



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CAMPUS DE QUIXADÁ**  
**CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO**

**DANIEL DE MOURA MASCARENHAS**

**MEDIDOR DE ÁGUA INTELIGENTE PARA MONITORAMENTO DO CONSUMO DE  
ÁGUA**

**QUIXADÁ**  
**2024**

DANIEL DE MOURA MASCARENHAS

MEDIDOR DE ÁGUA INTELIGENTE PARA MONITORAMENTO DO CONSUMO DE  
ÁGUA

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação do Campus de Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Orientador: Prof. Me. Luis Rodolfo Rebouças Coutinho.

QUIXADÁ

2024

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Sistema de Bibliotecas  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

M361m Mascarenhas, Daniel de Moura.

Medidor de água inteligente para monitoramento do consumo de água / Daniel de Moura

Mascarenhas. – 2024.

66 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá,  
Curso de Engenharia de Computação, Quixadá, 2024.

Orientação: Prof. Me. Luis Rodolfo Rebouças Coutinho.

1. Medidor de água inteligente. 2. Consumo de água. 3. Internet das coisas. 4. Sistemas embarcados.  
I. Título.

CDD 621.39

---

DANIEL DE MOURA MASCARENHAS

MEDIDOR DE ÁGUA INTELIGENTE PARA MONITORAMENTO DO CONSUMO DE  
ÁGUA

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação do Campus de Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Aprovada em: 25/09/2024.

BANCA EXAMINADORA

---

Prof. Me. Luis Rodolfo Rebouças  
Coutinho (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Thiago Werlley Bandeira da Silva  
Universidade Federal do Ceará (UFC)

---

Prof. Me. Carlos Igor Ramos Bandeira  
Universidade Federal do Ceará (UFC)

À minha Mãe, pelo apoio, incentivo e dedicação. Por ela eu não desisti e continuei firme a caminho da vitória.

## **AGRADECIMENTOS**

Primeiramente, agradeço a Deus por me guiar e abençoar em cada etapa desta jornada. Agradeço aos meus pais, por sempre me incentivarem a seguir meus sonhos e por todo o amor e apoio incondicional.

Ao meu orientador Prof. Me. Luis Rodolfo Rebouças Coutinho pelo apoio e contribuições no desenvolvimento desse trabalho. Principalmente, por ter aceitado ser meu orientador.

Aos demais professores do curso de Engenharia de Computação pelos conhecimentos adquiridos. Também, aos coordenadores e funcionários do curso pelo apoio.

Aos colegas de curso, pelas amizades construídas durante todo o período do curso.

Aos professores Prof. Dr. Thiago Werley Bandeira da Silva e Prof. Me. Carlos Igor Ramos Bandeira por aceitarem participar dessa banca examinadora e pelas contribuições.

"A internet é muito mais que uma tecnologia. É um meio de comunicação, de interação e de organização social." Manuel Castells ()

## RESUMO

Este trabalho, destaca a problemática do consumo de água e como utilizar a Internet das Coisas (IoT) para melhorar a gestão e a preservação desse recurso. Nesse contexto, IoT já vem desempenhando um papel fundamental no desenvolvimento de soluções tecnológicas e sustentáveis nessa área. Assim, pretende-se desenvolver um dispositivo inteligente (medidor de água) com funcionalidades para auxiliar o monitoramento do consumo de água. A partir do qual pretende-se avaliar se o dispositivo atende às necessidades de monitoramento do consumo de água dos usuários, como também, identificar se contribuiu para redução no consumo de água da residência.

**Palavras-chave:** medidor de água inteligente; consumo de água; internet das coisas; sistemas embarcados



## **ABSTRACT**

This work highlights the problem of water consumption and how to use the Internet of Things (IoT) to improve the management and preservation of this resource. In this context, IoT has already been playing a key role in the development of technological and sustainable solutions in this area. Thus, it is intended to develop an intelligent device (water meter) with features to help monitor water consumption. From which it is intended to assess whether the device meets the needs of monitoring the water consumption of users, as well as to identify whether it contributed to a reduction in the consumption of water in the residence.

**Keywords:** smart water meter; water consumption; internet of Things; embedded systems.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Elementos de <i>Internet das Coisas</i> (IoT) . . . . .	18
Figura 2 – Topologia: (a) barra e (b) estrela . . . . .	20
Figura 3 – Tipos de <i>Basic Service Set</i> (BSS) . . . . .	21
Figura 4 – Tipos de BSS, ESS e DS . . . . .	22
Figura 5 – Componentes de uma <i>Redes de Sensores Sem Fio</i> (RSSF) . . . . .	23
Figura 6 – Arquitetura dos dispositivos . . . . .	25
Figura 7 – Placa Wifi Atmega328 + Esp8266 Ch340 . . . . .	27
Figura 8 – Sensor . . . . .	28
Figura 9 – Componentes do HIL . . . . .	29
Figura 10 – Fluxo do Processo . . . . .	34
Figura 11 – Caso de uso . . . . .	37
Figura 12 – Sistema Arduino + Sensor de fluxo . . . . .	40
Figura 13 – Fluxo: Montar projeto . . . . .	41
Figura 14 – Configuração de <i>Switches</i> . . . . .	42
Figura 15 – Planilha . . . . .	43
Figura 16 – Fluxo de calibração do sensor . . . . .	44
Figura 17 – Fluxo criação da planilha Google . . . . .	46
Figura 18 – Gráfico de consumo de água . . . . .	47
Figura 19 – Gráfico Consumo Média Móvel . . . . .	47
Figura 20 – Gráfico de Consumo Médio e Previsão . . . . .	48
Figura 21 – E-mail . . . . .	49
Figura 22 – Simulador de dados . . . . .	50
Figura 23 – Testes feitos com simulação de sinais . . . . .	51
Figura 24 – Email de alerta de fluxo constante durante simulação . . . . .	51
Figura 25 – Medido de água instalado . . . . .	52

## LISTA DE TABELAS

Tabela 1 – Protocolo Wi-Fi, frequência e taxas máximas teóricas de dados. . . . .	22
Tabela 2 – Trabalhos relacionados. . . . .	32
Tabela 3 – Requisitos Funcionais . . . . .	35
Tabela 4 – Requisitos Não Funcionais . . . . .	36
Tabela 5 – Especificação dos casos de uso . . . . .	37

## LISTA DE ABREVIATURAS E SIGLAS

AP	<i>Access Point</i>
BSS	<i>Basic Service Set</i>
CAGECE	<i>Companhia de Água e Esgoto do Estado do Ceará</i>
CIoT	<i>Central Internet of Things</i>
CPU	<i>Central Processing Unit</i>
DIX	<i>Digital, Xerox e Intel</i>
DS	<i>Distribution System</i>
ESS	<i>Extended Service Set</i>
EXI	<i>Efficient XML Interchange</i>
HIL	<i>Hardware-in-the-loop</i>
IDE	<i>Integrated Development Environment</i>
IoT	<i>Internet das Coisas</i>
IP	<i>Internet Protocol ou protocolo de rede</i>
LAN	<i>Local Area Network</i>
MAN	<i>Metropolitan Area Network</i>
ONU	<i>Organização das Nações Unidas</i>
OWL	<i>Web Ontology Language</i>
RDF	<i>Resource Description Framework</i>
RSSF	<i>Redes de Sensores Sem Fio</i>
RUP	<i>Rational Unified Process</i>
TI	<i>Tecnologia da Informação</i>
UML	<i>Unified Modeling Language</i>
WAN	<i>Wide Area Network</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
<b>1.1</b>	<b>Motivação e Justificativa</b>	<b>15</b>
<b>1.2</b>	<b>Objetivo Geral</b>	<b>15</b>
<b>1.3</b>	<b>Objetivos Específicos</b>	<b>16</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>17</b>
<b>2.1</b>	<b>Internet das Coisas</b>	<b>17</b>
<b>2.1.1</b>	<i>Tecnologias de Comunicação</i>	<b>19</b>
<b>2.1.1.1</b>	<i>Padrão ETHERNET (IEEE 802.3)</i>	<b>20</b>
<b>2.1.1.2</b>	<i>Padrão IEEE 802.11</i>	<b>21</b>
<b>2.1.1.3</b>	<i>Rede de Sensores sem Fio (RSSF)</i>	<b>22</b>
<b>2.2</b>	<b>Sistemas Embarcados</b>	<b>24</b>
<b>2.2.1</b>	<i>Microcontroladores e Microprocessadores</i>	<b>25</b>
<b>2.2.2</b>	<i>Sensores</i>	<b>27</b>
<b>2.2.3</b>	<i>Hardware-In-The-Loop (Hardware-in-the-loop (HIL))</i>	<b>28</b>
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	<b>30</b>
<b>4</b>	<b>METODOLOGIA</b>	<b>33</b>
<b>4.1</b>	<b>Concepção</b>	<b>33</b>
<b>4.1.1</b>	<i>Requisitos funcionais e não funcionais</i>	<b>35</b>
<b>4.2</b>	<b>Elaboração</b>	<b>36</b>
<b>4.2.1</b>	<i>Casos de Uso</i>	<b>36</b>
<b>4.2.2</b>	<i>Especificação dos Casos de Uso</i>	<b>36</b>
<b>4.2.3</b>	<i>Esquemático do projeto</i>	<b>39</b>
<b>4.3</b>	<b>Construção</b>	<b>40</b>
<b>4.3.1</b>	<i>Montagem do medidor de água</i>	<b>40</b>
<b>4.3.2</b>	<i>Calibração do sensor</i>	<b>42</b>
<b>4.3.3</b>	<i>Integração com comunicação via Wi-Fi (ESP8266)</i>	<b>44</b>
<b>4.3.4</b>	<i>Planilha Google</i>	<b>45</b>
<b>4.3.5</b>	<i>Visualização dos dados</i>	<b>46</b>
<b>4.3.6</b>	<i>Configuração da ESP8266 para envio de e-mail</i>	<b>48</b>
<b>4.4</b>	<b>Transição</b>	<b>49</b>

<b>4.4.1</b>	<b><i>Testes e Validação</i></b> . . . . .	<b>49</b>
<b>5</b>	<b>CONSIDERAÇÕES FINAIS</b> . . . . .	<b>53</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>55</b>
<b>6</b>	<b>CÓDIGOS-FONTES UTILIZADOS</b> . . . . .	<b>58</b>
<b>6.1</b>	<b>Código Arduino</b> . . . . .	<b>58</b>
<b>6.2</b>	<b>Código ESP8266</b> . . . . .	<b>59</b>
<b>6.3</b>	<b>Script para Google Sheets</b> . . . . .	<b>64</b>

## 1 INTRODUÇÃO

Atualmente, a escassez dos recursos hídricos e a má utilização desses recursos têm sido objeto de muitos debates presentes nas mídias, no meio acadêmico e na sociedade em geral. No foco das discussões, encontram-se as preocupações com os recursos hídricos e como utilizá-los de forma sustentável no sentido de preservá-los para as gerações futuras.

No Brasil, em 2022, o consumo médio de água por habitante foi em média de, aproximadamente, 148,2 litros por habitante em um dia, sendo que foram atendidas aproximadamente, 171 milhões de pessoas, correspondendo a uma média de 84,9% da população. Entretanto, de toda água potável que se destina às residências dos brasileiros, foi registrado um percentual de 37,8% de água não contabilizada ou perdida na distribuição. Esse desperdício seria suficiente para abastecer 67 milhões de pessoas em um ano, equivalente a 30% da população brasileira em 2021. Portanto, garantir acesso à água de qualidade a todos os brasileiros é um dos principais desafios para os governantes do nosso país (Trata-Brasil, 2023).

Embora os percentuais de perdas possam estar relacionados a perdas no processo de distribuição da água, ou seja, no trajeto até as residências, não se pode desconsiderar as perdas que ocorrem pela má utilização da água ou em decorrência de vazamentos nas unidades consumidoras.

No Ceará, a *Companhia de Água e Esgoto do Estado do Ceará* (CAGECE) diante do agravamento da escassez hídrica, tem intensificado seus esforços para garantir o abastecimento contínuo de água à população. A empresa tem investido em novas tecnologias de tratamento de água, diversificado suas fontes hídricas e implementado medidas para reduzir as perdas na rede de distribuição. Mas também conta com o apoio da população, especialmente com as boas práticas de economia de água (CAGECE, 2024).

Essas boas práticas de consumo de água podem ser realizadas por meio de ações simples do dia a dia, como fechar a torneira, enquanto se escova os dentes; manter as instalações hidráulicas em boas condições, evitar banhos demorados entre outras, apesar de simples são ações importantes para evitar o desperdício e preservar o meio ambiente. Entretanto, a tarefa de monitorar o consumo de água em uma residência pode não ser fácil. Pois, não significa apenas controlar as ações de todos os familiares, o que já é uma tarefa desgastante, mas também identificar vazamentos, muitas vezes, imperceptíveis.

Diante do exposto, esta pesquisa faz os seguintes questionamentos: como estimular as pessoas a reduzirem o consumo de água? Que produtos, serviços ou ações podem ser

desenvolvidos para atingir esse objetivo? Como as tecnologias digitais podem contribuir nesse propósito?

