadores, como exemplo, Google Chrome, Internet Explorer, Edge, Firefox, dentre outros. Essa área do desenvolvimento pode ser dividida em três partes principais, o *frontend*, o *backend*, e a infraestrutura (SILVA JUNIOR, 2022).

O *frontend* é a parte do desenvolvimento Web responsável pela interação do usuário com o sistema através de interface gráfica, ou seja, todo o processamento realizado pelo *backend* e envio ao navegador para ser renderizado, além de lidar com as entradas de dados realizadas pelos usuários (GUERRA, 2015).

O *backend* da aplicação refere-se a todo o código que desempenha as funções do ciclo de processamento, desde as requisições até a geração da resposta, indicando o tipo possível, seja falha ou sucesso. Também pode ser conceituado como o código responsável pela tomada de decisão (SCHVARTZ; VALVERDE, 2013).

Embora o *frontend* e o *backend* abordem aspectos diferentes do sistema web, a interação eficiente entre ambos, demonstrando um comportamento único, é fundamental para tornar a aplicação viável.

2.6 Sistemas inteligentes de medição

Diante deste cenário de uso massivo de informações produzidas por sensores, decorrentes da quarta revolução industrial, apresenta-se inúmeros meios de controle e monitoramento de dados de interesse em prol do gerenciamento de recursos, através do uso do IoT (*Internet of Things*) e aplicações web.

Assim, visando a otimização dos serviços, uma tecnologia em desenvolvimento, denominada como medição inteligente, está sendo utilizada. Os medidores inteligentes, ou “*smart meters*” facilitaram a disponibilidade de dados detalhados sobre os padrões de consumo dos usuários, tanto de água, quanto de energia elétrica e gás (OTAKI; HONDA; UEDA, 2020).

Segundo Oliveira et al. (2018), vários aplicativos automatizados, utilizando o conceito de Internet das Coisas, estão sendo desenvolvidos para monitoramento, gerenciamento e controle. Devido a evolução das características, como preços e tamanhos, os sensores e atuadores estão se tornando mais acessíveis para uso em diversas áreas. Os dados capturados são convertidos em informações inteligentes, contribuindo para tornar o ambiente inteligente.

Em relação ao uso para medição de água, os hidrômetros inteligentes consistem em um hidrômetro volumétrico ou a jato, ao qual se acopla um ímã à parte mecânica que gira no seu interior, denominada rotor, hélice ou turbina. Ao entrar no medidor, o fluido é direcionado em um ou mais jatos e aciona a turbina,

gerando movimentos de rotação. A cada giro, o totalizador faz registros proporcionais à rotação da turbina, acumulando e indicando o volume em metro cúbico ou litros. Dessa forma, cada giro da turbina equivale a um volume de líquido que atravessou o medidor (MARTINS, 2021).

Ainda de acordo com Martins (2021), ao acoplar um ímã à turbina, a cada giro desta, se produz um pulso elétrico. Cada pulso equivale a um volume de líquido que atravessou o medidor, em metro cúbico ou litros. A geração de pulsos elétricos permite que a leitura do medidor, que antes só podia ser acessada localmente, seja armazenada, transmitida e processada remotamente.

Há um crescimento exponencial de uso devido aos benefícios desse método, quando comparado com o método tradicional, tanto na forma de cobrar pelo que foi consumido, quanto na facilidade de detecção de perda de água para o usuário e em tempo real, gerando economia para a concessionária e para o cliente (STEWART et al., 2018).

Assim, observa-se que os sistemas inteligentes de medição desempenham um papel importante no monitoramento mais eficaz do consumo de água, pois o acompanhamento e conhecimento mais preciso da quantidade de insumo consumido em um determinado ponto é um fator preponderante para o entendimento do sistema e identificação das características e padrões de consumo, dado fundamental para alertar possíveis anomalias no sistema.

3 METODOLOGIA

A classificação do trabalho quanto à natureza, aos fins, procedimentos técnicos e à forma de abordagem problemática foi determinada considerando os aspectos abaixo detalhados.

O presente trabalho é de natureza aplicada, assim classificada por desenvolver um sistema web com recebimento de dados via protótipo de baixo custo a partir de componentes existentes, ou seja, gerando conhecimento para solução de problemas específicos. Possui fins exploratórios, uma vez que objetiva familiarizar-se com o problema, objeto do trabalho, para permitir a construção de hipóteses ou tornar o tema mais claro, no caso em questão, referente à identificação de anomalias no consumo de água. Também é descritiva, já que foi realizado o estudo dos assuntos relacionados a área do desenvolvimento Web e Internet das Coisas aplicado a monitoramento de dados.

Com relação aos procedimentos do trabalho, classifica-se como experimental, pois utilizou parte de trabalhos semelhantes disponíveis na literatura para o desenvolvimento do protótipo (experimental).

A metodologia proposta neste trabalho foi dividida em duas etapas principais: Etapa I - desenvolvimento do sistema web para apresentação dos dados, e Etapa II - desenvolvimento do hidrômetro, utilizando conceitos do IoT. Na Figura 1 estão ilustradas as etapas metodológicas da pesquisa.

Figura 1 - Etapas metodológicas



Fonte: Elaborada pelo autor, 2023.

3.1 Etapa I - Desenvolvimento do sistema web

O objetivo desta etapa é o desenvolvimento de um sistema web baseado nas técnicas e boas práticas da Engenharia de *Software*, abordando os aspectos da produção de um *software*, desde os estágios iniciais até a etapa de uso e manutenção. Abaixo, seguem as etapas metodológicas produzidas para o desenvolvimento do *software*.

3.1.1 – Engenharia de Requisitos

Aplicou-se os conceitos de Engenharia de *Software* apresentados por Sommerville (2011), em que aborda descrições do que o sistema deve fazer e oferecer, além das restrições a seu funcionamento. Os requisitos representam as necessidades, desejos, demanda do cliente para uma finalidade específica. O processo de descobrir, analisar e verificar essas informações é chamado de Engenharia de Requisitos.

Neste sentido, buscou-se determinar os requisitos do sistema através de duas abordagens: requisitos funcionais e não funcionais.

Os requisitos funcionais abrangem o que o sistema deve fornecer e como deve se comportar mediante a entrada de informações. Para a determinação dos requisitos funcionais utilizou-se da observação de outros softwares similares disponíveis na internet, extraindo informações de comportamento, organização e dados a serem manipulados.

De maneira geral, as principais funcionalidades mapeadas nesta fase foram os CRUD (*Create, Read, Update, Delete*), que significa acrônimo para as operações de criar, ler, atualizar e deletar dados no processo de armazenamento de informações.

Dentro dos requisitos não funcionais, buscou-se introduzir os requisitos relacionados ao uso do sistema, sob os aspectos de usabilidade, compatibilidade, responsividade, manutenibilidade, disponibilidade.

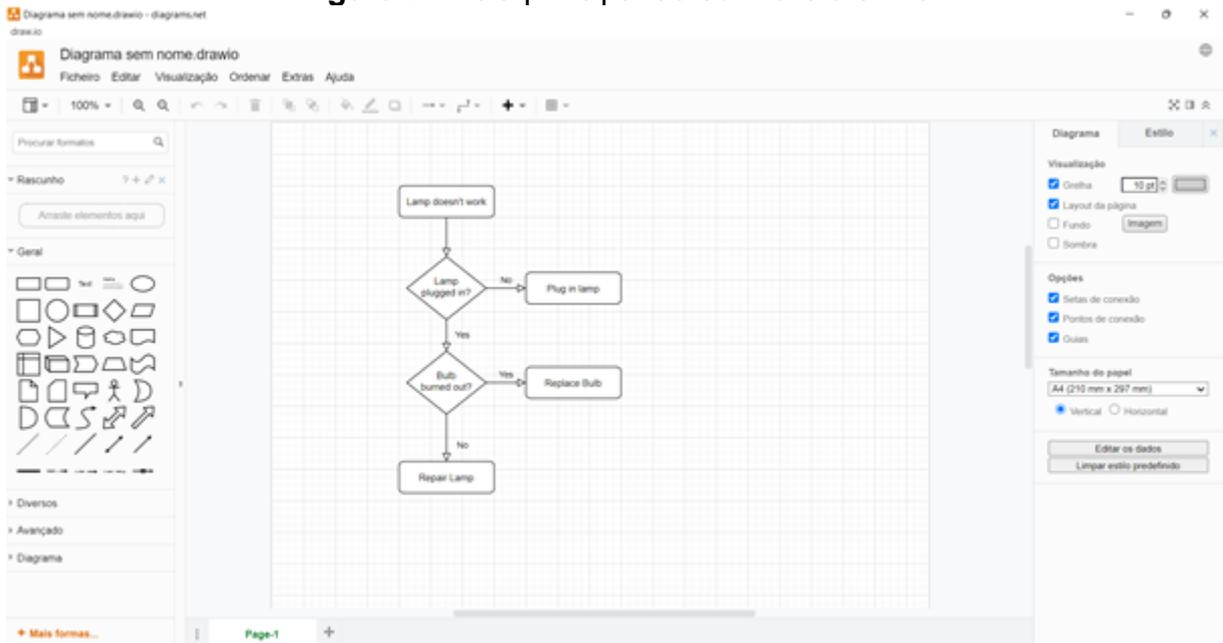
3.1.2 – Diagrama de Caso de Uso

O Diagrama de Caso de Uso é o diagrama que apresenta uma ideia geral de como o sistema irá se comportar, identificando os atores (envolvidos) e as funcionalidades disponíveis para os atores, ou seja, torna-se uma importante ferramenta para o levantamento dos requisitos funcionais do sistema.

Os atores podem ser usuários e/ou outros sistemas que interagem com o sistema em análise, sendo representado por um boneco do tipo palito (*stick man*). Os casos de uso são as representações de ações executadas pelo sistema, ilustrados através de elipses com o nome do caso de uso dentro do elemento.

Para a elaboração do diagrama de caso de uso foi utilizado o *software* de desenho gráfico draw.io, conforme Figura 2, por ser uma ferramenta gratuita e de código aberto, além da simplicidade de manipulação de elementos em tela.

Figura 2 - Tela principal do *software* draw.io



Fonte: Elaborada pelo autor, 2023.

3.1.3 – Diagrama Entidade Relacionamento

O Diagrama Entidade Relacionamento é um modelo utilizado para auxiliar no desenvolvimento do banco de dados, em que ilustra as entidades, demonstrando os relacionamentos entre cada uma delas, além da cardinalidade envolvida na relação. Exemplos de entidades são pessoas, lojas, objetos etc.

São utilizados símbolos, como retângulos e linhas para representar a interconectividade entre as entidades, relacionamentos e atributos. Assim sendo, essa modelagem torna o processo de definição do banco de dados mais claro e ilustrativo para o desenvolvedor.