Atualmente, as tecnologias já desempenham um papel fundamental para o desenvolvimento de soluções sustentáveis, com destaque para IoT. As tecnologias que utilizam IoT partem da ideia de que as coisas do mundo físico podem se comunicar com o virtual, fornecendo informações aos usuários que podem ser acessadas de qualquer lugar e auxiliá-los na tomada de decisões.

No rol dos serviços/produtos desenvolvidos com tecnologia IoT, encontram-se várias soluções que contribuem para a sustentabilidade. São exemplos disso os dispositivos inteligentes para monitorar gás, energia elétrica, entre outros. Em relação aos recursos hídricos, existem sistemas que utilizam sensores para detectar vazamento nas tubulações de distribuição da água, para monitorar o consumo de água na agricultura, comércios e indústrias. Também, existem concessionárias de água que já têm hidrômetros inteligentes instalados em algumas unidades. Esses são alguns dos exemplos que comprovam o potencial dessa tecnologia em prol da sustentabilidade.

Portanto, em busca de respostas para os questionamentos da pesquisa, inicialmente, observou-se que o consumo de água é realizado de forma tão automática, instintiva pelas pessoas, provavelmente devido a importância desse recurso para a vida humana, que muitas vezes não refletimos sobre as melhores maneiras para sua utilização.

Mesmo aqueles que têm interesse em adotar um consumo consciente, sentem dificuldade em mensurar esse consumo. Ou seja, a necessidade instintiva pelo consumo de água atrelada às dificuldades em mensurá-lo fazem com que as pessoas não percebam o quanto consomem de água e não consigam fazer uma distinção entre o consumo adequado às suas necessidades e o que representa desperdício.

Partindo desse entendimento, supõe-se que o uso de um dispositivo inteligente, no caso, um medidor de água, além de resolver as dificuldades dos usuários com o monitoramento do consumo de água, também possa contribuir para a redução do consumo desse recurso.

Portanto, é nessa linha de entendimento que se pretende desenvolver esse trabalho, cujas motivações e justificativas são apresentadas a seguir.



## 1.1 Motivação e Justificativa

Considerando a importância que a água tem para a preservação da vida no planeta, a motivação deste trabalho está centrada em contribuir com a sustentabilidade dos recursos hídricos, ou seja, suprir as necessidades de consumo de água no presente preservando esse recurso para as gerações futuras.

A pesquisa pretende desenvolver um dispositivo inteligente para monitoramento do consumo de água, capaz de fornecer, remotamente e em tempo real, informações de consumo de água da unidade consumidora.

Considera-se que esse projeto se justifica pelo potencial que a tecnologia IoT tem em contribuir para uma melhor gestão do consumo de água, como também, da possibilidade de que o uso desse dispositivo possa levar a uma maior conscientização das pessoas para o uso sustentável no consumo de água.

Ademais, a temática aqui abordada está tão em evidência que cabe destacar a realização da Conferência de Água da *Organização das Nações Unidas* (ONU), em 2023, após mais de 40 anos da última conferência, impulsionados pela crise global da água que sofreu uma aceleração significativa nas últimas décadas. Comparando-se com a década de 1970, os especialistas concluíram que os sistemas de água doce do planeta estão se extinguindo. Observaram que populações de espécies monitoradas que vivem em água doce reduziram em média 83% desde 1970 e que já foi perdido 64% das terras úmidas do planeta desde 1900, e que apenas 37% dos rios mais longos permanecem desimpedidos e fluindo livremente (NATURE, 2023). Portanto, demonstrando a relevância da temática e a urgência em buscar soluções para preservação desse recurso natural.

Dessa forma, considera-se importante, não apenas esse projeto, mais todos que promovam uma utilização mais eficiente dos recursos naturais e contribuam com sua conservação. Por conseguinte, ajudando a construir um futuro mais sustentável para as próximas gerações.

## 1.2 Objetivo Geral

Buscando responder aos questionamentos da pesquisa é que se pretende desenvolver um dispositivo inteligente, com funcionalidades para auxiliar o monitoramento do consumo de água. Portanto, o objetivo geral deste trabalho é:

- Desenvolver dispositivo inteligente (medidor de consumo de água) utilizando placa Arduino, sensor de vazão para monitorar remotamente e em tempo real o consumo de água em residências, possibilitando o uso racional do consumo de água.

### **1.3 Objetivos Específicos**

- Desenvolver medidor de água inteligente para o monitoramento de consumo de água;
- Verificar se o medidor de água apresenta as funcionalidades necessárias para o monitoramento do consumo de água pelos usuários;
- Identificar as potencialidades e limitações do medidor de água para monitoramento do consumo de água.

Para finalizar, destaca-se que este trabalho possui quatro capítulos. O Capítulo 2 traz a fundamentação teórica que fornece os conhecimentos necessários para o desenvolvimento do dispositivo inteligente proposto. O Capítulo 3 discute os trabalhos relacionados, revisando pesquisas e projetos semelhantes. Em seguida, o Capítulo 4 descreve detalhadamente a metodologia adotada no desenvolvimento do trabalho. E, por fim, o último capítulo traz as considerações finais, onde são apresentadas as conclusões a partir dos resultados obtidos.

## 2 FUNDAMENTAÇÃO TEÓRICA

Atualmente, diante de um mundo tecnológico e altamente conectado em que vivemos, pode-se dizer que a grande maioria das pessoas, em algum momento do seu cotidiano, utilizou, presenciou ou ouviu falar de objetos inteligentes. Tais dispositivos proporcionam conforto, comodidade e praticidade nas atividades diárias de uma residência, por exemplo, por meio de eletrodomésticos inteligentes, mas também, existem aqueles desenvolvidos para auxiliar no controle das despesas residenciais relacionadas ao consumo de energia, gás e água.

### 2.1 Internet das Coisas

A Internet das Coisas (IoT) é a tradução do inglês *Internet of Things*. Segundo Magrani (2018), a IoT engloba um ambiente de objetos físicos interconectados com a internet por meio de sensores pequenos e embutidos, criando um ecossistema de computação onipresente, voltado para a facilitação do cotidiano das pessoas, introduzindo soluções funcionais nos processos do dia a dia.

Apesar da complexidade tecnológica embutida na IoT, Santos *et al.* (2016) facilitam esse entendimento ao concluírem que a IoT nada mais é do que uma extensão da Internet atual, que proporciona a quaisquer objetos de se conectarem à Internet desde que tenham capacidade computacional e de comunicação. Essa conexão com a rede mundial de computadores viabiliza controlar remotamente os objetos e permitir que os próprios objetos sejam acessados como provedores de serviços.

Hoje já existe uma gama de produtos e soluções que utilizam a IoT que pelo grande potencial de desenvolvimento de soluções nas mais diversas áreas, tem atraído cada vez mais investidores com interesse de aplicá-la em novas soluções (Maschietto *et al.*, 2021). Porém, todo esse desenvolvimento só foi possível a partir da evolução do protocolo ethernet (do desenvolvimento do IPv6) e da mobilidade dos dispositivos de comunicação que possibilitaram a IoT endereçar qualquer dispositivo, permitindo a comunicação entre eles e criando uma rede de informações com uma imensa quantidade de dados Júnior e Farinelli (2018).

Para uma melhor percepção do real significado e funcionalidades da IoT, de acordo com Al-Fuqaha *et al.* (2015), é preciso compreender os blocos de construção da IoT:

- Identificação: é um elemento crucial para a IoT poder nomear e combinar o serviço com a demanda correspondente. Entre os métodos de endereçamento possíveis de serem

Figura 1 – Elementos de IoT



Fonte: Al-Fuqaha *et al.* (2015)

utilizados, incluem-se o IPv6, IPv4 e 6LoWPAN<sup>1</sup>;

- Sensoriamento: os sensores são responsáveis por coletarem as informações dos objetos relacionados dentro da rede e, em seguida, transmitem esses dados para uma *data warehouse*, armazenamento em nuvem ou centros de armazenamento;
- Comunicação: refere-se as tecnologias de comunicação da IoT. Eis alguns exemplos de protocolos de comunicação utilizados para IoT: WiFi<sup>2</sup>, Bluetooth<sup>3</sup>, IEEE 802.15.4<sup>4</sup>, IEEE, 802.15.4<sup>5</sup>, Z-wave<sup>6</sup>, and LTE-Advanced e RFID<sup>7</sup>, a primeira usada para conectar os objetos inteligentes;
- Computação: constituído por unidades de processamento como, por exemplo, microcontroladores, microprocessadores e aplicativos de softwares que representam a capacidade computacional da IoT;
- Serviços: A IoT pode fornecer vários serviços que podem ser agrupados em quatro tipos: identidade, agregação de informações, colaborativos conscientes e ubíquos. Os serviços relacionados à identidade são básicos para serem usados em outros tipos de serviços, considerando que todo aplicativo que precisa trocar informações entre o mundo físico e virtual precisa estar identificado na rede. Os serviços de agregação de informações coletam dados brutos de medições sensoriais que precisam ser processadas e relatadas ao aplicativo IoT. Os serviços ubíquos, ou serviços de ubiquidade, objetivam oferecer serviços sempre que forem necessários para qualquer pessoa em qualquer lugar. Podendo-se considerar que atingir esse nível seria o objetivo de todas as aplicações IoT;
- Por fim, a semântica: este serviço se refere à capacidade de extrair o conhecimento de

<sup>1</sup> Protocolos de comunicação de rede

<sup>2</sup> Conexão com internet sem fio

<sup>3</sup> Conexão entre dispositivos

<sup>4</sup> Protocolo padrão de comunicação

<sup>5</sup> Protocolo de rede sem fio para baixas taxas de transmissão

<sup>6</sup> Comunicação por ondas de rádio de baixa energia

<sup>7</sup> Comunicação infravermelha

forma inteligente dos objetos na IoT para fornecer os serviços necessários. Dessa forma, a semântica representa o cérebro da IoT enviando demandas para o recurso correto. Eis algumas das tecnologias que podem ser utilizadas: *Resource Description Framework (RDF)*<sup>8</sup>, *Web Ontology Language (OWL)*<sup>9</sup> e *Efficient XML Interchange (EXI)*<sup>10</sup>.

Ante o exposto, é possível não apenas identificar a presença da IoT no nosso cotidiano, como geladeiras, cafeteiras, máquina de lavar etc, mas também, compreender o seu funcionamento. Entretanto, a IoT não está restrita a objetos domésticos, também está presente em dispositivos que atuam nas mais diversas áreas: indústria, agricultura, hospitais dentre outras, tornando-se uma das tecnologias mais importante do nosso século que atrelada a velocidade na transferência de dados proporcionada pela rede 5G tem provocado uma nova revolução tecnológica.

Nesse contexto da IoT surgiram casas e prédios inteligentes, são os chamados *Smart buildings* que se espalharam pelo mundo, inclusive, presente no Brasil. São construções que oferecem além do conforto, também, proporcionam segurança, economia e eficiência.

Morales (2012) esclarece que prédios inteligentes significam aplicar soluções de gerenciamento de *Tecnologia da Informação (TI)* para manter, melhorar a eficiência e reduzir os gastos como, por exemplo, com energia elétrica. Nessa categoria se enquadram os edifícios Eldorado Business Tower e a Torre Santander ambos localizados em São Paulo que utilizam diversos recursos tecnológicos que auxiliam na manutenção do prédio como o sistema de operação e monitoramento remoto das instalações prediais, incluindo iluminação, sistema de ar-condicionado e sistema hidráulico. Assim como os prédios, as casas também podem ser equipadas com recursos tecnológicos que além de conforto, segurança e comodidade, podem auxiliar no monitoramento do consumo de energia, gás e água que representam os principais custos para manutenção de uma residência. Portanto, pode-se considerar que a IoT tem se tornado uma das tecnologias mais importante do século XXI (ORACLE, 2023).

### **2.1.1 Tecnologias de Comunicação**

No contexto do mundo da IoT, incluindo os sensores, atuadores, microcontroladores ou qualquer dispositivo que esteja sendo usado para estabelecer comunicação entre mundo externo e o mundo virtual, necessitam estar conectados a uma rede para poderem enviar ou

---

<sup>8</sup> Especificação da W3CC

<sup>9</sup> Linguagem para definir instâncias na internet

<sup>10</sup> Representação de alto desempenho no XML

receber informações Essa conexão pode ser realizada com ou sem cabos, mas independente da forma adotada, todas precisam ser estruturadas com um protocolo de comunicação (Ideali, 2021).

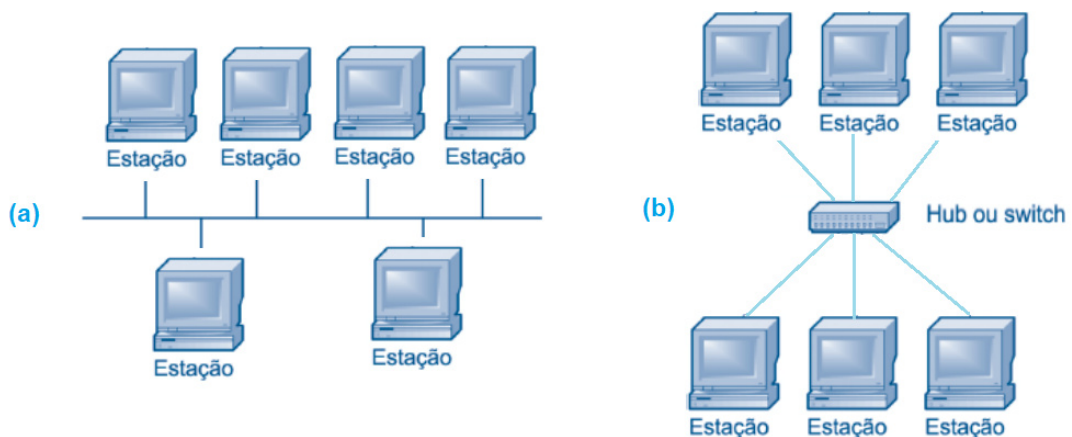
### 2.1.1.1 Padrão ETHERNET (IEEE 802.3)

O padrão ETHERNET se refere a um conjunto de tecnologias para rede de computadores usadas em redes locais *Local Area Network* (LAN), redes de área metropolitana *Metropolitan Area Network* (MAN) e redes de longa distância *Wide Area Network* (WAN).

Essa tecnologia foi desenvolvida por Robert Metcalfe, no centro de pesquisa Palo Alto da Xerox, na década de 1970. Em sua primeira versão, utilizava cabo coaxial, suportava 256 estações, com taxas de transmissão de até 3 Mbps. Posteriormente, na década de 1980, as empresas *Digital, Xerox e Intel* (DIX) se uniram para desenvolver produtos com tecnologia Ethernet, com taxas de transmissão de até 10 Mbps. Em 1983, o IEEE criou o padrão 802.3 para redes locais, que, em grande parte, teve por referência o trabalho desenvolvido pelo DIX (Maia, 2013).

Foi sendo aprimorada, tornando-se a tecnologia de rede dominante em todo o mundo, substituindo amplamente as tecnologias LAN com fio concorrentes (Ideali, 2021). Essa evolução, segundo Maia (2013), ocorreu em transmissão, topologia e cabeamento. As taxas de transmissão saíram da faixa 10 Mbps para 100 Gigabit e crescendo. Já a topologia, inicialmente em barra, foi substituída pela topologia em estrela, utilizando hubs e switches (Figura 2), o que trouxe maior escalabilidade, facilidade de instalação e manutenção. O cabeamento, também, passou por transformações, evoluindo de cabos coaxiais para o par trançado e fibra óptica.

Figura 2 – Topologia: (a) barra e (b) estrela



Fonte: Adaptado de Maia (2013)

### 2.1.1.2 Padrão IEEE 802.11

Diz respeito a um conjunto de protocolos que se referem a redes locais sem fio (WLAN), popularmente conhecido como WiFi.

Esse padrão possui estações agrupadas em conjuntos chamados BSS que podem ser organizadas de duas formas diferentes: infraestruturada e *ad hoc*. Em uma rede infraestruturada, existe a figura do ponto de acesso ou *Access Point* (AP), que centraliza a conexão entre as estações dentro do BSS. Enquanto que em uma rede *ad hoc* ou rede não estruturada, as estações podem comunicar-se diretamente umas com as outras, sem a necessidade de passar por um dispositivo central, conforme pode ser observado na Figura 3 (Maia, 2013).

Figura 3 – Tipos de BSS

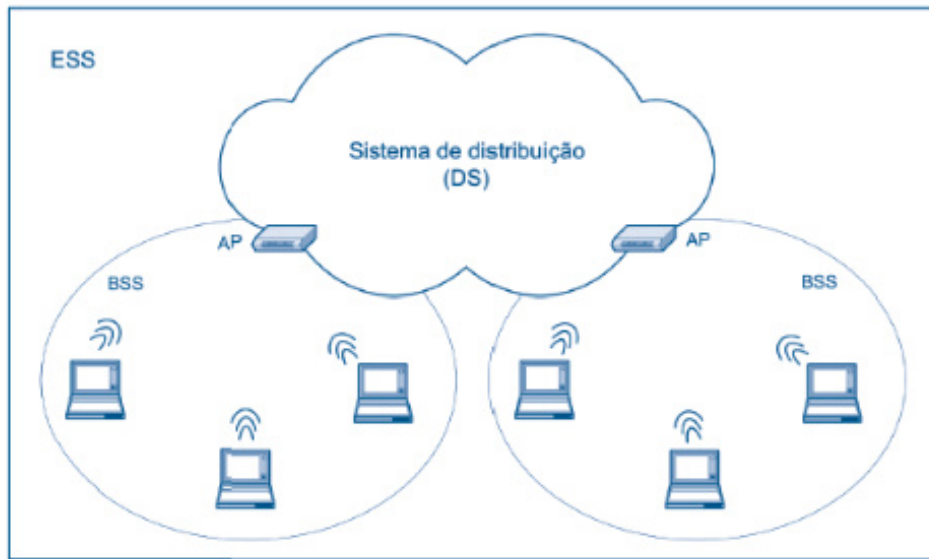


Fonte: Maia (2013)

Maia (2013) esclarece-se que o ponto de acesso, organiza as estações de um BSS, mas também tem a função de conectar os diferentes BSS, formando o que se chama de um *Extended Service Set* (ESS). No qual o ponto de acesso funciona como uma ponte que conecta o BSS a um sistema de distribuição ou *Distribution System* (DS). O sistema de distribuição tem a função de interconectar os vários BSS, podendo ser uma rede cabeada Ethernet ou uma outra rede sem fio conforme pode ser observado na Figura 4.

Maia (2013) ainda complementa que uma das grandes vantagens do padrão IEEE 802.11 é permitir a mobilidade das estações enquanto estão em operação, podendo uma estação BSS migrar para um outro BSS de maneira transparente, ocorrendo sem que o usuário tenha que interromper o seu trabalho. Entretanto, a migração transparente entre dois BSS é suportada apenas entre os BSS que estejam no mesmo ESS, não sendo suportada a mobilidade transparente entre ESS diferentes sem a interrupção dos serviços.

Figura 4 – Tipos de BSS, ESS e DS



Fonte: (Maia, 2013)

São muitos os membros da família 802.11, sendo assim, para finalizar com uma visão geral, apresenta-se, na Tabela 1, uma relação dos IEEE 802.11 ao longo de sua existência, com algumas das suas características básicas, como frequência e taxa máxima de dados estimada.

Quadro 1 – Protocolo Wi-Fi, frequência e taxas máximas teóricas de dados.

Protocolo	Frequência	Taxa máxima de dados (teórica)
802.11ax	2,4 ou 5 GHz	2,4 Gbps
802.11ac onda 2	5 GHz	1,73 Gbps
802.11ac onda 1	5 GHz	866,7 Mbps
802.11n	2,4 ou 5 GHz	450 Mbps
802.11g	2,4 GHz	54 Mbps
802.11a	5 GHz	54 Mbps
802.11b	2,4 GHz	11 Mbps
Legado 802.11	2,4 GHz	2 Mbps

Fonte: Adaptado de Intel-Corporation (2024).

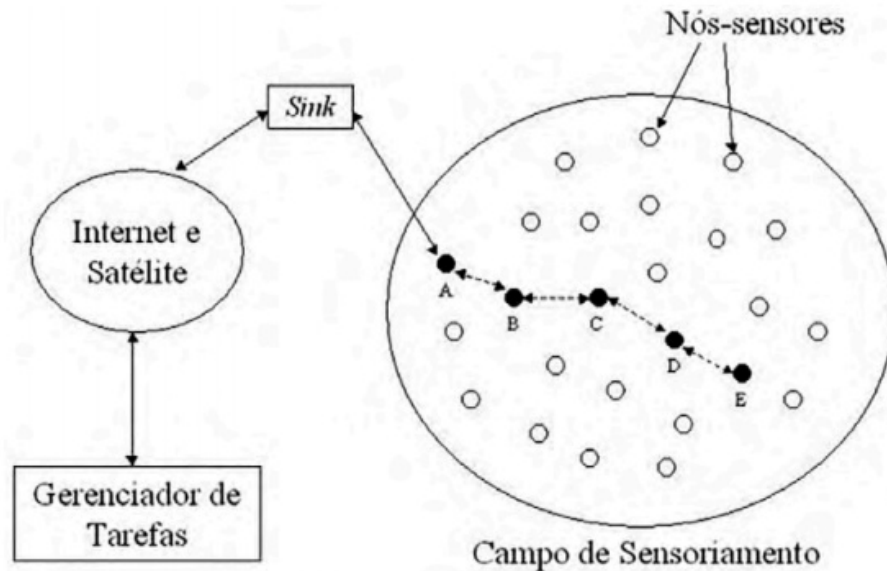
### 2.1.1.3 Rede de Sensores sem Fio (RSSF)

Uma RSSF é constituída por um conjunto de nós sensores com a capacidade de se comunicar entre si, gerando e transmitindo dados de forma colaborativa. Para melhor compreensão, pode-se observar o funcionamento de uma RSSF na Figura 5, na qual os nós sensores estão distribuídos em uma área de sensoriamento e o nó sensor E, coleta os dados e repassa para os nós intermediários (B, C, B, A) através de saltos múltiplos (mukltihop), até chegar ao nó de



controle (sink) que utiliza um meio de comunicação (internet ou um satélite) para repassar os dados para um gerenciador de aplicações (Silva, 2021).

Figura 5 – Componentes de uma RSSF



Fonte: (Silva, 2021)

Segundo Mondal e Zulfi (2017), as RSSF's estão cada vez mais sendo usadas em tecnologias e ideias mais recentes, pois possibilitam a comunicação entre os dispositivos inteligentes. Entretanto, são um desafio para os sistemas embarcados, levando em conta a limitação de dados a serem transmitidos por causa do espaço e pela limitação de energia, visto que um sistema embarcado pode funcionar com base em baterias e com isso uma clara limitação energética. Os referidos autores também acrescentam que os métodos de comunicação estão evoluindo cada vez mais em comparação com os já existentes e que esses tipos diferentes de comunicação podem fazer uma rede que conecta diversos sistemas, que é como é a internet agora, todos conectados em uma gigantesca rede que cobre o mundo todo. Com isso, informações coletadas em um certo sensor podem ser enviadas para outros pela conexão da internet e enviar dados como temperatura, umidade, vibração e diversos outros para qualquer lugar do mundo a partir desses sensores de rede que podem se comunicar pelo endereço *Internet Protocol* ou *protocolo de rede* (IP) que é conjunto de regras que definem o formato de dados enviados pela Internet.

Embora haja dificuldades com diversos sensores conectados por meio da rede é possível fazer um conjunto de sensores que poderão juntos cobrir uma enorme área, podendo

cada um se dedicar a uma certa região do sistema e assim remediar dois dos problemas que os sensores teriam: um é a área de atuação individual, pois com diversos sensores não seria necessário que cubram uma grande área de atuação, já que seriam usados diversos sensores para cobrir tudo; o outro problema que seria evitado refere-se a limitação energética, pois cada sensor separado pode ter sua própria fonte de alimentação. Além disso, também tem métodos de evitar o desperdício de energia, como não deixando-os ligados o tempo todo, funcionando a cada período de tempo pré-definido. Outra forma de usar seria também colocar um tipo de ativação espacial, em que se um sensor de uma área detecte algo, então os sensores em volta também se ativaram para ter mais exatidão (Zhao; Guibas, 2004).

## 2.2 Sistemas Embarcados

Atualmente, tem-se uma gama de produtos como televisões, fornos de micro-ondas, sistemas de gerenciamento de aviação, esteiras transportadoras entre outros, que possuem sistemas embarcados em seus dispositivos. Pode-se considerar que quase todo dispositivo que funcione com eletricidade possui um sistema embarcado coordenando seu funcionamento (Moraes *et al.*, 2016).

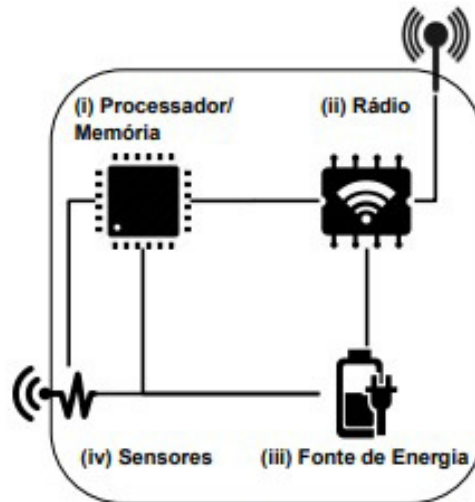
Um sistema embarcado é um conjunto de *hardware* e *software* projetado para executar uma tarefa específica em um sistema maior, ou seja, têm um propósito específico e se concentram em fazer apenas aquilo para o qual foi projetado. São compostos por um conjunto limitado de componentes, como microcontroladores, sensores e atuadores que são integrados em outros produtos ou equipamentos, visando controlar ou monitorar uma determinada função, ou processo (Souza, 2023).