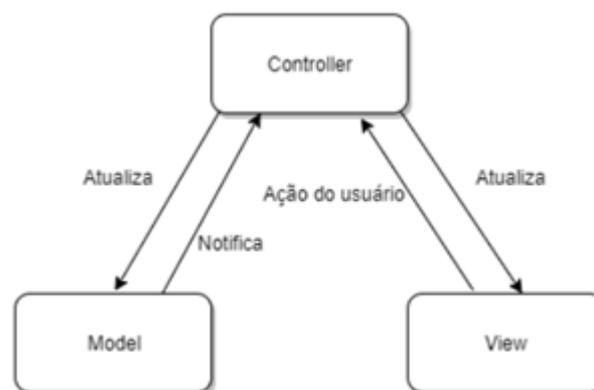
Da mesma forma, como a anterior, utilizou-se o *software* draw.io na elaboração do diagrama em questão.

3.1.4 – Arquitetura MVC

O padrão arquitetural *Model-View-Controller* (MVC) é focado no reuso de código e na separação de funcionalidades em três camadas interconectadas, no qual, cada uma é responsável por uma característica específica. Uma camada responsável pela apresentação dos dados, em que o usuário interage com o sistema (*View*), uma para a modelagem dos dados (*Model*) e outra para a manipulação desses dados (*Controller*) (IVANOVICH et al., 2019).

Neste contexto, o padrão *Model-View-Controller* (MVC) tem como objetivo separar a interface de usuário da parte lógica de negócio, sendo projetado para ser utilizado em aplicações focadas na interação com o usuário, voltadas para interfaces gráficas. Pode-se observar o esquema do padrão MVC na Figura 3.

Figura 3 - Diagrama do Model-View-Controller



Fonte: KEARNEY, 2019.

O modelo MVC tem diversas vantagens que o tornam um padrão de desenvolvimento de *software* eficaz, tendo destaque pela facilidade de manter,

testar e atualizar, principalmente em sistemas compostos e de pequeno porte. A aplicação se torna escalável, facilita o reuso de código, melhora o nível de sustentabilidade e, devido a sua arquitetura modular, permite que módulos distintos do sistema sejam criados simultaneamente durante sua criação (BARBOSA, 2022).

No projeto em análise, utilizamos o padrão MVC como abordagem para o contexto do software, utilizando um arquivo separado, composto da camada *View* para a parte do *frontend* (interface do usuário) e outro arquivo compreendendo o *Model* e *Controller* para a parte do *backend*.

3.1.5 – Tecnologias de uso comum do frontend e backend

3.1.5.1 Visual Studio Code

O *Visual Studio Code* (VSCode) é um editor de código gratuito e aberto, desenvolvido pela Microsoft para os principais sistemas operacionais Windows, Linux e macOS. Possibilita o desenvolvimento em diversas linguagens, contando com recursos de depuração, realce de sintaxe, complementação inteligente de código, *snippets* e refatoração de código, e ainda possui o software Git incorporado (RODRIGUES, 2021).

O VSCode, imagem mostrada na Figura 4, foi escolhido como única ferramenta para escrita do código-fonte em virtude das vantagens apresentadas, sendo adicionado plugins para suplementar funcionalidades necessárias de cada tecnologia usada.

Figura 4 - Logo do visual *studio code*



Fonte: <https://code.visualstudio.com>

3.1.5.2 Git e Github

O Git é um sistema de controle de versão de arquivos, em que possibilita manter um histórico de modificações dos códigos, facilitando a recuperação de estados anteriores, bem como a comparação de partes modificadas ao longo de marcos específicos.

O Github é uma plataforma de hospedagem de código-fonte e desenvolvimento de código colaborativo, baseado no sistema de controle de versão Git. Possui como principais funcionalidades: armazenamento e versionamento de código, *issue tracking*, gerenciamento de projetos, revisão colaborativa e rede de interação social (KALLIAMVAKOU, 2014).

O Git e Github, representados na Figura 5, foram escolhidos como ferramentas auxiliares no desenvolvimento deste projeto, pois são ambientes consolidados e difundidos para versionamento e armazenamento de códigos, além do aspecto de *open-source*.

Figura 5 - Logo do Git e Github



Fonte: <https://git-scm.com/> e <https://github.com/>

3.1.6 – Tecnologias usadas no desenvolvimento frontend

Para o desenvolvimento do *frontend* foi utilizado o *React JS* (Figura 6), que, de acordo com Rodrigues (2021), consiste em uma biblioteca *JavaScript* gratuita e de código aberto, mantida pelo Facebook e utilizada no desenvolvimento de interfaces de usuário em aplicações web. Possui módulos poderosos que trabalham com o conceito de componentes e seus estados, possibilitando atualizar e renderizar de forma eficiente apenas os componentes necessários, na medida em que os dados mudam (alteração de estado). Atualmente, o *React JS* é uma das bibliotecas

Javascript mais utilizadas no mundo, trazendo performance, eficiência, flexibilidade, entre outros benefícios, além de possibilitar a criação de interações complexas.

Para o uso adequado do ReactJS na aplicação deste estudo, foi necessário a instalação das seguintes bibliotecas principais: *axios*, *react-router-dom*, *i18next*, *react-flatpickr*, *react-icons*, *react-toastify*, *apexchart*, *react-redux*, *redux-saga*.

Em relação ao padrão de *design* utilizado na aplicação, fez-se uso de uma biblioteca de componentes React, denominada de Material UI, em que implementa o *design* do Google.

Figura 6 - Logo do React



Fonte: <https://pt-br.reactjs.org/>

3.1.7 – Tecnologias usadas no desenvolvimento backend

3.1.7.1 Spring boot

O *Spring Framework* é uma ferramenta desenvolvida para a plataforma Java, amplamente utilizada no desenvolvimento de sistemas modernos, possuindo um ecossistema de tecnologias robustas, que simplificam muitas tarefas realizadas pelos programadores, aumentando assim a sua produtividade e a fluência de seus códigos. Com o *Spring*, pode-se construir aplicações de alto desempenho, utilizando os diversos módulos que o integram, de acordo com as necessidades do projeto, como: *Spring Web*, *Spring MVC*, *Spring Security*, *Spring Data JPA*, *Spring Cloud*, entre outros (RODRIGUES, 2021).

Ainda segundo Rodrigues (2021), como forma de melhorar ainda mais o processo de criação das aplicações, foi desenvolvido o módulo *Spring Boot*, o qual permite que o desenvolvedor foque mais na execução do projeto, do que na etapa de configurações, realizando a abstração de toda a sua complexidade, sendo muito utilizado para desenvolver aplicações com estilo arquitetural *REST*, por meio de seu

módulo *web*, que concentra os recursos necessários para criação e exposição de serviços remotos. Destaca-se na Figura 7 o logo e site do *Spring*.

Neste contexto, foi utilizado o *VSCode* para o projeto em *Spring Boot*, sendo necessária a instalação de algumas extensões, de modo a adicionar recursos específicos a IDE, como: *Extension Pack Java*, *Spring boot extension pack*, *Spring boot dashboard* e *lombok*.

Figura 7 - Logo do *Spring Boot*



Fonte: <https://spring.io>

3.1.7.2 *Postman*

Postman é uma aplicação que permite realizar requisições HTTP, facilitando o teste e depuração de serviços REST. Proporciona também o processo de validação dos serviços *web*, através de requisições HTTP aos *endpoint* criados, utilizando JSON e o protocolo HTTP para troca de informações entre as requisições. Os resultados esperados apresentam como respostas os códigos de status de sucesso da classe 2xx (SANTOS, 2018).

Desta forma, durante o desenvolvimento do *backend*, utilizou-se o *Postman*, Figura 8, na realização dos testes com a aplicação do cliente consumindo a API implementada no *Spring Boot*.

Figura 8 - Logo do *Postman*



Fonte: <https://postman.com>

3.1.7.3 H2 Database

O banco de dados utilizado no projeto foi o H2 *database*, um banco relacional em memória, de código aberto, compatível com outros banco de dados, cujo funcionamento ocorre apenas durante a execução da aplicação, não realizando a persistência dos dados em disco. O *Spring Boot* fornece integração embutida e não exige instalação, tornando-o conveniente para testes em aplicações. Na Figura 9 apresenta-se o logotipo do H2.

Figura 9 - Logo do H2 *database*



Fonte: <http://h2database.com/>

3.1.7.4 API REST

O conceito de API (*Application Programming Interface*) está relacionado a uma maneira de comunicação entre sistemas, permitindo a integração entre ambos, em que um deles fornece informações e serviços que podem ser utilizados pelo

outro, sem a necessidade de o sistema que consome a API conhecer detalhes de implementação do *software* (PEREIRA, 2019).

Desta forma, no contexto de aplicações *web*, tem-se apenas um *backend* fornecendo informações para inúmeros consumidores *frontend*, o que permite uma independência entre camadas e separação de funcionalidades.

Assim sendo, de modo a melhor conceituar API, apresenta-se abaixo uma descrição comparativa e didática utilizando exemplo do cotidiano:

[...] API é como um garçom de um restaurante. O cliente, neste caso a aplicação que deseja receber os serviços, recebe do garçom o menu com todos os itens daquele restaurante. Ao escolher uma opção o garçom leva este pedido até a cozinha, aplicação da API, onde por sua vez os cozinheiros, que são os serviços compartilhados pela aplicação, realizam o pedido como foi descrito pelo cliente. Ao concluir o pedido o cozinheiro avisa o garçom, este por sua vez entrega o pedido ao cliente completando o processo de exemplificação uma requisição de API. (GOUVEIA, 2016).

Há algumas formas de interação entre cliente e API, sendo uma das mais utilizadas denominada de arquitetura *REST* (*Representational State Transfer*), na qual seu criador, Fielding (2000), define como sendo um conjunto de *constraints* (restrições ou princípios) com objetivo de definir a melhor forma de particionar um sistema, a maneira como os componentes se identificam e se comunicam entre si, como as informações são comunicadas, como os elementos podem evoluir de forma independente.

Em interações utilizando a internet, aplica-se o protocolo HTTP e suas requisições através da URL, em que correspondem aos chamados *web services*, ou traduzido, serviços *web*, utilizando principalmente *XML* ou *JSON* como formato de comunicação (GALINDO JUNIOR et al, 2021).

Ainda segundo Galindo Júnior et al (2021), com a determinação das URI dos recursos, é possível consumir as ações sobre eles, em *Web Services*, seu consumo é feito através do protocolo HTTP, de forma mais específica, através dos verbos, sendo os mais utilizados: *GET*, *POST*, *PUT* e *DELETE*.

O projeto foi moldado seguindo os conceitos de API e da arquitetura *REST*, permitindo comunicação entre aplicações sem a dependência do sistema operacional e a linguagem de programação.

3.2 Etapa II - Desenvolvimento do hidrômetro inteligente IoT

O sistema é composto de um hidrômetro inteligente IoT dividido em dois componentes básicos: *hardware* (parte física dos computadores) e *software* (sequência de instruções a serem executadas).