Em suma, são sistemas que possuem uma capacidade de processamento de informações proveniente de um *software* que está sendo processado dentro de uma unidade de processamento (Oliveira; Andrade, 2010). Utiliza-se a expressão “sistemas embarcados” justamente por serem sistemas que trabalham acoplados dentro de sistemas maiores, ou seja, internamente.

Dados os conceitos apresentados anteriormente, a primeira coisa que se pôde compreender foi que os objetos inteligentes possuem sistemas embarcados. Para um melhor entendimento dos objetos inteligentes, Santos *et al.* (2016) detalham a arquitetura básica por trás desses dispositivos inteligentes e as tecnologias de comunicação utilizadas, com foco nas soluções de comunicação sem fio adotadas no ambiente de IoT. Esclarecem que os objetos inteligentes

possuem quatro unidades sendo, uma de processamento/memória, a de comunicação, outra para energia e os sensores/atuadores, conforme apresentado na Figura 6:

Figura 6 – Arquitetura dos dispositivos



Fonte: Santos *et al.* (2016)

Os referidos autores esclarecem que as unidades de processamento/memória são compostas por uma memória interna que armazena os dados e programas, um microcontrolador e um conversor para receber os sinais dos sensores. As *Central Processing Unit* (CPU)'s utilizadas nesses dispositivos são geralmente de baixo poder computacional. Já a unidade de comunicação consiste em, pelo menos, um canal de comunicação, sendo mais comum o meio sem fio que na maioria das plataformas usam rádio de baixo custo e baixa potência. Como consequência, a comunicação é de curto alcance e apresentam perdas frequentes. Em relação a unidade fonte de energia, esta é a responsável por fornecer energia aos componentes do objeto inteligente, podendo ser uma bateria (recarregável ou não) ou outras fontes de alimentação, tais como energia elétrica, solar entre outras. Por fim, tem os sensores/atuadores que realizam o monitoramento do ambiente no qual o objeto se encontra. Os sensores captam os valores de grandezas físicas, tais como temperatura, umidade, pressão, presença e vazão. Enquanto que os atuadores são dispositivos que produzem alguma ação, atendendo a comandos que podem ser manuais, elétricos ou mecânicos.

### 2.2.1 Microcontroladores e Microprocessadores

De acordo com Oliveira e Andrade (2010), os microprocessadores surgiram substituindo milhões de transistores nos computadores. Hoje, estão presentes nos microcomputadores

peçoais, em vários aparelhos eletrônicos domésticos e equipamentos médicos. Entretanto, para que sejam úteis, necessitam de outros dispositivos externos, como memória de programa contendo o programa que será executado pelo microprocessador; barramentos entre essa memória e o microprocessador e barramento de endereços que seleciona o endereço em que o dado será lido ou escrito na memória. Desta forma, microprocessadores deve ter acoplado externamente todo componente necessário para o cumprimento da aplicação.

Já os microprocessadores são circuitos integrados que possuem internamente: o processador, as memórias, as portas de entrada e saída e os periféricos. No mercado, existem várias empresas que fabricam microcontroladores, como por exemplo: Texas Instruments; Microchip / Atmel; STMicroelectronics e NXP Semiconductors (Neto; Oliveira, 2017).

Nesse contexto, para o desenvolvimento do dispositivo inteligente deste trabalho, optou-se por utilizar uma Placa Arduino Uno com Wifi (Figura 7).

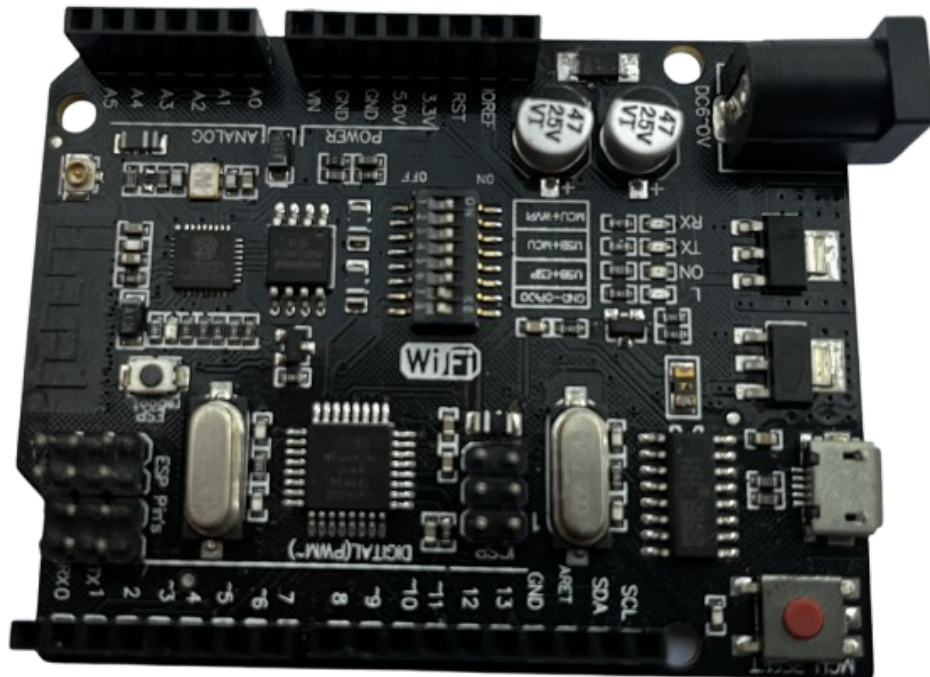
Segundo Oliveira e Zanetti (2015) a Placa Arduino é composta por um microcontrolador Atmel, que pode ser programado por meio de uma linguagem de programação similar a C/C++, permitindo a elaboração de projetos unindo *software* e *hardware*. Pode ser comparada a um computador de pequeno porte, composto por um microcontrolador, memória RAM, armazenamento secundário (memória flash) e clock, entre outras funcionalidades.

Trata-se de uma plataforma de *hardware* de código aberto (*open source*) e possui as seguintes vantagens:

- Baixo custo, sendo relativamente baratas em comparação com outras plataformas de microcontroladores;
- Ambiente de programação simples e claro que possui um software fácil de ser usado por iniciantes, mas também flexível o suficiente para usuários avançados;
- Multiplataforma que permite que a *Integrated Development Environment* (IDE) do software Arduino ser executado nos sistemas operacionais Windows, Macintosh OSX e Linux;
- *Software* de código aberto e extensível que está disponível para extensão por programadores experientes;
- *Hardware* de código aberto e extensível, pois os planos das placas Arduino são publicados sob uma licença *Creative Commons*, para que projetistas de circuitos experientes possam fazer sua própria versão do módulo, estendendo-o e melhorando-o (ARDUINO, 2018).

A placa ESP 8266 integrada a Placa Arduino servirá como a conexão entre o medidor de água inteligente e a internet, sendo inseridos o nome e senha de uma rede wi-fi e podendo

Figura 7 – Placa Wifi Atmega328 + Esp8266 Ch340



Fonte: Próprio autor

enviar dados para serem armazenados em outro lugar e posteriormente utilizados.

### 2.2.2 Sensores

Os sensores, de forma corriqueira, são definidos como dispositivos que detectam grandezas físicas, tais como: temperatura, pressão, luz etc e as convertem em sinais eletrônicos proporcionais à grandeza observada. No entanto, convém esclarecer que os sensores realmente são dispositivos que detectam as variações de uma grandeza qualquer, mas quando correlacionam essa variação com outra grandeza, essa ação caracteriza esse dispositivo como transdutor. Apesar de os termos sensor e transdutor serem comumente confundidos, tal entendimento não traz nenhum prejuízo a função que os sensores exercem (Silva; Junior, 2015). Portanto, tomando por base esses autores, no decorrer desse trabalho, será adotado apenas o termo sensor.

Nesse sentido, será utilizado para desenvolver o medidor de água inteligente, um sensor de vazão (Figura 8), com as seguintes especificações técnicas:

- modelo YF-S201, rosca: 1/2";
- tensão de operação: 5V 18V (5V Nominal);
- pressão de operação:  $\leq 1,75$  MPa;
- faixa de vazão: 1 a 30 litros por minuto;

- pulsos: nível alto 4,5V / nível baixo 0,5V (quando alimentado a 5V);
- sinal de saída: pulsos conforme a velocidade do rotor interno;
- fios: preto (GND) vermelho (VCC) amarelo (Sinal de Saída);

Este modelo de sensor possui uma turbina em formato de hélice que gira a medida que a água passa, gerando pulsos elétricos que são captados e enviados ao Arduino. O sensor produz 450 pulsos por litro e utiliza o princípio do Efeito Hall (Côrte, 2020). De acordo com Silva e Junior (2015), os sensores de efeito Hall detectam a presença de um campo magnético. Dessa forma, quando a água flui, a hélice gira, e o sensor de efeito Hall registra o volume de água que passa a cada rotação, permitindo o controle do consumo de água através do cano(Côrte, 2020).

Figura 8 – Sensor



Fonte: Adaptado de Silva e Junior (2015)

Em síntese, a fundamentação teórica apresentada neste tópico forneceu as teorias e o conhecimento dos autores, sobre IoT, sistemas embarcados e comunicação entre dispositivos, que foram essenciais para o desenvolvimento do dispositivo inteligente proposto neste trabalho.

### 2.2.3 *Hardware-In-The-Loop (HIL)*

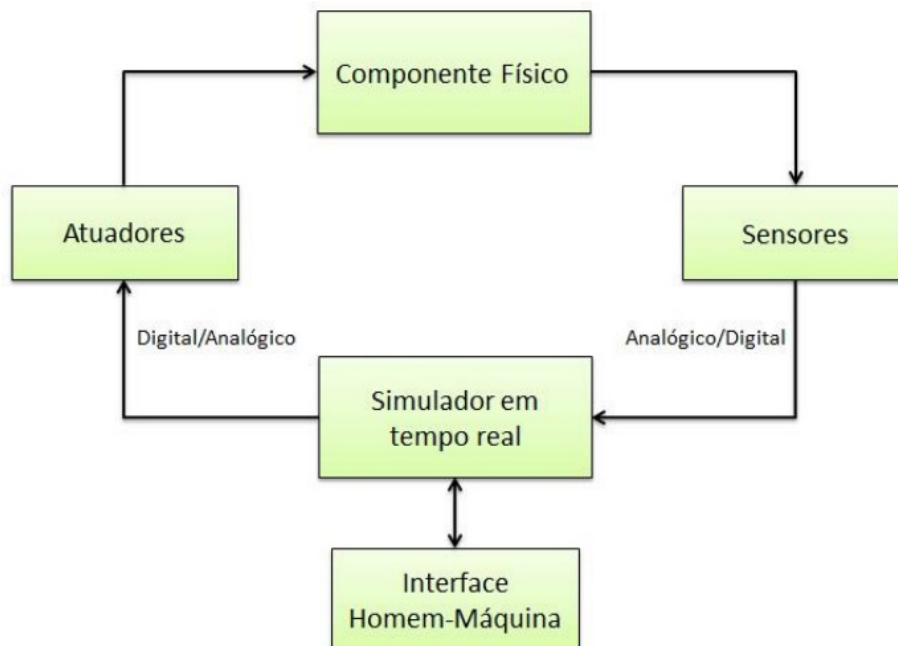
A técnica Hardware-in-the-Loop (HIL) é amplamente reconhecida como um método padrão para testar sistemas embarcados antes de sua implantação final. Nessa abordagem, componentes de um sistema real são substituídos por simulações computacionais, permitindo a combinação de elementos reais e simulados em uma configuração que possibilita a simulação e o teste dos componentes reais de um sistema (Costalonga, 2020).

Segundo Lopes (2017), geralmente um sistema HIL apresenta os componentes conforme esquematizados na Figura 9 e descritos detalhadamente a seguir:

- **Componente físico:** Parte do ambiente real que interage com o modelo simulado por meio de sensores e atuadores.
- **Atuadores e sensores:** Dispositivos responsáveis por medir ou transmitir sinais do processo.
- **Simulador em tempo real:** Inclui um sistema de aquisição de sinais (placa) e uma unidade de controle que processa esses sinais, comunicando-os com a interface gráfica e o sistema de aquisição.
- **Interface homem-máquina (IHM):** Permite visualizar os diversos sinais do processo através de uma interface gráfica, possibilitando a interação e a alteração de parâmetros e características em tempo real.

As vantagens do uso da técnica é identificado por Costalonga (2020), que destaca como benefícios do HIL: a redução dos custos do projeto e dos riscos à segurança; a possibilidade de testes em diferentes condições, sem a necessidade de construção de protótipos físicos na fase inicial do projeto; além de oferecer uma análise mais abrangente do sistema, permitindo a identificação precoce de falhas e erros no projeto, o que contribui para a melhoria da qualidade e confiabilidade antes da sua finalização.

Figura 9 – Componentes do HIL



Fonte: Lopes (2017)

### 3 TRABALHOS RELACIONADOS

A pesquisa nessa etapa teve o propósito de identificar trabalhos relacionados à temática em estudo, no caso, o desenvolvimento de medidor de água inteligente. Dentre os trabalhos que apresentaram afinidade com a proposta desta pesquisa, identificou-se os seguintes trabalhos em ordem cronológica: (Brandão *et al.*, 2019); (Côrte, 2020); (Medeiros *et al.*, 2021) e (Mattos; Conceição, 2022).

Brandão *et al.* (2019) apresentou um trabalho por meio do qual propôs o desenvolvimento de um dispositivo de medição inteligente de água de baixo custo, visando o monitoramento automatizado do consumo de água dos prédios do campus da Universidade Federal de Campina Grande. Para o desenvolvimento do *hardware* foi utilizado um microcontrolador tipo NodeMCU com Wi-Fi integrado e um sensor de fluxo foi do tipo turbina com contagem de pulsos. O *software* foi desenvolvido utilizando a linguagem de programação do próprio Arduino. Para armazenar as informações coletadas pelo hidrômetro digital e disponibilizá-las em tempo real para os usuários monitorarem o consumo de água foi criado um site em linguagem HTML desenvolvido em parceria com o curso de Ciência da Computação da própria universidade. Quanto aos resultados, estes foram positivos, considerando a durabilidade, estabilidade no envio de dados e o baixo custo do dispositivo inteligente desenvolvido.

Já o trabalho de Côrte (2020) tem o objetivo de desenvolver um sistema de gerenciamento autônomo de medidores de consumo de água em residências. Para tanto, desenvolveu um dispositivo IoT, utilizando placa arduino, sensor de fluxo de água e Protocolo de comunicação sem fio entre dispositivos inteligentes denominado ZigBee, para comunicação do Arduino com o servidor *Central Internet of Things* (CIoT), que trata-se de um servidor local desenvolvido pela UFSC que controla vários subsistemas e dispositivos de uma casa inteligente dentro do campus. A pesquisa concluiu que o protótipo produzido foi capaz de obter medições simples necessárias para implementar todas as outras funcionalidades propostas, mas sugere a substituição do sensor adotado por outro mais robusto, considerando as taxas de erro observadas durante a pesquisa.

Medeiros *et al.* (2021) apresentaram o desenvolvimento de um protótipo inteligente para monitoramento e controle de vazamentos de água remoto, constituído por um sensor de fluxo tipo turbina com contagem de pulso e um controlador lógico programável com wi-fi integrado (NodeMCU) com programação em linguagem Arduino IDE, cujos os dados coletados foram armazenados em uma planilha Google para acesso remoto e em tempo real. Após concluído, o protótipo foi instalado no laboratório do Departamento de Pavimentação da Universidade Federal



de Campina Grande para monitoramento do consumo de água. Os resultados mostraram que o protótipo inteligente é viável para a detecção de vazamentos e monitoramento do consumo, contribuindo positivamente para uma gestão mais eficiente de água nas edificações.

Mattos e Conceição (2022), retrata o desenvolvimento de um hidrômetro inteligente, como o objetivo de diminuir o consumo e custo de água nas residências, utilizando um sensor de fluxo e placa Arduino UNO entre outros dispositivos eletrônicos necessários para a construção do protótipo. A proposta foi de um dispositivo de fácil instalação (tipo Plug Play) que pode ser colocado em diversas áreas como cozinha, banheiro, área de serviço e entre outros, fornecendo ao usuário o consumo individual de cada área, sem a necessidade de alterar a infraestrutura do local. Por fim, as autoras concluíram que os objetivos foram alcançados, fornecendo ao usuários os dados de consumo de água de forma local e remota.

Conforme exposto, pode-se observar que todos os trabalhos apresentados estão em consonância com a temática relativa ao uso de dispositivo inteligente para monitoramento do consumo de água, no qual observou-se entre os autores pontos de semelhanças e algumas particularidades.

Dito isso, têm-se que Brandão *et al.* (2019) e Medeiros *et al.* (2021) as similaridades no desenvolvimento do hardware ao utilizarem um microcontrolador tipo NodeMCU, enquanto que (Côrte, 2020) e (Mattos; Conceição, 2022) utilizaram a Placa Arduino, todos desenvolvidos no ambiente de desenvolvimento do Arduino. O diferencial entre as tecnologias é que o NodeMCU já vem com Wi-Fi integrado, diferentemente do Arduino. Entretanto, já existem placa Arduino compatível que possuem o modulo ESP8266 integrado.

Assim, a opção pela placa Arduino apresenta-se atrativa para o desenvolvimento deste trabalho, principalmente, devido ao seu baixo custo, ampla gama de aplicações, além de sua biblioteca padrão oferecer diversos recursos que simplificam os trabalhos de programação Monk (2014).

Para o armazenamento dos dados coletados pelo dispositivo inteligente, destaca-se o utilizado por Medeiros *et al.* (2021) que optaram por armazenar em planilha Google. De acordo com as referidas autoras, a escolha mostrou-se viável para armazenar e disponibilizar os dados em tempo real e remoto. Portanto com potencial de uso nesse trabalho.

Por fim, observa-se que Côrte (2020) sugere a substituição do sensor YF-S210 por outro com menor taxa de erro e maior robustez. Contudo, considera-se que o sensor atende às necessidades desta pesquisa, portanto passível de ser adotado. Principalmente, no que se refere à

faixa de medição de vazão e ao custo; além de estar amplamente mencionado nas publicações que abordam o tema desta pesquisa, como em Silva e Junior (2015), e Júnior e Farinelli (2018).

Quadro 2 – Trabalhos relacionados.

<b>Referência</b>	<b>Temática principal da referência</b>	<b>Similaridades com meu projeto</b>
Brandão <i>et al.</i> (2019)	Dispositivo inteligente de baixo custo para monitoramento do consumo de água em prédios	Uso do wi-fi integrado ao ao sensor de fluxo de água.
Côrte (2020)	Desenvolvimento de sistema para gerenciamento do consumo de água em residências.	Visão IoT, além da compreensão do funcionamento do sensor YF-S201 de efeito Hall, também uso da Placa Arduino e Módulo ESP8266
Medeiros <i>et al.</i> (2021)	Desenvolvimento de protótipo para monitoramento e controle de vazamento, com sensor de fluxo, NodeMCU e planilha Google para acesso dos dados em tempo real	Armazenamento de dados em planilha Google
Mattos e Conceição (2022)	Desenvolvimento de hidrômetro inteligente, para controle do consumo de água em residências com sensor de fluxo, Placa Arduino e de fácil instalação (Plug and Play).	Sensor de fluxo, Placa Arduino e dispositivo de fácil instalação (Plug and Play) sem necessidade de alterar a infraestrutura existente.

Fonte: Próprio autor.

Em suma, conclui-se que todos os trabalhos apresentam potencial para contribuir com o desenvolvimento deste trabalho, conforme observado na Quadro 2 que pretende utilizar Placa Arduino, Módulo wi-fi ESP8266, sensor de fluxo de água YF-S201 de efeito Hall e armazenamento dos dados em planilha Google, disponibilizados utilizando o Microsoft Power BI.

## 4 METODOLOGIA

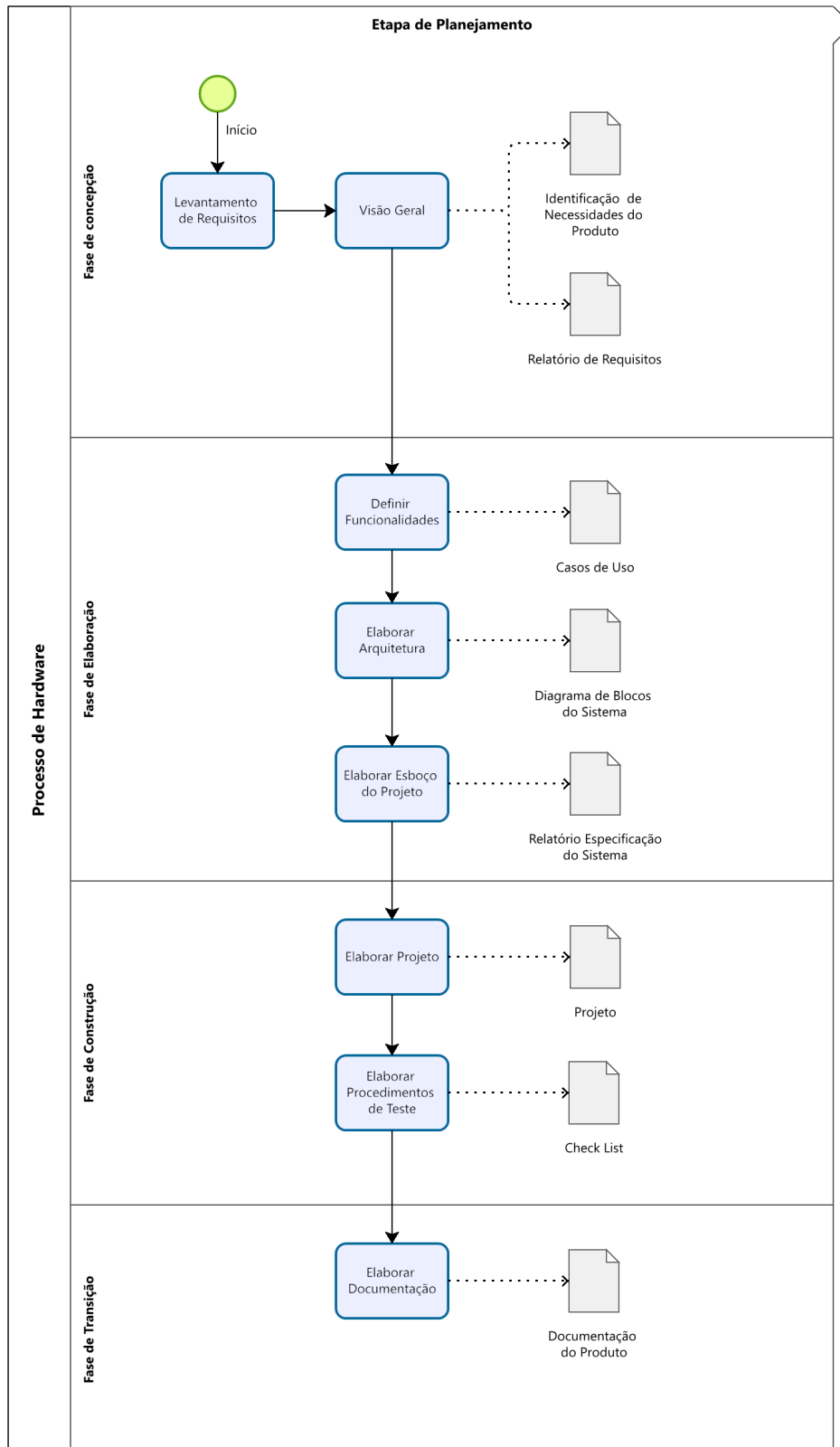
### 4.1 Concepção

O *Rational Unified Process* (RUP), usualmente denominado de Processo Unificado foi desenvolvido por executivos da Rational, Inc., e posteriormente adquirido pela IBM. Trata-se de uma metodologia para desenvolvimento de *software* criado para dar suporte ao desenvolvimento orientado a objetos e organizar o desenvolvimento de *software* que adota as fases de concepção, elaboração, construção e transição.

Este projeto, alinhado à metodologia do Processo Unificado, seguiu as fases mencionadas para o desenvolvimento do medidor de água, como ilustrado na Figura 10. Além disso, foi empregada a abordagem de modelagem visual *Unified Modeling Language* (UML) conhecida como Linguagem de Modelagem Unificada, que possibilita aos usuários criar representações gráficas do produto. Essa abordagem oferece a vantagem de permitir a visualização dos objetos e das interações entre eles.

Além do mais, fez-se uso de desenvolvimento iterativo e incremental. Nesse sentido, Rubin (2017) explica que o desenvolvimento iterativo é uma abordagem que envolve retrabalho planejado. Assim, são realizadas várias iterações sobre o que está sendo construído, permitindo refinar e melhorar gradualmente a solução até alcançar um bom resultado. Já o incremental, em vez de construir o produto de uma só vez, o produto é dividido em partes menores, permitindo que cada uma seja construída separadamente. Isso possibilita aprender como cada parte funciona no ambiente em que será utilizada, fazer ajustes com base nesse aprendizado e, em seguida, continuar construindo outras partes. A partir desses conhecimentos, deu-se início a fase de concepção que consiste na compreensão plena do campo de aplicação do produto para poder se construir um modelo de negócio que atenda aos propósitos desejados (Schach, 2009). Portanto, faz parte dessa fase o levantamento dos requisitos necessários para se ter uma visão geral do produto, cujos resultados desse levantamento serão descritos detalhadamente e/ou diagramados, de forma clara e precisa, com as especificações do produto que se pretende desenvolver.