3.2.1 – Hardware

O *hardware* é composto por um sensor de vazão, um microcontrolador (ESP8266 NodeMCU), placa *protoboard*, *jumpers*, carregador de energia e caixa de proteção em PVC. Abaixo, observa-se a descrição detalhada dos componentes utilizados no projeto.

3.2.1.1 Microcontrolador

O microcontrolador utilizado foi o NodeMCU – ESP8266, dispositivo eletrônico programável, que permite a obtenção, tratamento e disponibilização de dados ou ações de acordo com a programação desenvolvida e sensores utilizados. O NodeMCU possui uma placa da família ESP8266, possibilitando a conexão com a internet via WiFi e sensores. Possui comunicação via cabo micro-usb, com antena embutida e conector micro-usb para conexão ao computador, além de 11 pinos de I/O e conversor analógico-digital, conforme Figura 10. A linguagem e o ambiente de programação (IDE) são os mesmos do Arduino.

Figura 10 - NodeMCU Esp-8266



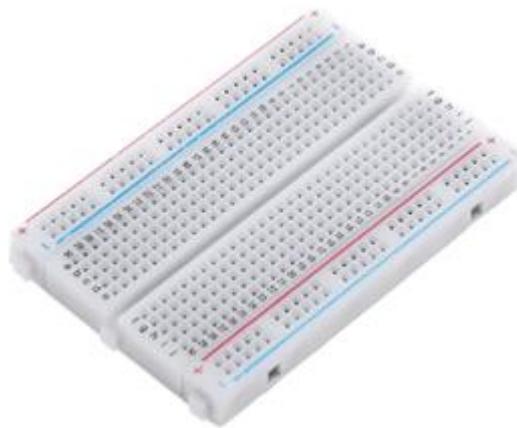
Fonte: <https://www.armazemdaeletrica.com.br/>

Destaca-se que foi utilizado o protocolo de transmissão de dados através das redes sem fio (*wifi*), pois o escopo e proposta do protótipo limita-se a ambientes com acesso a internet e com disponibilidade de energia elétrica, como por exemplo, laboratórios, residências, comércio etc. Sendo assim, outros sistemas (Lora) tornam-se dispensáveis em virtude da finalidade e requisitos exigidos no projeto.

3.2.1.2 Placa protoboard

Para a montagem e conexão dos componentes foi utilizada a placa *protoboard*, pois é muito útil para o desenvolvimento de projetos com componentes eletrônicos em fase de prototipagem. Possui furos e conexões internas para montagem de circuitos, possibilitando conectar diversos componentes através de ligações simples, bastando apenas fazer ligações por meio dos orifícios presentes na placa, o que dispensa o uso de solda para conectar os circuitos. Diante de poucos componentes utilizados no projeto, apenas o microcontrolador e sensor de fluxo, a placa utilizada foi a composta de 830 furos, conforme Figura 11.

Figura 11 - Protoboard



Fonte: <https://www.casadarobotica.com/>

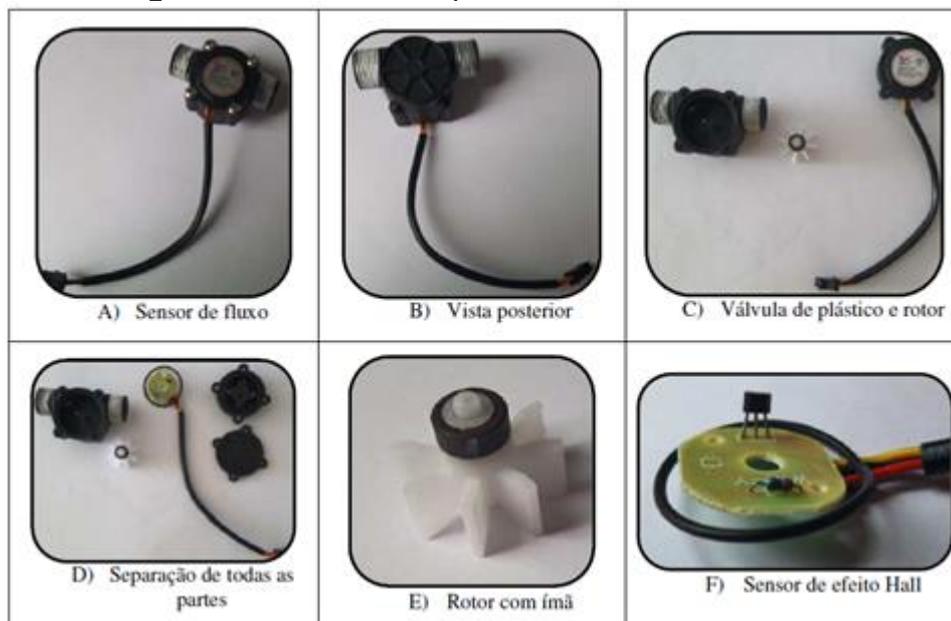
3.2.1.3 Sensor de fluxo

O sensor de fluxo é o equipamento destinado a medir a quantidade de fluído que atravessa determinada seção em certa quantidade de tempo. Para a obtenção

desta variável foi escolhido e utilizado o modelo YF-201, pois é um componente de fácil obtenção, baixo custo e largamente utilizado em projetos similares.

Esse dispositivo é composto por uma válvula de plástico, um rotor de água e um sensor de efeito *Hall* (Figura 12C). Quando a água passa pelo rotor, faz girar e a velocidade muda de acordo com a taxa de fluxo. Acoplado ao rotor existe um ímã (Figura 12E), e ao girar, o sensor de efeito Hall (Figura 12F) detecta esse movimento e envia a quantidade de rotações em forma de pulsos para o microcontrolador (BRANDÃO, 2022).

Figura 12 - Partes componentes do sensor de fluxo



Fonte: BRANDÃO, 2022.

A instalação do sensor de fluxo (Figura 13) é realizada conectando o equipamento na tubulação de água, geralmente de PVC, com diâmetro de 20mm. Em algumas situações faz-se necessário o uso de conexões para compatibilização de diferenças de diâmetro entre peças, bem como válvulas de retenção e registros.

O mecanismo básico de contabilização da vazão ocorre através do envio dos pulsos elétricos para o NodeMCU, que converte essa informação em vazão, mediante programação desenvolvida.

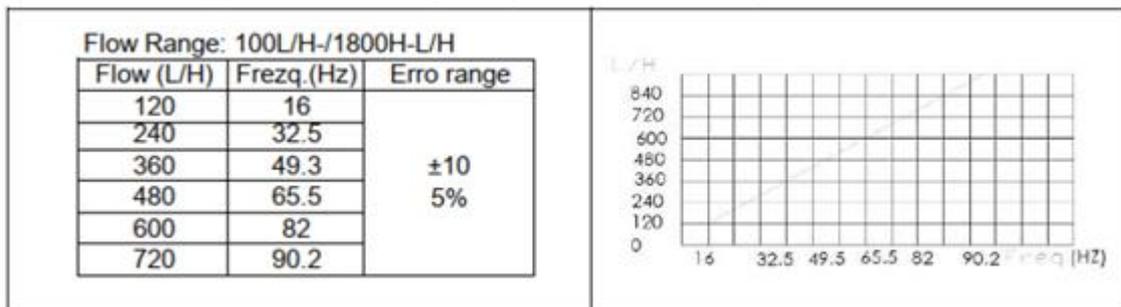
Figura 13 - Sensor de fluxo



Fonte: <https://www.casadarobotica.com>

Dentre as principais características técnicas do sensor de fluxo, tem-se: possui três saídas (positiva, terra e analógica), faixa de limite de fluxo operacional de 1 a 30 L/min, pressão máxima de água de 2MPa. A frequência em Hertz, captada pelo sensor de efeito *Hall*, pode ser convertida em vazão por segundo (Figura 14).

Figura 14 - Tabela e gráfico para conversão da frequência em vazão



Fonte: Datasheet (2022).

3.2.1.4 Jumpers

Para realizar a conexão do sensor de fluxo com o microcontrolador via placa *protoboard*, utilizou-se os *jumpers* com essa finalidade. Os *jumpers*, utilizados neste projeto, foram do tipo conector macho-macho, com revestimento de isolamento em PVC, conforme mostrado na Figura 15.

Figura 15 - Jumpers



Fonte: <https://www.makehero.com/produto/jumpers-macho-macho-x40-unidades/>

3.2.1.5 Caixa protetora

Utilizou-se neste projeto uma caixa termoplástica, apresentada na figura 16, para proporcionar segurança e proteção contra as intempéries do meio, em situações potenciais como: interferência externa, proteção dos usuários contra descargas elétricas, roubo, chuva, sol, corrosão etc.

Figura 16 - Caixa de proteção



Fonte: <https://www.zcmateriais.com/produto/caixa-passagem-sobrepor-152x109x67-pvc-cinza-soprano--5038-2003-01/4494-0>

3.2.1.6 Carregador com cabo micro USB

Para alimentação elétrica do equipamento e transferência da codificação para o microcontrolador, utilizou-se carregador convencional com cabo micro USB, de acordo como mostra a Figura 17.

Figura 17 - Carregador com cabo micro USB



Fonte: https://www.ferrovia.com.br/kit-carregador-galaxy-j7-neo-cabo-micro-usb-v8-branco-kingo-sm-j701mt_1566843609

3.2.2 - Software

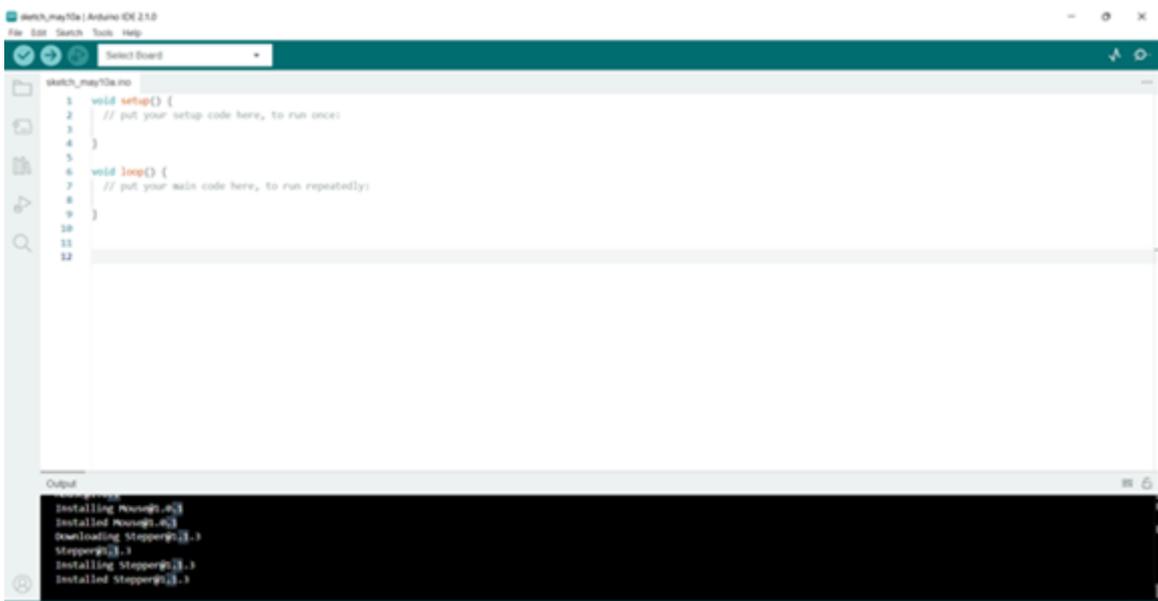
O *software* consiste na codificação do microcontrolador para processamento dos dados do sensor e envio das informações para o sistema *web*. A linguagem utilizada e a IDE (*Integrated Development Environment*) são as mesmas da plataforma do Arduino, cuja linguagem de programação é baseada na linguagem C, contendo bibliotecas específicas para interação com os sensores utilizados.