Figura 10 – Fluxo do Processo



Para tanto, foi realizado um levantamento por meio de entrevista junto a moradores de uma residência na qual foi identificada a preocupação com o risco de vazamentos e do consumo excessivo de água. Também, foi relatado experiências anteriores com contas de água elevadas e dificuldades na identificação da origem do problema. Além disso, os moradores expressaram o desejo de monitorar o consumo diário para entender melhor os hábitos de uso de água e identificar oportunidades para reduzir o desperdício.

A partir desse levantamento, foi possível elencar dois principais pontos de interesse: estimativa de consumo para o mês e alerta de vazamento (ex: constante ou demasiado).

Segundo Reinehr (2020), em projetos simples que possuem um usuário bem definido e perfis conhecidos, os requisitos podem ser obtidos por contado direto com os próprios usuários. A autora, ainda complementa que na fase de concepção os primeiros requisitos se apresentam de forma mais abrangente atrelados aos objetivos gerais do negócio e não se sabe exatamente o que precisa ser construído, até que se conheçam os requisitos que representam as funcionalidades do produto.

#### **4.1.1 Requisitos funcionais e não funcionais**

Os requisitos que compõem a solução proposta neste trabalho são apresentados divididos entre requisitos funcionais e não funcionais, conforme a seguir.

Por requisitos funcionais, entende-se como sendo as funcionalidades que o sistema deve fornecer para atender as demandas dos usuário. Assim, foram definidos no Quadro 3:

Quadro 3 – Requisitos Funcionais

<b>Identificação</b>	<b>Requisito Funcional</b>
<b>RF1</b>	Identificar o consumo de água da residência.
<b>RF2</b>	Disponibilizar as informações de consumo de água da residência.
<b>RF3</b>	Disponibilizar gráfico de consumo diário e mensal.
<b>RF4</b>	Alertar sobre a possibilidade de vazamento.

Fonte: Próprio autor

Já os requisitos não funcionais enfatizam as configurações e as particularidades do sistema, tais como segurança, tempo de resposta, implementação, hardware, entre outros. (Santos *et al.*, 2021), identificados, nesse trabalho, conforme Quadro 4.

Quadro 4 – Requisitos Não Funcionais

<b>Identificação</b>	<b>Requisito Não Funcional</b>
<b>RNF1</b>	Adotar plataforma Arduino e ESP para processar e transmitir dados.
<b>RNF2</b>	Utilizar sensor de vazão.
<b>RNF3</b>	Utilizar serviço Power BI Service gratuito.
<b>RNF4</b>	Usar conexão Wi-Fi da ESP.
<b>RNF5</b>	Definir 10 segundos como intervalo de comunicação entre o sensor e o Arduino.
<b>RNF6</b>	Detectar fluxo contínuo que ultrapasse 1 hora.

Fonte: Próprio autor

Ao final dessa etapa, o levantamento possibilitou definir o escopo de atuação do sistema a ser detalhado na etapa seguinte.

## 4.2 Elaboração

Na fase de elaboração, foram realizados os procedimentos necessários para a análise do projeto, que compreendem planejamento e modelagem a partir da elaboração dos casos de uso e especificações. Além disso, incluiu a estrutura do projeto, bem como início das codificações.

### 4.2.1 Casos de Uso

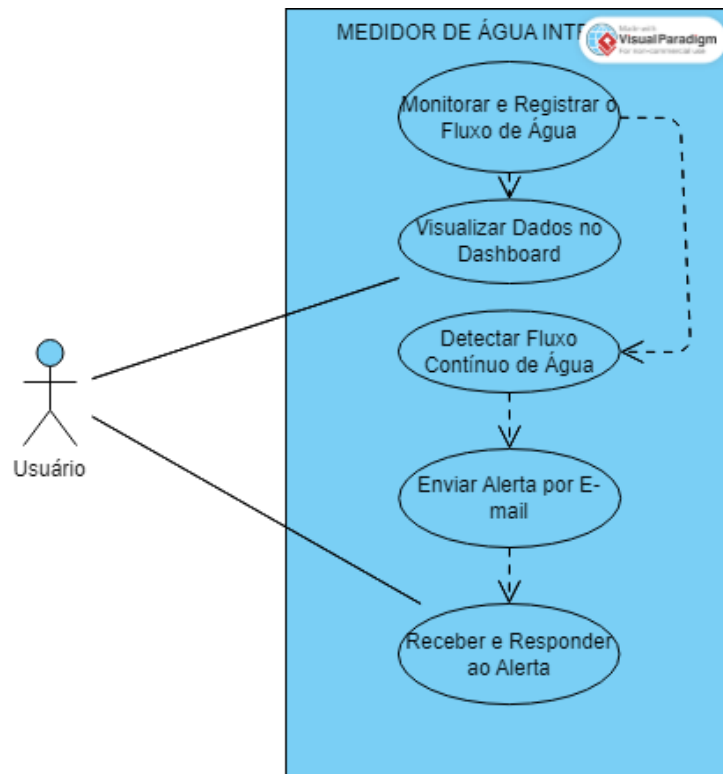
Os casos de uso são uma técnica para identificar os requisitos funcionais de um sistema. Eles servem para descrever as interações entre os usuários de um sistema e o próprio sistema, capturando as funcionalidades que o sistema deve oferecer para atender às necessidades do usuário. (Fowler, 2011)

Nesse trabalho, a Figura 11 traz uma representação gráfica das interações do usuário com o projeto que está sendo desenvolvido.

### 4.2.2 Especificação dos Casos de Uso

Na sequência estão descritos as especificações dos casos de uso definidos anteriormente, observados no Quadro 5.

Figura 11 – Caso de uso



Fonte: Próprio autor

Quadro 5 – Especificação dos casos de uso

Caso de Uso	Descrição
<b>Detectar o fluxo de água</b>	<p><b>Ator:</b> Sistema</p> <p><b>Descrição:</b> O sistema identifica o fluxo de água através do sensor conectado ao Arduino.</p> <p><b>Pré-condições:</b> O sistema está ligado e configurado corretamente.</p> <p><b>Fluxo Principal:</b></p> <ul style="list-style-type: none"> <li>– O Arduino lê os pulsos do sensor de fluxo.</li> <li>– O Arduino calcula o fluxo de água e envia os dados para a ESP a cada 10 segundos.</li> <li>– A ESP recebe os dados e os envia para uma planilha do Google.</li> </ul> <p><b>Fluxo Alternativo:</b></p> <ul style="list-style-type: none"> <li>– Se não houver fluxo, o Arduino ainda envia leituras indicando ausência de fluxo.</li> </ul> <p><b>Pós-condições:</b> A ESP possui os dados do fluxo de água para análise e os armazena na planilha do Google.</p>

<p><b>Detectar fluxo contínuo de água</b></p>	<p><b>Ator:</b> Sistema</p> <p><b>Descrição:</b> A ESP monitora os dados do Arduino e detecta se há um fluxo contínuo de água por mais de 1 hora.</p> <p><b>Pré-condições:</b> A ESP está recebendo dados do Arduino.</p> <p><b>Fluxo Principal:</b></p> <ul style="list-style-type: none"> <li>– A ESP recebe os dados do Arduino e verifica se há fluxo contínuo.</li> <li>– Se a ESP detectar que o fluxo contínuo excede 1 hora, ela registra o tempo do fluxo.</li> </ul> <p><b>Fluxo Alternativo:</b></p> <ul style="list-style-type: none"> <li>– Se o fluxo for interrompido por pelo menos 1 minuto, o contador de fluxo contínuo é resetado.</li> </ul> <p><b>Pós-condições:</b> O sistema sabe se o fluxo contínuo excedeu 1 hora.</p>
<p><b>Enviar alerta por e-mail</b></p>	<p><b>Ator:</b> Sistema</p> <p><b>Descrição:</b> A ESP envia um e-mail de alerta se detectar que o fluxo de água está contínuo por mais de 1 hora.</p> <p><b>Pré-condições:</b> A ESP está conectada à internet.</p> <p><b>Fluxo Principal:</b></p> <ul style="list-style-type: none"> <li>– A ESP verifica se o fluxo contínuo de água excedeu 1 hora.</li> <li>– A ESP envia um e-mail de alerta para o destinatário configurado.</li> <li>– A ESP espera 1 hora antes de enviar outro e-mail, caso o fluxo não seja interrompido.</li> </ul> <p><b>Fluxo Alternativo:</b></p> <ul style="list-style-type: none"> <li>– Se o fluxo for interrompido, a ESP para de enviar e-mails.</li> </ul> <p><b>Pós-condições:</b> Um e-mail de alerta é enviado, notificando o fluxo contínuo de água.</p>



<p><b>Receber e responder ao alerta</b></p>	<p><b>Ator:</b> Usuário/Técnico</p> <p><b>Descrição:</b> O usuário recebe o e-mail de alerta e toma as medidas necessárias para interromper o fluxo de água.</p> <p><b>Pré-condições:</b> O e-mail de alerta foi enviado e recebido.</p> <p><b>Fluxo Principal:</b></p> <ul style="list-style-type: none"> <li>– O usuário recebe o e-mail de alerta.</li> <li>– O usuário verifica o sistema para identificar a causa do fluxo contínuo.</li> <li>– O usuário toma as ações necessárias para interromper o fluxo.</li> </ul> <p><b>Pós-condições:</b> O fluxo de água é interrompido, e o sistema para de enviar alertas.</p>
<p><b>Acessar dados pelo dashboard</b></p>	<p><b>Ator:</b> Usuário</p> <p><b>Descrição:</b> O usuário acessa os dados do fluxo de água em tempo real através de um dashboard.</p> <p><b>Pré-condições:</b> O sistema está em funcionamento e coletando dados do fluxo de água.</p> <p><b>Fluxo Principal:</b></p> <ul style="list-style-type: none"> <li>– O usuário abre o dashboard através de um navegador ou aplicativo dedicado.</li> <li>– O usuário visualiza os dados do fluxo de água em tempo real.</li> <li>– O usuário pode analisar os dados históricos e detectar padrões ou anomalias.</li> </ul> <p><b>Pós-condições:</b> O usuário tem acesso aos dados de monitoramento do fluxo de água, permitindo uma análise detalhada do sistema.</p>

Fonte: Próprio autor

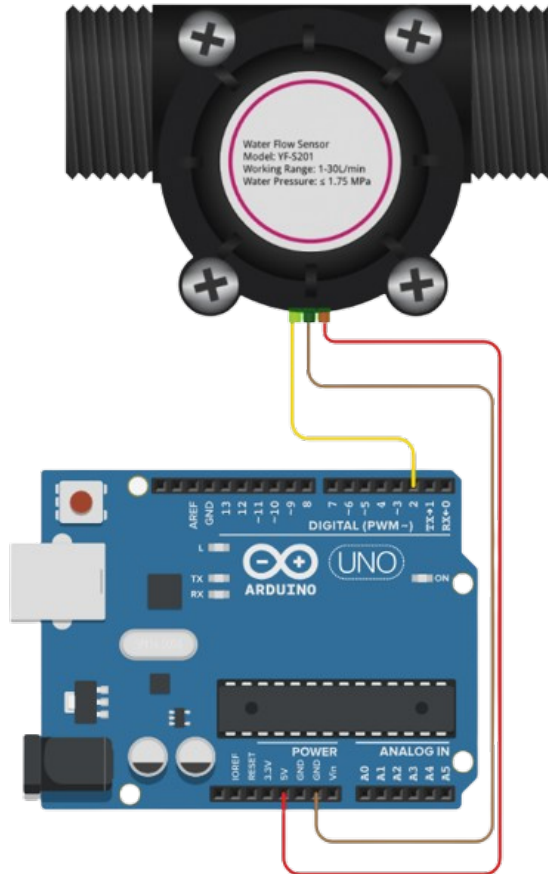
#### 4.2.3 Esquemático do projeto

O esquemático do projeto (Figura 12) é o diagrama que mostra como os componentes do projeto estão conectados eletricamente, mostrando como ocorre a ligação entre o Arduino e o sensor de vazão. No caso específico deste projeto o sensor utilizado foi o YF-S201, rosca 1/2".

Conforme pode ser observado, o sensor possui três fios: o fio marrom estabelece

a conexão entre os terminais GND (terra) do Arduino e do sensor. O fio vermelho fornece a alimentação de 5V ao sensor, conectando-se ao terminal VCC. Por fim, o fio amarelo transmite o sinal digital do sensor para o pino 2 do Arduino.

Figura 12 – Sistema Arduino + Sensor de fluxo



Fonte: Próprio autor

### 4.3 Construção

Nesta etapa, foi realizada a construção do medidor de água, como demonstrado nos fluxos apresentados a seguir, que detalham todo o processo de desenvolvimento do dispositivo de forma iterativa e incremental. Esse método permitiu a construção e o refinamento gradual do equipamento, com base em observações feitas ao longo de todo o processo, possibilitando os ajustes necessários até a sua conclusão.