O ambiente de desenvolvimento da IDE Arduino, mostrado na Figura 18, é bastante intuitivo e fácil de entendimento, possibilitando diversos procedimentos de forma simples e direta. Na parte superior, tem-se uma barra com botões de verificação e *upload* do código para o microcontrolador e processos de *debug* do

algoritmo. Na parte central está o local de inserção das linhas de programação, no qual divide-se em duas funções principais: *setup* e *loop*.

Na função *setup* são inseridas as configurações do projeto em que serão executadas pelo microcontrolador apenas uma única vez, por exemplo, inicialização da velocidade do *serial*, apresentação de alguma mensagem inicial em *display* etc. Na função *loop* é inserida a programação principal, em que o programa ficará repetindo continuamente o algoritmo presente neste trecho de código.

Figura 18 – Ambiente de programação IDE Arduino



Fonte: Elaborada pelo autor, 2023.

4 RESULTADOS E DISCUSSÕES

Os resultados aqui apresentados estão discutidos em dois assuntos principais: i) desenvolvimento do sistema *web*; ii) desenvolvimento do hidrômetro IoT.

4.1 Desenvolvimento do sistema *web*

4.1.1 – Engenharia de Requisitos

Diante das observações e análises das principais necessidades inerentes ao sistema de monitoramento de água, foram destacados os requisitos funcionais e não funcionais do sistema, em que são detalhados e registrados abaixo.

Ressalta-se que a proposta do projeto é apresentar um sistema simples e básico, com o objetivo de demonstrar as potencialidades de uma solução nesse contexto, que pode ter um impacto positivo no gerenciamento e controle de recursos em geral.

Assim, os requisitos funcionais foram compreendidos da parte de CRUD (*Create, Read, Update, Delete*) para o cadastramento de um ponto de monitoramento, inserção e apresentação de valores de consumo deste ponto, conforme detalhado no Quadro 1.

Quadro 1 - Requisitos funcionais

| ID | REQUISITOS | DESCRIÇÃO |
|-----|--|--|
| RF1 | Criar, ler, atualizar e excluir pontos de monitoramento. | Cadastramento de ponto de consumo, especificando nome, atividade, descrição, área, número de consumidores, endereço. |
| RF2 | Inserir dados de consumo de água. | Inserir o consumo em determinado tempo para um ponto de monitoramento. |
| RF3 | Detalhar dados de consumo do ponto monitorado. | Mostrar com detalhes o consumo de água. |
| RF4 | Selecionar período para detalhamento do consumo de água. | Escolher as datas de início e fim do período de apresentação das informações. |
| RF5 | Selecionar o grau de detalhamento do consumo. | Escolher o grau de detalhamento da informação do consumo de água em dia, mês ou ano. |

| | | |
|------|--------------------------------------|---|
| RF6 | Gerar gráfico com o consumo de água. | Gerar gráfico de barras com os consumos de água de acordo com período e grau de detalhamento. |
| RF7 | Manipular gráfico. | Alterar tamanho, arrastar e exportar gráfico em formato de imagem. |
| RF8 | Gerar tabela com o consumo de água. | Gerar tabela com os consumos de água de acordo com período e grau de detalhamento. |
| RF9 | Manipular tabela. | Filtrar dados, ocultar colunas, alterar tamanho das células e exportar tabela |
| RF10 | Descrever e resumir os dados. | Mostrar informações consolidadas dos dados, como consumo total, médio, máximo e mínimo |

Fonte: Elaborada pelo autor, 2023.

Dentre os requisitos não funcionais, destacam-se, no Quadro 2, aqueles que mais nortearam o desenvolvimento do projeto.

Quadro 2 - Requisitos não funcionais

| ID | REQUISITOS | DESCRIÇÃO |
|------|------------------|---|
| RNF1 | Usabilidade | O sistema deve ser simples e de fácil entendimento e manuseio. |
| RNF2 | Compatibilidade | O sistema deve ser aceito pelos navegadores mais usados pelos usuários. |
| RNF3 | Responsividade | O sistema deve se adaptar ao tamanho e o layout do site para a tela do dispositivo utilizado. |
| RNF4 | Manutenibilidade | O sistema deve possuir baixo acoplamento e alta coesão. |
| RNF5 | Manutenibilidade | O sistema deve possuir tecnologias modernas, consolidadas e difundidas no mercado. |
| RNF6 | Disponibilidade | O sistema deve estar funcionando 24 horas por dia e os 7 dias da semana. |

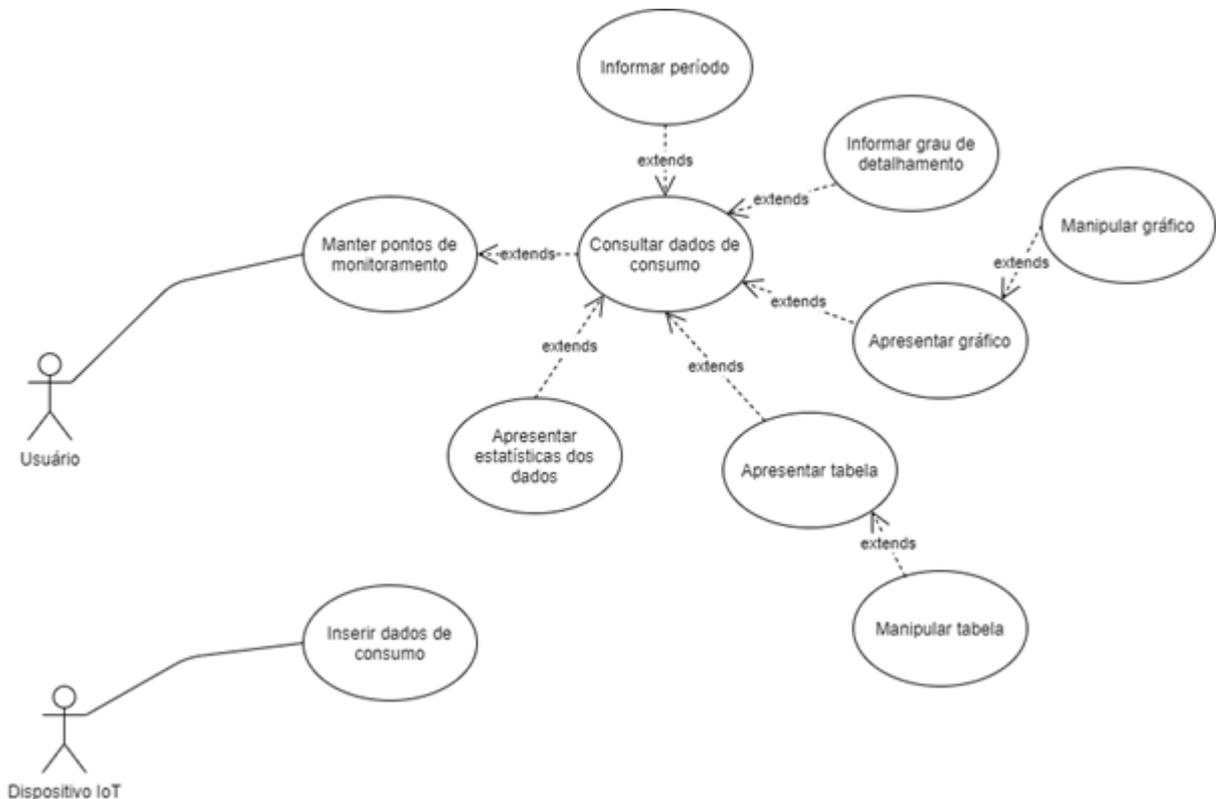
Fonte: Elaborada pelo autor, 2023.

4.1.2 – Diagrama de Casos de Uso

O caso de uso elaborado indica dois atores principais: usuário do sistema (aquele que irá cadastrar os pontos de monitoramento e visualizar os dados) e o hidrômetro IoT (que enviará as informações de consumo para o sistema).

Em relação às funcionalidades disponíveis para os atores, percebe-se que possuem relação direta com os requisitos funcionais elencados na seção 5.1.1, representando-os através de ilustrações mais didáticas de entendimento, conforme Figura 19.

Figura 19 - Diagrama de casos de uso



Fonte: Elaborada pelo autor, 2023.

4.1.3 – Diagrama de Entidade Relacionamento

O Diagrama Entidade Relacionamento, construído para o projeto, possui duas entidades simples (*consumer* e *consumption*) que representam o ponto de

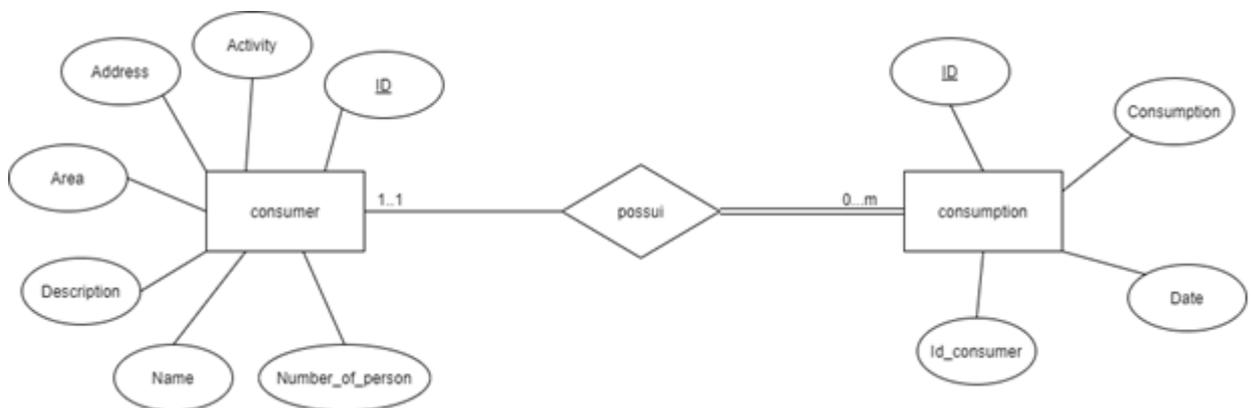
monitoramento e o consumo registrado no dispositivo IoT deste ponto, respectivamente, conforme mostrado na Figura 20.

A entidade “*consumer*” representa o ponto de monitoramento, ou seja, uma casa, banheiro, torneira etc., possui como atributos: Id (identificador único – chave primária), *activity* (atividades relacionadas ao ponto de monitoramento, ex.: sala de aula, indústria etc.), *address* (endereço do ponto de monitoramento), *area* (área em m² do espaço monitorado), *description* (descrição do ponto de monitoramento, ex.: edificação com 2 pavimentos etc.), *name* (nome do ponto de monitoramento), e *number_of_person* (número de pessoas com ligação ao ponto de consumo, ex.: reitoria com 20 pessoas).

A entidade “*consumption*” representa o consumo medido em determinado intervalo de tempo de um ponto de monitoramento, possui como atributos: Id (identificador único – chave primária), *consumption* (consumo medido), *date* (informação de data e horário de medição do consumo) e *id_consumer* (identificador do ponto de monitoramento relacionado).

Em relação a cardinalidade, destaca-se que a entidade “*consumer*” pode possuir zero ou muitos consumos, ao tempo em que a entidade “*consumption*” é possuída por apenas 1 (um) ponto de monitoramento. Desta forma, a entidade “*consumption*” recebe uma chave estrangeira, que será a chave primária da entidade “*consumer*”.

Figura 20 - Diagrama de entidade relacionamento



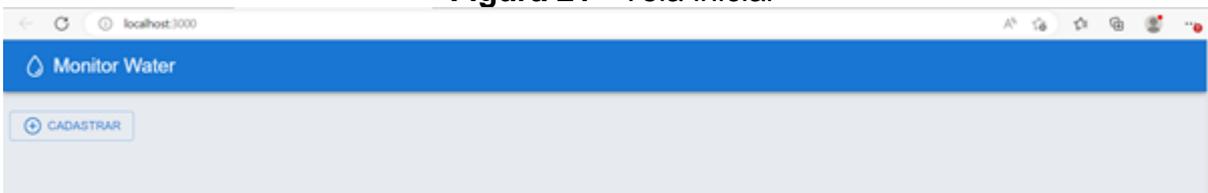
Fonte: Elaborada pelo autor, 2023.

4.1.4 – Desenvolvimento frontend

Esta seção apresenta a parte desenvolvida do *frontend*, ou seja, a parte de interface com o usuário e apresentação das informações.

A aplicação denominada Monitor *Water* possui tela inicial, conforme Figura 21, apresentando barra superior com identificação da aplicação e logotipo. No corpo da página tem-se um botão de cadastramento de um ponto de monitoramento.

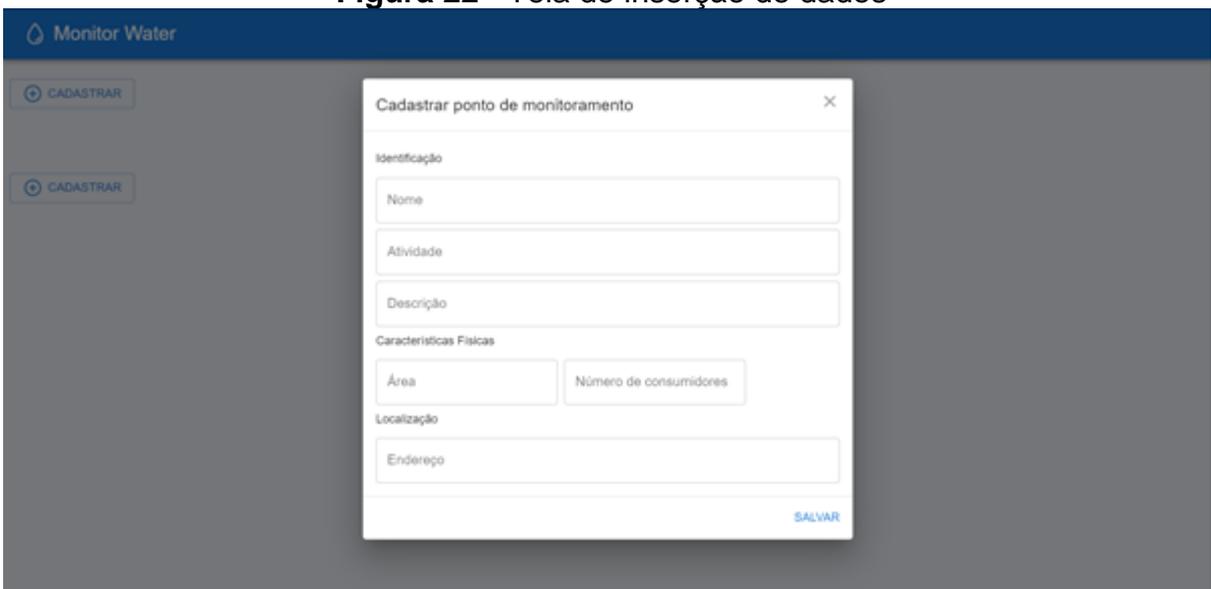
Figura 21 - Tela inicial



Fonte: Elaborada pelo autor, 2023.

Ao clicar no botão “cadastrar”, apresenta-se uma nova tela em plano principal (Figura 22), com os campos para preenchimento das informações do ponto de monitoramento. Os campos disponíveis para identificação são: nome, atividade, descrição, área, número de consumidores e endereço. Destaca-se que é criado automaticamente no banco de dados um identificador único para esse ponto de monitoramento.

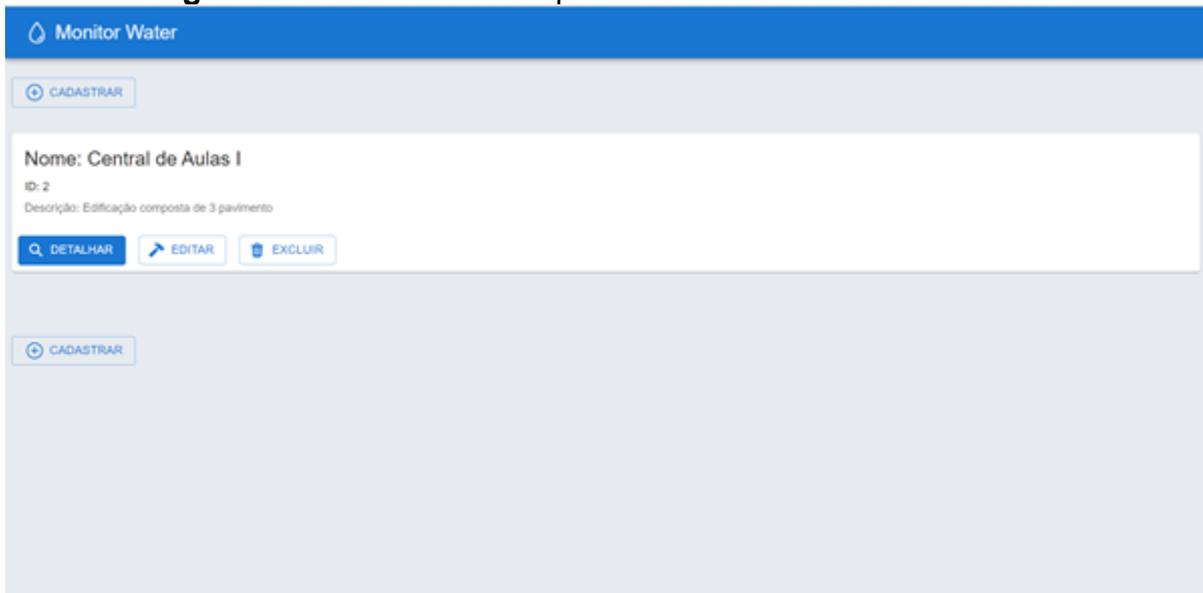
Figura 22 - Tela de inserção de dados



Fonte: Elaborada pelo autor, 2023.

Após o preenchimento dos dados e realizado o salvamento, clicando no botão salvar, é inserido na página inicial um “card”, representando o ponto de monitoramento recém-criado, com os dados de nome, ID e descrição, como ilustrado na Figura 23. Ressalta-se que o ID apresentado é o valor único e gerado automaticamente pelo banco de dados.

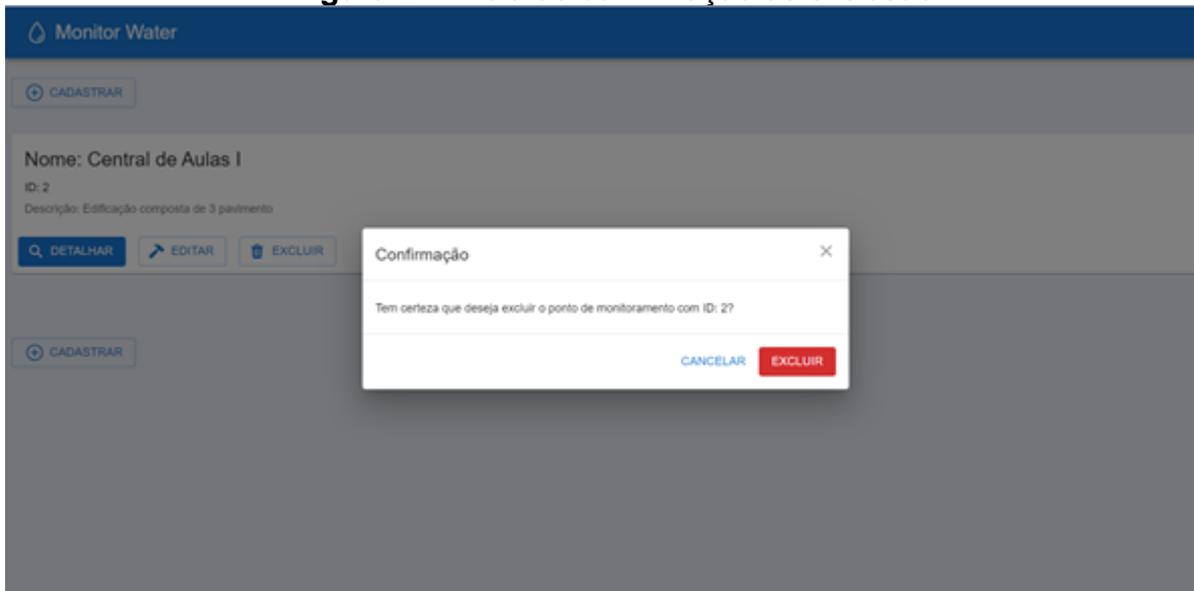
Figura 23 - Tela inicial com ponto de monitoramento cadastrado



Fonte: Elaborada pelo autor, 2023.

No ponto de monitoramento criado são disponibilizados 03 (três) botões para as funcionalidades de detalhar os dados de consumo, editar e excluir o referido ponto de monitoramento.

Acessando o botão excluir, é aberta uma janela para confirmação do processo de exclusão, de modo a alertar o usuário para uma operação crítica de eliminação de dados (Figura 24). Caso haja a confirmação, o *card* referente ao ponto de monitoramento é excluído do sistema e conseqüentemente da tela.