#### 4.3.1 Montagem do medidor de água

Para dar início à construção do medidor de água, foram utilizados os componentes listados a seguir, incluindo o Arduino, o sensor de vazão e outros componentes, de acordo com o

esquemático do projeto apresentado na fase anterior.

- Arduino: o microcontrolador principal que será responsável por ler os dados do sensor e processar a informação.
- Sensor de vazão modelo YF-S201, rosca 1/2": esse sensor é responsável por medir a quantidade de água que passa através dele.
- Fiação, conexões e cabo de energia: fios e cabo para conectar o sensor ao Arduino.

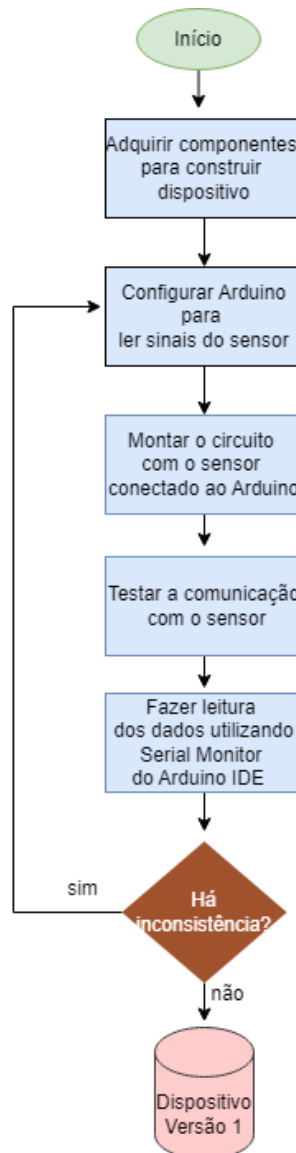


Figura 13 – Fluxo: Montar projeto  
Fonte: Próprio autor

O primeiro passo foi ajustar os *switches* responsáveis pela configuração da placa. De acordo com as especificações da placa, ver Figura 14, para configurar o Arduino, os *switches* 3 e 4 devem ser colocados em ON, enquanto os demais permanecem em OFF.

Após esses ajustes, o passo seguinte foi configurar o Arduino para ler os pulsos

Figura 14 – Configuração de *Switches*

	1	2	3	4	5	6	7	8
CH340 connect to ESP8266 (upload sketch)	OFF	OFF	OFF	OFF	ON	ON	ON	NoUSE
CH340 connect to ESP8266 (connect)	OFF	OFF	OFF	OFF	ON	ON	OFF	NoUSE
CH340 connect to ATmega328 (upload sketch)	OFF	OFF	ON	ON	OFF	OFF	OFF	NoUSE
Mega328+ESP8266	ON	ON	OFF	OFF	OFF	OFF	OFF	NoUSE
All modules work independent	OFF	OFF	OFF	OFF	OFF	OFF	OFF	NoUSE

Fonte: Fabricante

enviados pelo sensor conectado à porta D2 e calcular o fluxo de água com base nesses pulsos. Conforme as especificações do sensor, 450 pulsos correspondem a 1 litro de água.

Em seguida, foram feitos testes básicos para verificar a comunicação entre sensor e o Arduino, utilizando o Monitor Serial para visualização dos dados.

Os principais passos dessa etapa podem ser observado na Figura 13 e os detalhes da codificação no Apêndice 6.1.

#### 4.3.2 *Calibração do sensor*

Nesta etapa da construção, foram realizados aperfeiçoamentos dos dados e cálculos do volume para garantir a precisão dos dados medidos pelo sensor. Assim, para calcular o volume de água que é captado pelo sensor, definiu-se que a cada 10 segundos, o código verifica se houve fluxo de água. Caso positivo, ele calcula o volume total de água que passou, convertendo os pulsos em litros.

Após teste, verificou-se o volume medido pelo Arduino era inferior ao volume real. Dessa forma, foi necessário introduzir um fator de correção para corrigir a discrepância e aumentar a precisão da medição.

Para tanto, foram realizados testes, com base nos requisitos definidos anteriormente. Durante esse período de teste, foi necessário validar a precisão do medidor e garantir que os dados coletados e armazenados estejam consistentes.

Os testes consistiram em comparar o volume lido pelo sensor e a quantidade de água a ser registrada. Inicialmente, o valor registrado foi de 730ml, no entanto, o valor real foi de 786 ml, havendo assim uma discrepância entre os valores. Para a correção das alterações observadas, foi adicionado uma variável alpha no código do arduino que corrige os valores registrados para

que se tenha um valor desejado.

Analisado por um cálculo de regra de 3, a razão do valor real pelo valor obtido foi de 1,077. Sendo assim, uma variável foi criada para corrigir os valores, no caso desse teste a correção será multiplicando o valor obtido por 1.077, assim para esse trabalho específico o fator de correção é 1.077.

Após a correção de valores, foi realizado um novo teste no qual verificou-se que um valor de água real de 800ml teve o registro de 810ml. Portanto, apresentou um valor bem próximo do real, atendendo aos objetivos do projeto.

Os dados obtidos podem ser observados na planilha do Google Sheets, mostrada na Figura 15.

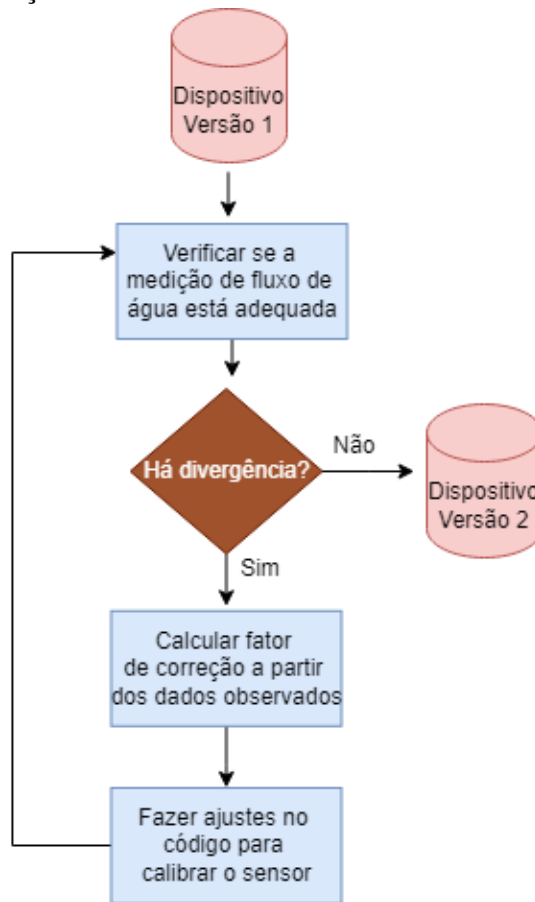
Figura 15 – Planilha

	A	B	C	D	E
1	01/09/2024 20:24:50	0,55			
2	01/09/2024 20:25:00	0,46			
3	01/09/2024 20:28:31	0			
4	01/09/2024 20:58:01	0,38			
5	02/09/2024 12:13:55	0,01			
6	02/09/2024 12:14:04	0,43			
7	02/09/2024 12:14:14	0,1			
8	02/09/2024 12:17:14	0,01			
9	02/09/2024 12:26:04	0,24			
10	02/09/2024 12:26:17	0,02			
11	02/09/2024 12:26:34	0,19			
12	02/09/2024 12:26:44	0,02			
13	02/09/2024 12:37:36	0,01			
14	02/09/2024 12:37:54	0,01			
15	02/09/2024 12:45:44	0			
16	02/09/2024 12:45:57	0,01			
17	02/09/2024 12:55:15	0			
18	02/09/2024 12:55:25	0,02			
19	02/09/2024 12:55:34	0,02			
20	02/09/2024 13:06:45	0,03			
21	02/09/2024 16:25:45	0,42			
22	02/09/2024 18:07:59	0,31			
23	02/09/2024 18:08:32	0,89			
24	02/09/2024 18:11:03	0,52			
25	02/09/2024 18:11:21	0,38			
26	02/09/2024 18:12:14	0,79			

Fonte: Próprio autor

Para um melhor visualização dessa etapa, a Figura 16 traz um resumo dos procedimentos realizados.

Figura 16 – Fluxo de calibração do sensor



Fonte: Próprio autor

#### 4.3.3 Integração com comunicação via Wi-Fi (ESP8266)

Essa etapa tem o objetivo de expandir a funcionalidade do dispositivo, conectando-o à internet o que permitirá o envio de dados remotamente. Para tanto, a ESP8266 foi configurada para receber os dados do fluxo de água do Arduino e enviar esses dados para uma planilha do Google, podendo os detalhes da codificação serem observados no Apêndice 6.2. Para essa configuração, inicialmente, foram utilizados os ajustes dos *switches* 5, 6 e 7 em ON, com os demais em OFF, para a inserção do código na placa. Para a execução, os *switches* 5 e 6 permaneceram em ON, enquanto os outros foram ajustados para OFF.

Outra configuração envolveu os *switches* 7 e 8 em OFF, com os demais em ON, permitindo o uso do Arduino pelos pinos 3 e 4, com os *switches* 5 e 6 para a ESP. Além disso, os *switches* 1 e 2 foram ajustados para habilitar a comunicação serial entre as placas de forma interna, eliminando a necessidade de conexões físicas com fios ou jumpers, conforme mostrado na Figura 14.

Essa última configuração foi a mais utilizada na execução, pois a ESP depende dos

dados fornecidos pelo Arduino, cujo fluxo de trabalho é descrito a seguir.

- **Conexão à rede Wi-Fi:** No início do código (`setup()`), o ESP8266 se conecta a uma rede Wi-Fi utilizando as credenciais fornecidas (`ssid` e `password`). A execução do código só continua após a conexão ser estabelecida com sucesso.
- **Recebimento de dados via porta serial:** O Arduino está conectado ao ESP8266 através da porta serial. O Arduino envia dados para essa porta serial compartilhada, e o ESP8266 os recebe para processar. Esses dados podem incluir leituras de sensores, como o volume de água medido por um sensor de fluxo.
- **Envio de dados para Google Sheets:** Quando o ESP8266 recebe dados do Arduino pela porta serial, ele os envia para uma planilha do Google através de uma requisição HTTP segura (HTTPS) para um servidor Google Apps Script. O script é responsável por processar e registrar os dados na planilha.
- **Monitoramento de fluxo:** O código monitora o fluxo de água ao verificar se ele foi interrompido por mais de 1 minuto. Caso o fluxo seja interrompido por esse tempo, uma flag interna é acionada para indicar essa interrupção.

Em resumo, o Arduino coleta e envia os dados para o ESP8266 pela porta serial compartilhada, e o ESP8266 processa esses dados, registrando-os em uma planilha do Google e enviando e-mails de alerta caso o fluxo de água indique um possível vazamento.

#### 4.3.4 Planilha Google

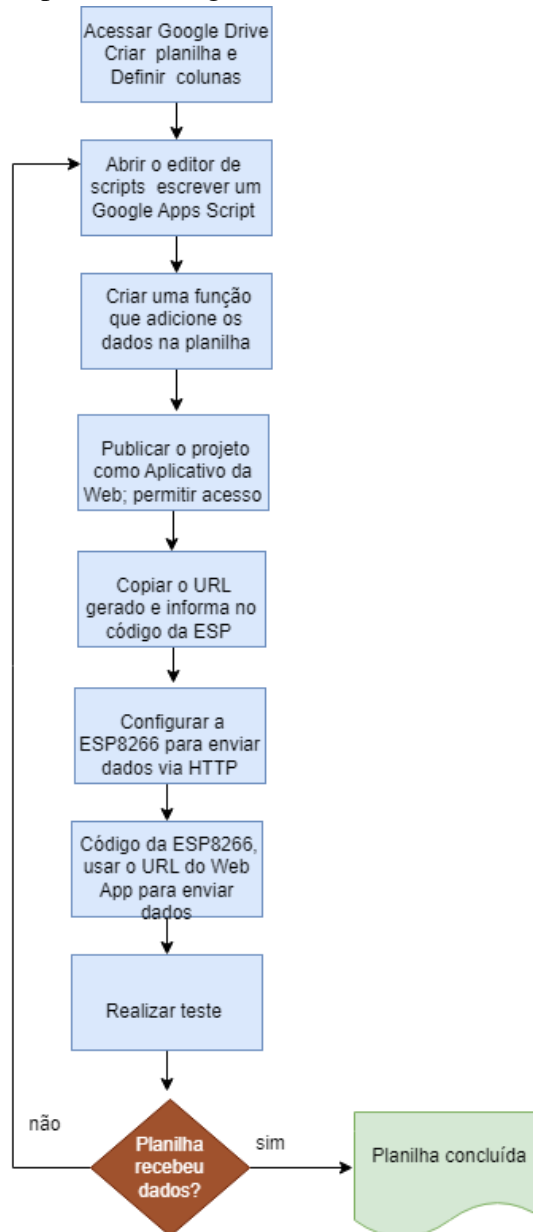
Foi utilizada uma planilha do Google para armazenar os dados recebidos da ESP8266. A integração entre a ESP8266 e a planilha do Google foi possível graças ao código desenvolvido no Apêndice 6.3.