Figura 24 - Tela de confirmação de exclusão

Fonte: Elaborada pelo autor, 2023.

Caso o usuário clique no botão “Editar”, é mostrada uma tela (Figura 25), similar a tela de cadastramento, contendo as últimas informações salvas do ponto de monitoramento, possibilitando ao usuário alterações de quaisquer dados gravados anteriormente.

Figura 25 - Tela de edição de dados

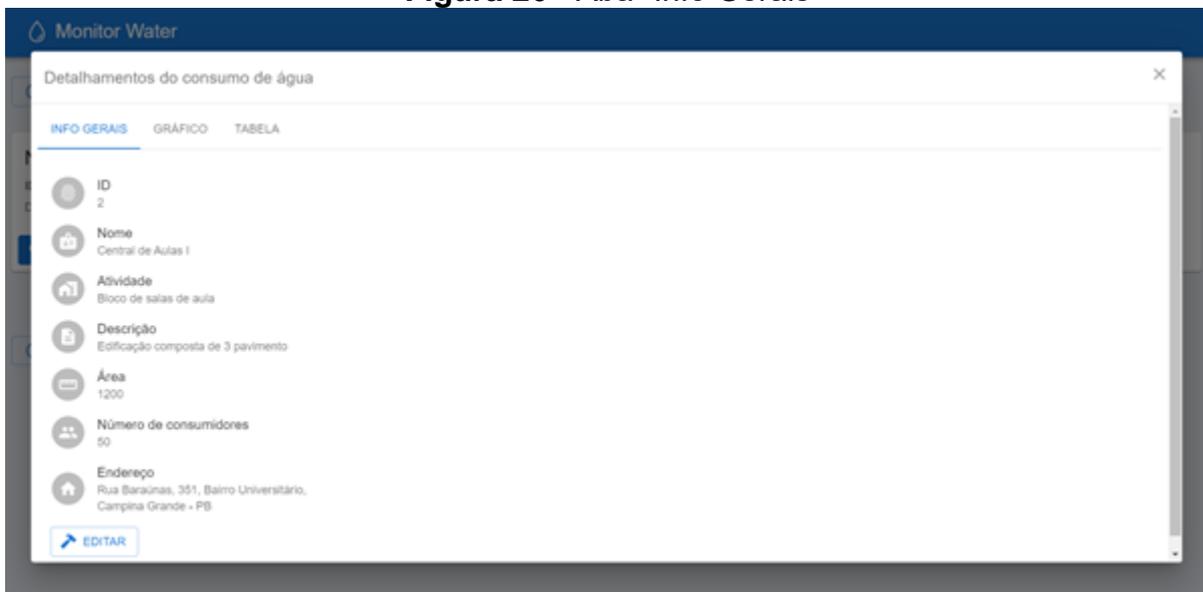
Fonte: Elaborada pelo autor, 2023.

Como outra opção do card, tem-se o botão “Detalhar”, cuja finalidade é mostrar os consumos de água registrados do ponto de monitoramento através do envio de informações do dispositivo IoT.

Assim, ao clicar, abre-se uma tela com 3 (três) abas para detalhamento do consumo de água, sendo:

- **Info Gerais:** apresenta as últimas informações cadastradas (ID, nome, atividade, descrição, área, número de consumidores e endereço). Destaca-se que nesta aba há a possibilidade de edição das informações através do botão “Editar”, que se localiza na parte inferior da janela, conforme Figura 26.

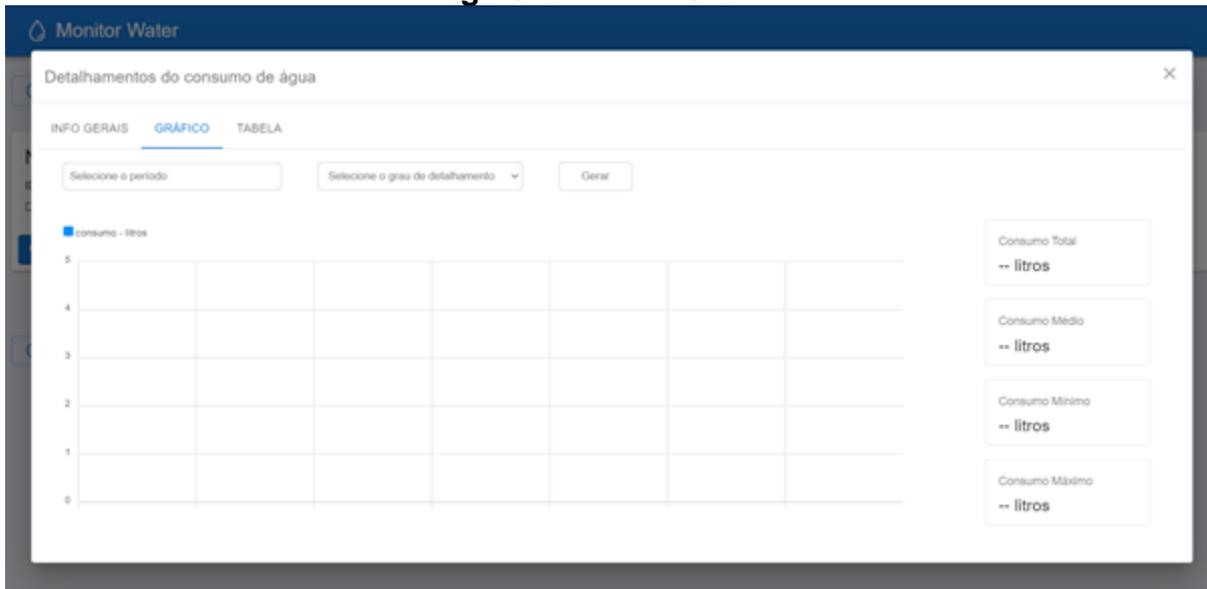
Figura 26 - Aba “Info Gerais”



Fonte: Elaborada pelo autor, 2023.

- **Gráfico:** apresenta um gráfico de barras verticais, com eixo X representando o período e eixo Y sendo o consumo em litros. Na parte superior, existem 2 (dois) componentes do tipo “select” em que possibilitam ao usuário escolher, através de uma lista de opções: i) o intervalo de tempo definido pela data inicial e final no calendário e ii) o grau de detalhamento da representação dos dados, podendo optar por dia, mês ou ano. Detalhes na Figura 27.

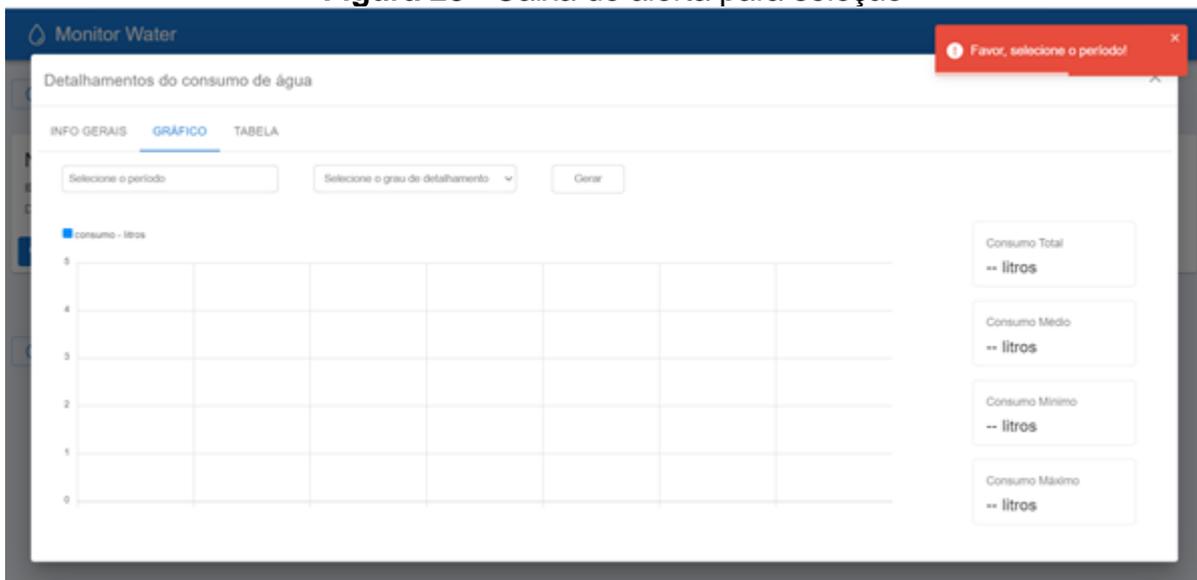
Figura 27 - Aba “Gráfico”



Fonte: Elaborada pelo autor, 2023.

É importante frisar que a aplicação torna obrigatória a seleção dos campos “período” e “grau de detalhamento” para geração do gráfico, pois caso não sejam informados e o usuário clicar no botão “Gerar”, surgirá na parte superior direita da tela um caixa de alerta dinâmico, na cor vermelha, solicitando a seleção dessas variáveis não informadas, como ilustrado na Figura 28.

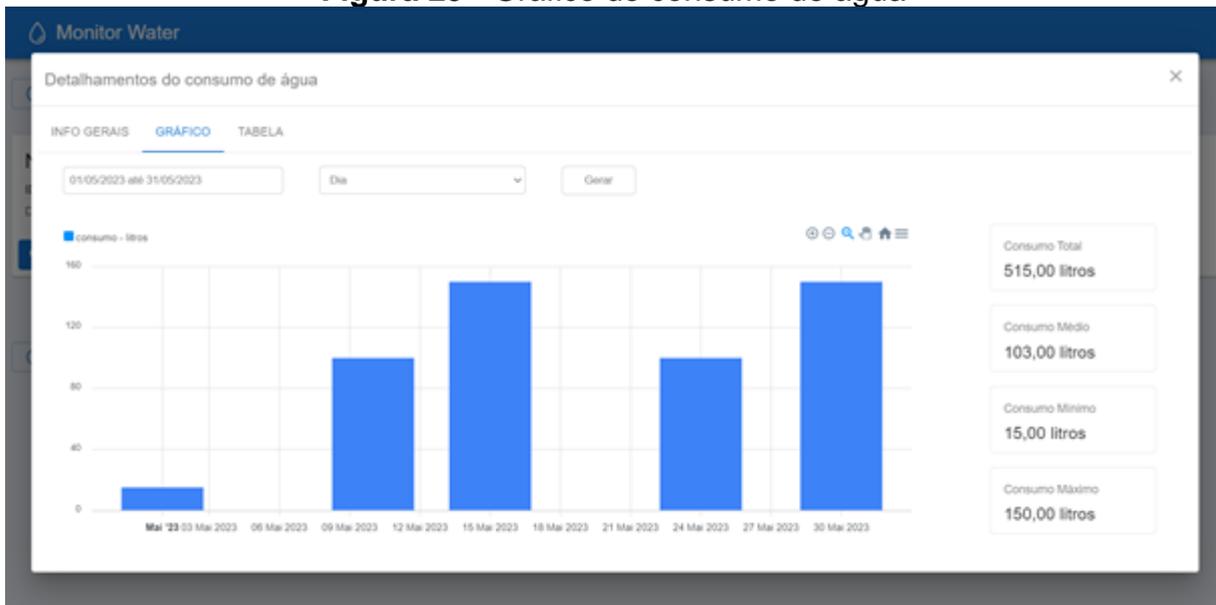
Figura 28 - Caixa de alerta para seleção



Fonte: Elaborada pelo autor, 2023.

Ao selecionar os componentes “período” e “grau de detalhamento”, e acionar o botão “Gerar”, é apresentado na parte central da tela o gráfico de detalhamento de consumo de água com os dados referentes ao intervalo de tempo selecionado e grau de detalhamento (dia, mês ou ano), de acordo com Figura 29.

Figura 29 - Gráfico de consumo de água



Fonte: Elaborada pelo autor, 2023.

Observa-se que existe a possibilidade de manipulação do gráfico, como: aumentar, diminuir ou reiniciar o zoom, arrastar lateralmente e baixar o gráfico nos formatos SVG, PNG ou CSV.

Outra funcionalidade relevante, apresentada nesta tela, é a parte de dados consolidados, localizada na lateral direita, em que exhibe o consumo total, consumo médio, consumo mínimo e consumo máximo do referido intervalo de tempo. Essas informações auxiliam o usuário na determinação de padrões de consumo do ponto monitorado e principalmente na identificação de anomalias de consumo, como faltas de água ou consumos exagerados, decorrentes de perdas e/ou desperdícios de água.

- **Tabela:** apresenta uma tabela com as colunas número, data e consumo (L). As informações disponíveis são resultantes da escolha dos campos “período” e “grau de detalhamento”. Da mesma forma como o gráfico, a seleção dos campos é obrigatória. Como ferramentas de manipulação da tabela, tem-se: localizar ou

ocultar coluna, filtrar dados, ordenar, alterar largura das linhas, exportar em formato CSV e enviar para impressão. Ver Figura 30.

Figura 30 - Tabela de consumo de água

Monitor Water

Detalhamentos do consumo de água

INFO GERAIS GRÁFICO TABELA

Selecione o período Seleção o grau de detalhamento Gerar

COLUNAS FILTROS DENSIDADE EXPORTAR

| Nº | DATA | CONSUMO (L) |
|----|------------|-------------|
| 1 | 2023-05-01 | 15,00 |
| 2 | 2023-05-10 | 100,00 |
| 3 | 2023-05-15 | 150,00 |
| 4 | 2023-05-24 | 100,00 |
| 5 | 2023-05-30 | 150,00 |

Linhas por página: 100 1-5 de 5

Fonte: Elaborada pelo autor, 2023.

4.1.5 – Desenvolvimento backend

O desenvolvimento do *backend* da aplicação foi realizado através do *framework Java Spring Boot*, em que possibilita a implementação de camadas intermediárias no lado do servidor, facilitando a coesão e manutenção da aplicação.

Utilizou-se as camadas de “*domain*”, “*controller*” e “*repository*”, tendo as seguintes responsabilidades:

- *Domain*: Camada responsável pela modelagem dos dados que servem para criação das tabelas no banco de dados relacional. Possui como anotação *spring* o “@Entity” e apresenta todos os atributos relacionados a determinada entidade. Por exemplo, na entidade “*consumer*” tem-se os atributos ID, nome, atividade, descrição, área, número de pessoas e endereço. Na Figura 31 observa-se uma parte do código da camada *domain*.

Figura 31 - Trecho do código da camada *domain*

```
@Getter
@Setter
@Entity
@Table(name = "consumer")
public class Consumer {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @JsonProperty
    private Long id;

    @Column(name = "name")
    @JsonProperty
    private String name;

    @Column(name = "activity")
    @JsonProperty
    private String activity;
}
```

Fonte: Elaborada pelo autor, 2023.

- *Controller*: Camada responsável por expor as funcionalidades (API) para os consumidores externos, ou seja, faz a interação entre o *View (frontend)* com a parte de modelagem interna dos dados (*repository* e *domain*). Possui como anotação *spring* o “@RestController”, e para cada protocolo HTTP disponível na API atribui-se uma anotação específica, como por exemplo, para requisição *GET* tem-se anotação “@GetMapping”, para *POST* tem-se anotação “@PostMapping” etc. Segue abaixo, Figura 32, um trecho da implementação da camada *controller*.

Figura 32 - Trecho do código da camada *controller*

```
@RestController
@RequestMapping("/consumer")
@CrossOrigin(origins = "**")
public class ConsumerController {

    @Autowired
    private ConsumerRepository consumerRepository;

    @GetMapping
    public List<Consumer> getConsumer() {
        return consumerRepository.findAll();
    }
}
```

Fonte: Elaborada pelo autor, 2023.

- *Repository*: Camada responsável pelo acesso e persistência de dados. Possui como anotação *Spring* o “@Repository” e utilizou-se para realizar as consultas ao banco de dados as *Query Natives*, que possibilitam o uso de consultas escritas com a linguagem SQL, padrão do banco de dados. Na Figura 33 tem-se um trecho do código da camada *repository*.

Figura 33 - Trecho do código da camada *Repository*

```

@Repository
public interface ConsumptionRepository extends JpaRepository<Consumption, Long> {

    // -----
    // Valores unitários

    @Query(value = "SELECT * FROM CONSUMPTION WHERE id_consumer = (:id_consumer) AND date >= (:minDate) AND date <= (:maxDate)", nativeQuery = true)
    List<Consumption> findAllConsumption(Long id_consumer, String minDate, String maxDate);

    @Query(value = "SELECT SUM(consumption) "
        + "FROM consumption "
        + "WHERE id_consumer = (:id_consumer) AND date >= (:minDate) AND date <= (:maxDate) ", nativeQuery = true)
    Double sumConsumption(Long id_consumer, String minDate, String maxDate);

    @Query(value = "SELECT AVG(consumption) "
        + "FROM consumption "
        + "WHERE id_consumer = (:id_consumer) AND date >= (:minDate) AND date <= (:maxDate) ", nativeQuery = true)
    Double avgConsumption(Long id_consumer, String minDate, String maxDate);
}

```

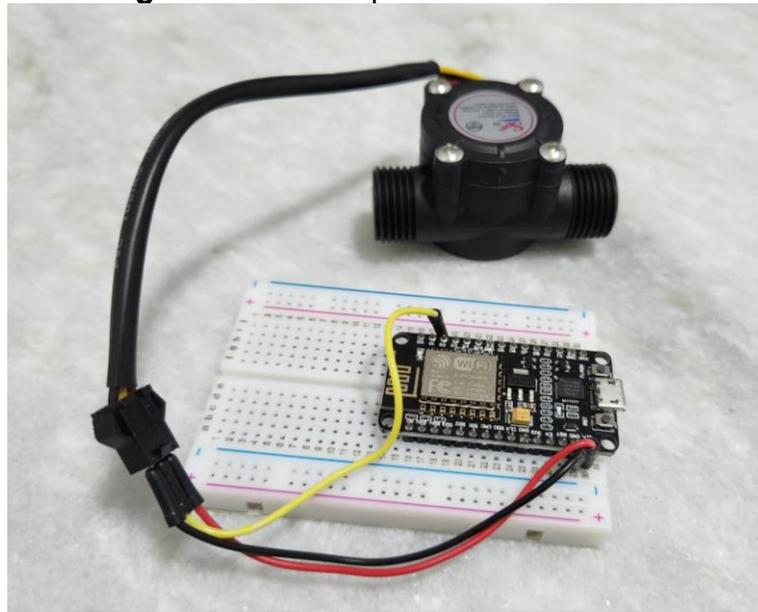
Fonte: Elaborada pelo autor, 2023.

4.2 Desenvolvimento do hidrômetro IoT

4.2.1 – Hardware

O desenvolvimento do hidrômetro IoT para a medição de consumo de água de um ponto de monitoramento é uma das etapas e propostas desse trabalho. A Figura 34 ilustra o resultado do protótipo construído.

Figura 34 - Protótipo do hidrômetro IoT



Fonte: Elaborada pelo autor, 2023.

A montagem dos componentes do hidrômetro IoT aconteceu mediante as seguintes etapas:

- a) Conectou-se o microcontrolador NodeMCU – ESP32 na *protoboard*, fazendo o encaixe dos pinos com os furos da placa;
- b) Realizou-se a conexão do sensor de fluxo aos pinos da *protoboard*, correspondentes aos pinos específicos do microcontrolador. O sensor de fluxo possui três saídas conectados ao NodeMCU: alimentação (cabo vermelho) conectada ao pino de 5V do microcontrolador; terra (cabo preto) conectado ao pino GND e a Saída PWM (cabo amarelo) conectado à porta digital D2;
- c) Realizou-se a conexão de um carregador com micro USB para alimentação de energia do sistema;
- d) Colocou-se os componentes dentro da caixa de proteção em PVC.

A instalação e funcionamento do sistema são bastante simples. Coloca-se o sensor de fluxo em linha com uma tubulação de água, utilizando algumas conexões hidráulicas. À medida que o fluxo de água vai passando na tubulação, é enviado pulsos PWM para o microcontrolador, sendo esses pulsos responsáveis pela mensuração da vazão de água circulante.

Destaca-se que é necessário um ponto de tomada para energizar os componentes e um sinal de internet sem fio (*wifi*) para envio das informações do hidrômetro IoT para a aplicação “*Monitor Water*”.

4.2.2 – Software

A codificação do microcontrolador tem como objetivo principal converter os pulsos gerados pelo sensor de fluxo e encaminhar para o sistema *web* “*Monitor Water*”. Os dados enviados referem-se às variáveis de: identificador do ponto de monitoramento, data em formato ano, mês, dia, hora e minuto e, por fim, o valor de consumo de água mensurado em determinado tempo. Observa-se abaixo, na Figura 35, o formato do envio da informação.

Figura 35 - Formato de envio da informação

```
{
  "id_consumer": 2,
  "date": "2023-05-24T09:15:55",
  "consumption": 100
}
```

Fonte: Elaborada pelo autor, 2023.

É importante destacar que o microcontrolador foi programado para enviar o consumo monitorado a cada 2 minutos, quando conectado à energia e internet.

A estrutura principal do desenvolvimento da codificação do microcontrolador pode ser apresentada nas seguintes etapas:

- Instalação das bibliotecas (*wifi*, data e horário);
- Definição das variáveis, como: nome e senha de acesso a rede de internet, URL de envio dos dados, porta de entrada do sinal do sensor, fator de calibração para converter pulsos em volume;
- Definição da função de contagem e conversão dos pulsos do sensor em consumo;
- Configuração do protocolo de envio das informações para o sistema *web*.

5 CONCLUSÃO

Considerando a escassez de água devido às mudanças climáticas, aumento da demanda e baixa eficiência dos sistemas de distribuição, além da falta de consciência da sociedade em relação ao uso do recurso hídrico, é importante aplicar conceitos que promovam o uso mais eficiente da água, como o gerenciamento da demanda.

Deste modo, de acordo com Savenije e Van Der Zagg (2002), o gerenciamento da demanda de água é implementado por meio de medidas estruturais, que envolvem a adoção de alternativas tecnológicas para redução do consumo, e medidas não-estruturais, baseadas em incentivos econômicos e legais para promover a mudança de comportamento dos usuários, entre outras.

As ferramentas tecnológicas da ciência da computação oferecem recursos valiosos para a implementação de iniciativas estruturais. Por exemplo, sistemas web, IoT com sensores e bancos de dados podem ser utilizados para melhorar a eficiência do uso da água.

No contexto deste trabalho, um dispositivo IoT com um sensor de fluxo e conectividade *wifi* foi desenvolvido para monitorar o consumo de água em um local específico. Além disso, um sistema *web* foi criado para armazenar os dados coletados e apresentá-los aos usuários de forma visualmente atraente.

Tendo isso em vista, é evidente que as ferramentas da ciência da computação desempenham um papel fundamental em diversas áreas do conhecimento, proporcionando uma integração eficaz na solução de problemas práticos do dia a dia.

REFERÊNCIAS

- ALBUQUERQUE, T. M. A. Seleção multicriterial de alternativas para o gerenciamento da demanda urbana de água na escala de bairro. Dissertação (Mestrado em Engenharia Civil e Ambiental). Unidade Acadêmica de Engenharia Civil. Universidade Federal de Campina Grande. Campina Grande. 2004.
- ATZORI, L.; LERA, A.; MORABITO, G. The Internet of things: A survey. *Computer Network*, pp. 2787-2805. 2010.
- BARBOSA, Adriley Samuel Ribeiro. Análise comparativa entre os padrões mvc, mvp, mvvm e mvi na plataforma android. Monografia. Instituto Federal Goiano – Campus Urutaí. 2022.
- BRAGA, C. F. C. Avaliação multicriterial e multidecisaria no gerenciamento da demanda urbana de água. Dissertação apresentada ao curso de mestrado em Engenharia Civil da Universidade Federal da Paraíba. 2001.
- BRANDÃO, Igor Antonio de Paiva. Dispositivo IoT de micromedição integrado com Cloud Computing para monitoramento do consumo de água. Dissertação (Pós-Graduação em Engenharia Civil e Ambiental). Unidade Acadêmica de Engenharia Civil. Universidade Federal de Campina Grande. Campina Grande, 2022.
- DEVALAL, S.; KARTHIKEYAN, A. LoRa technology-an overview. In: SECOND INTERNATIONAL CONFERENCE ON ELECTRONICS, COMMUNICATION AND AEROSPACE TECHNOLOGY (ICECA), IEEE, n. Iceca, p. 284–290, 2018.
- ELISIARIO, A. T. S et al. Levantamento das tecnologias sem fio para IoT e Indústria 4.0. *Revista Acadêmica - Ensino de ciências e tecnologias*. Número 10. IFSP – CAMPUS CUBATÃO. 2022.
- ELSISI, Mahmoud; MAHMOUD, Karar; LEHTONEN, Matti; DARWISH, Mohamed M. F. Reliable Industry 4.0 Based on Machine Learning and IoT for Analyzing, Monitoring, and Securing Smart Meters. *Sensors*, v. 21, n. 2, p. 487, jan. 2021.
- FIELDING, Thomas. Architectural Styles and the Design of Network-based Software Architectures. Tese (Doutorado em Filosofia em ciência da informação e computação: Ensino Superior). 180p. University of california, irvine 2000.
- FRIEDMAN, K. et al. Water Demand Management optimization methodology. American Water Works Association. 2011.
- GAIA, P. A quarta revolução industrial e as tendências tecnológicas no segmento de equipamentos, máquinas e acessórios industriais. *O Papel: revista mensal de tecnologia em celulose e papel*, v. 77, n. 5, p. 21-25, 2016.
- GALINDO JÚNIOR, E. A, ROCHA, R. D, MACIEL, R. S. Desenvolvimento de API rest com spring boot. *Revista Científica do UniRios*. 2021.

GONÇALVES, D. F. H. Análise das perdas de água do sistema de abastecimento do Município de Pomerode/SC. Monografia (Bacharelado em Engenharia Sanitária e Ambiental) - Universidade Federal de Santa Catarina, Santa Catarina, 2018.

GOUVEIA, Alexandre. O que é uma API. Universidade Positivo, 2016. Disponível em: <https://blogs.up.edu.br/engenharia-da-computacao/2016/07/01/o-que-e-uma-api/>. Acesso em: 14 maio. 2023.

GUBBI, J.; BUYYA, R.; MARUSIC, S. & PALANISWAMI, M. Internet of things (iot): A vision, architectural elements, and future directions. Future Generation Computer Systems, 2013.

GUERRA, C. L. de B. N. Teleodontologia. In: A Telessaúde no Brasil e a Inovação Tecnológica na Atenção Primária. Natal: EDUFRRN, 2015.

IVANOVICH, B. V. et al. Using mvc pattern in the software development to simulate production of high cylindrical steel ingots. Journal of Crystal Growth, v. 526, p. 125240, 2019.

KALLIAMVAKOU, E., BLINCOE, K., SINGER, L., DAMIAN, D. The promises and Perils of Mining Github. Working Conference on Mining Software Repositories. 2014.

KAUFMANN, S.; KÜNZEL, K.; LOOCK, M. Customer value of smart metering: Explorative evidence from a choice-based conjoint study in Switzerland. Energy Policy 2013.

KEARNEY, A. The MVC Schema. 2019. Disponível em: <https://medium.com/@adamkearney124/model-view-controller-f2bdf1ee999>.
MAGALHÃES, R; VENDRAMINI, A. Os impactos da quarta revolução industrial. Revista GVExecutivo. V17, número 1. 2018.

MARTINS, Gesáina. Proposta de um sistema inteligente de distribuição de água (Smart Water System) em um campus universitário. Dissertação de mestrado. Universidade Federal de Ouro Preto. 2021.

MARTNOSKI, A. K.; GHISI, E. Aproveitamento de água pluvial para usos não potáveis em instituições de ensino: estudo de caso em Florianópolis - SC. Ambiente Construído, Porto Alegre, V.8, n.2, p. 67-84. 2008.

MEDEIROS, Luísa Eduarda Lucena. Desenvolvimento de sistema automatizado de baixo custo para Medição inteligente de vazão em redes de abastecimento. Tese (Doutorado em Engenharia e Gestão de Recursos Naturais). Universidade Federal de Campina Grande. 2022.

OLIVEIRA, Leonan et al. Arquitetura baseada em Computação em Névoa para Sistemas de Gerenciamento Inteligente de Água. In: Anais do II Workshop de Computação Urbana. SBC, 2018.

ONU - ORGANIZAÇÃO DAS NAÇÕES UNIDAS. Declaração da Conferência de ONU no Ambiente Humano, 1972.

OTAKI, Yurina; HONDA, Hidehito; UEDA, Kazuhiro. Water demand management: Visualising a public good. PLOS ONE, v. 15, n. 6, p. e0234621, 16 jun. 2020.

PEREIRA, Weberson. API: conceito, exemplos de uso e importância da integração para desenvolvedores. Take, 2019. Disponível em : <<https://take.net/blog/devs/api-conceito-e-exemplos>>. Acesso em: 14 maio de 2023.

RODRIGUES, Felipe Moraes. Sistema de egressos ufpa – campus castanhal. Monografia. Universidade Federal do Pará. 2021.

SANTOS, Maria Aparecida da Silva. Processo para criação de serviços web para o encapsulamento e acesso de uma api específica. Monografia. Universidade Federal do Ceará. 2018.

SAVENIJE, H. H. G.; VAN DER ZAAG, P. Water as an economic good and demand management: paradigms and pitfalls. Water International, v. 27, n°. 1, p. 98-104. 2002.

SCHNEIDER, Gabriela Pelles. O impacto da globalização e da quarta revolução industrial para os estados periféricos: o caso do “estado-nação” brasileiro e as consequências para o mundo do trabalho. Dissertação (Mestrado). Programa de Pós-Graduação em Direitos e Garantias Fundamentais da Faculdade de Direito de Vitória. Vitória, 2019.

SCHVARTZ, A.; VALVERDE, L. Desenvolvimento ágil e modelo de negócios voltados para aplicações web. Monografia. Universidade Federal do Rio de Janeiro, 2013.

SCHWAB, Klaus. A quarta revolução industrial. São Paulo: Edipro, 2016. 159p.
SICHE, R. et al. Índices versus indicadores: precisões conceituais na discussão da sustentabilidade de países. Ambient. soe. vol. 10. No.2 Campinas. 2007.

SIKIMIĆ, M; et al. An Overview of Wireless Technologies for IoT Network. In: INTERNATIONAL SYMPOSIUM INFOTEH-JAHORINA. 2020.

SILVA JUNIOR, K. M. V. da. Softfarm: um sistema de gestão de propriedades rurais Monografia. Unidade Acadêmica de Computação UEPB. 2022.

Sistema Nacional de Informações sobre Saneamento (SNIS). Instituto Trata Brasil. Relatório. São Paulo. 2022. Disponível em: https://tratabrasil.org.br/wp-content/uploads/2022/09/Relatorio_Completo.pdf. Acesso em: 24 maio. 2023.

SOARES, Antonio Leomar Ferreira. Gerenciamento da demanda de água em ambientes de uso público: o caso da Universidade Federal de Campina Grande. Dissertação (Pós-Graduação em Engenharia Civil e Ambiental). Unidade Acadêmica de Engenharia Civil. Universidade Federal de Campina Grande. Campina Grande, 2012.

SOMMERVILLE, I. Engenharia de Software. 9. ed. Pearson Education. 2011.

SOUZA, I. V. A; VIEIRA, V. P. P. B. A influência do Banco Mundial no Gerenciamento dos Recursos Hídricos do Nordeste Brasileiro. VII Simpósio de Recursos Hídricos do Nordeste, 2004.

STEWART, R.A.; NGUYEN, K.; BEAL, C.; ZHANG, H.; SAHIN, O.; BERTONE, E.; VIEIRA, A.S.; CASTELLETTI, A.; COMINOLA, A.; GIULIANI, M.; Integrated intelligent water-energy metering systems and informatics: Visioning a digital multi-utility service provider. *Environ. Model. Softw.* 2018.

STUDART, T. M. C; CAMPOS, N. Gestão da demanda. *Gestão das águas: princípios e práticas*. Organizadores: CAMPOS, N.; STUDART, T. M. 2a edição. ABRH: Fortaleza, 2001.

VERMESAN, O.; FRIESS, P. *Internet of things: converging Technologies for smart environments and integrated ecosystems*. River Publishers, 2013.