A Figura 17 apresenta um diagrama simplificado desse processo, descrito a seguir:

1. **Criar um Script de Google Apps:** Primeiro, você cria um script no Google Apps Script que define como os dados serão processados e inseridos na planilha. Esse script fica vinculado à sua planilha do Google.
2. **Publicar o Script como Web App:** Depois, você publica o script como um *Web App*. Isso gera um URL que pode ser acessado por outros aplicativos ou dispositivos para enviar dados.
3. **Enviar Dados para o Web App:** Seu dispositivo ou aplicativo faz uma requisição para o URL do *Web App*, enviando os dados que você deseja adicionar à planilha.

4. **Processamento dos Dados:** O *Web App* recebe os dados e executa a lógica definida no script para inserir esses dados na planilha.

Figura 17 – Fluxo criação da planilha Google



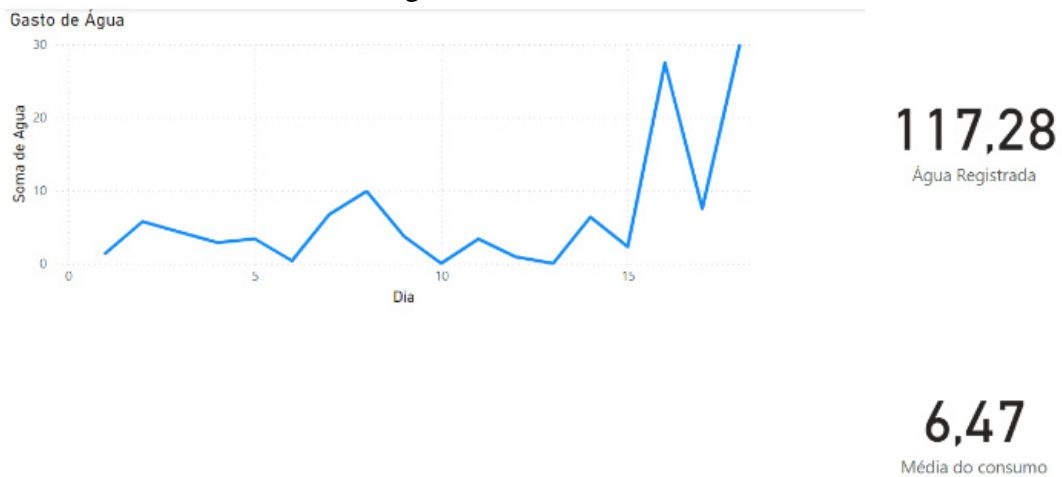
Fonte: Próprio autor

#### 4.3.5 Visualização dos dados

Nesta etapa, os dados coletados pelo medidor de água são apresentados em um dashboard, permitindo que os usuários acompanhem remotamente o consumo em tempo real. Essa interface visual facilita a interação entre o usuário e o medidor de água, fornecendo informações precisas e atualizadas sobre o consumo, conforme pode ser observado na Figura 18.



Figura 18 – Gráfico de consumo de água

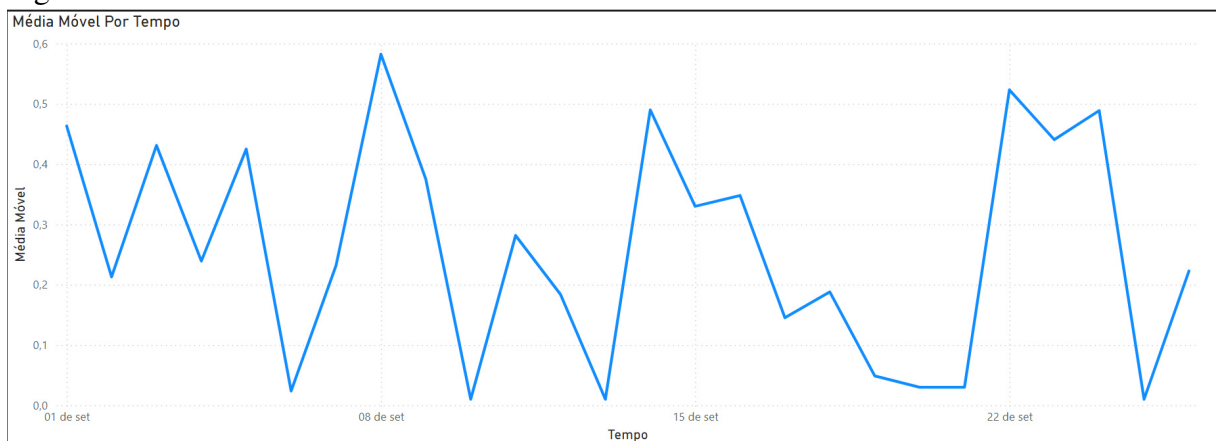


Fonte: Próprio autor

No dashboard, além do consumo em tempo real, foi desenvolvido um gráfico de linhas que utiliza a técnica de média móvel simples para calcular a média de consumo dos últimos 7 dias. Essa técnica suaviza as flutuações diárias, evidenciando padrões de consumo ao longo da semana e, ao utilizar um período de 7 dias, os usuários podem antecipar tendências e tomar decisões para controlar os gastos mensais, conforme é apresentado na Figura 19.

Ao analisar o gráfico da Figura 19, percebe-se um padrão de consumo estável ao longo do período estudado, indicando uma variação pequena no consumo semanal. Isso significa que não houve eventos anormais como vazamentos que tenham causado picos de consumo.

Figura 19 – Gráfico Consumo Média Móvel



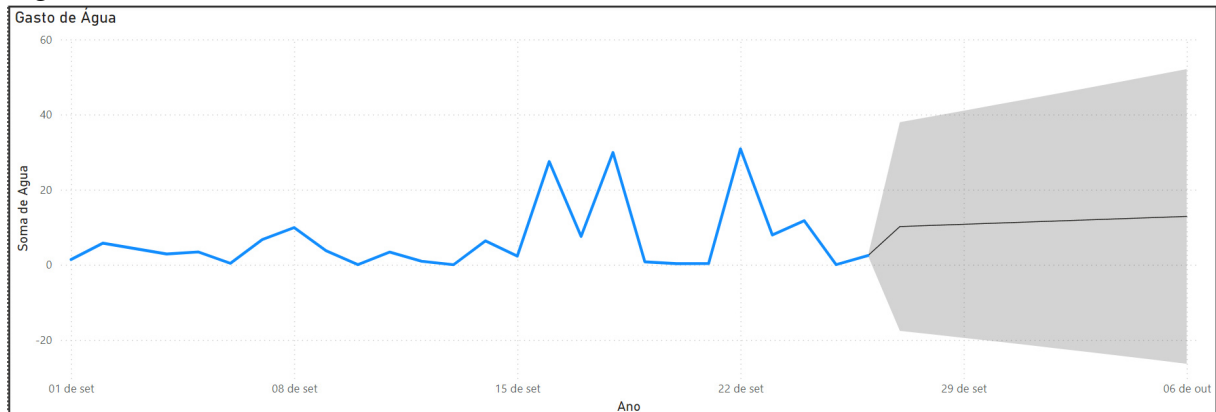
Fonte: Próprio autor

O Power BI também oferece funcionalidades avançadas para análise de séries temporais, utilizando algoritmos para identificar tendências e sazonalidades nos dados (Power-BI,

2023).

Essa ferramenta permite criar gráficos de previsões de forma automatizada, conforme mostrado na Figura 20, que apresenta um exemplo de previsão de consumo de água gerada pelo Power BI. A área destacada no gráfico representa a previsão de consumo para os próximos períodos, indicando um crescimento gradual com base nos dados históricos analisados.

Figura 20 – Gráfico de Consumo Médio e Previsão



Fonte: Próprio autor

Em suma, o usuário pode identificar um aumento gradual na média móvel, analisar as previsões de consumo e tomar medidas para otimizar o uso da água, como verificar vazamentos ou adotar hábitos mais conscientes.

#### 4.3.6 Configuração da ESP8266 para envio de e-mail

O propósito dessa fase é acrescentar uma camada de inteligência ao dispositivo, permitindo o envio de alerta de fluxo contínuo.

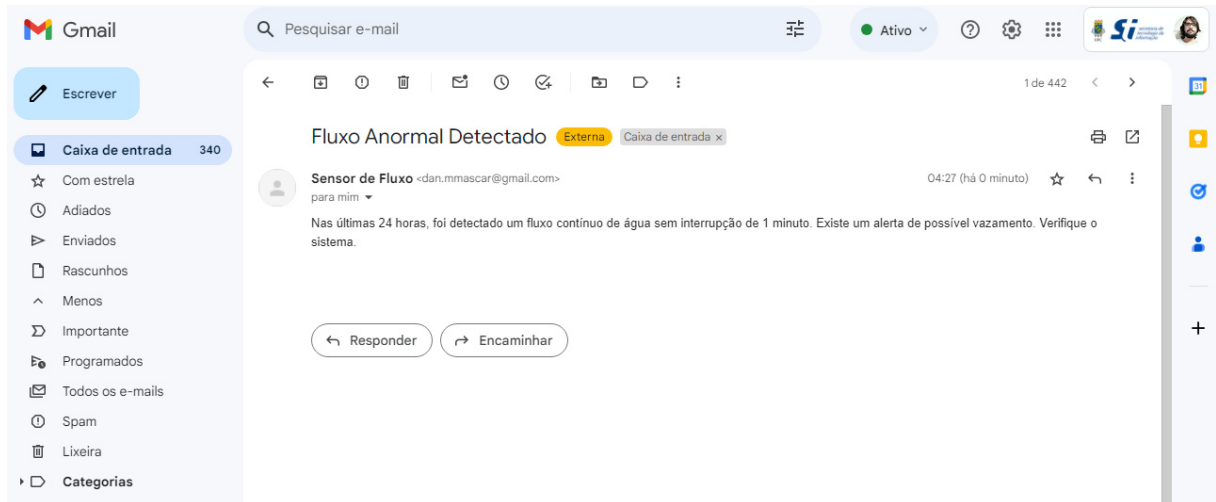
Para atender a essa funcionalidade, foi adicionada a configuração da ESP8266 novas condições:

- **Envio de alerta por e-mail:** Se o fluxo de água continuar ininterrupto por 24 horas, o ESP8266 envia um e-mail de alerta ao destinatário configurado. Esse alerta sugere um possível vazamento de água no sistema. O envio do e-mail é feito utilizando o protocolo SMTP e os detalhes da conta de e-mail configurada no código.
- **Reenvio do e-mail após 1 hora:** Se o fluxo continuar ininterrupto por mais uma hora após o envio do primeiro e-mail, o sistema envia outro e-mail de alerta para o usuário, garantindo que o problema continue sendo notificado até que o fluxo anômalo seja resolvido.

Após adicionada a nova funcionalidade, foram realizados testes para garantir que os

e-mails estavam sendo enviados de forma correta, conforme observado na Figura 21.

Figura 21 – E-mail



Fonte: Próprio autor

Concluídas as etapa de construção, o medidor de consumo de água foi utilizado por períodos mais extensos para a coleta de dados. Durante esse processo, constatou-se que a bateria de 9V não atendia à demanda energética para os intervalos requeridos, sendo necessária a utilização de uma fonte de alimentação externa conectada à rede elétrica.

#### 4.4 Transição

A fase de transição corresponde à etapa final do ciclo de desenvolvimento do produto no Processo Unificado e objetiva assegurar que o produto desenvolvido esteja adequado para utilização pelos usuários.

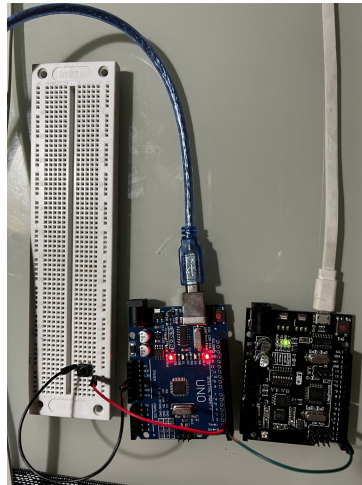
Sendo assim, foram realizados testes simulados para garantir que o medidor de consumo de água atenda às expectativas dos usuários. Dessa forma, os testes avaliaram a precisão das medições e a capacidade do sistema de fornecer dados em tempo real e de forma remota, para permitir um acompanhamento adequado do consumo de água.

##### 4.4.1 Testes e Validação

Para validação do medidor de água desenvolvido foram realizadas simulações utilizando um segundo microcontrolador (Arduino Uno), conforme mostrado na Figura 22.

O objetivo desse procedimento é identificar possíveis perdas de dados, erros de conexão ou atrasos na transmissão.

Figura 22 – Simulador de dados



Fonte: Próprio autor

Para isso, foi elaborado um plano de teste que consistiu no envio de 200 pulsos digitais para o pino D2 da placa do medidor de água principal. O segundo Arduino foi configurado para simular o envio desses pulsos, e o recebimento dos dados foi monitorado conforme pode ser observado na Figura 23.

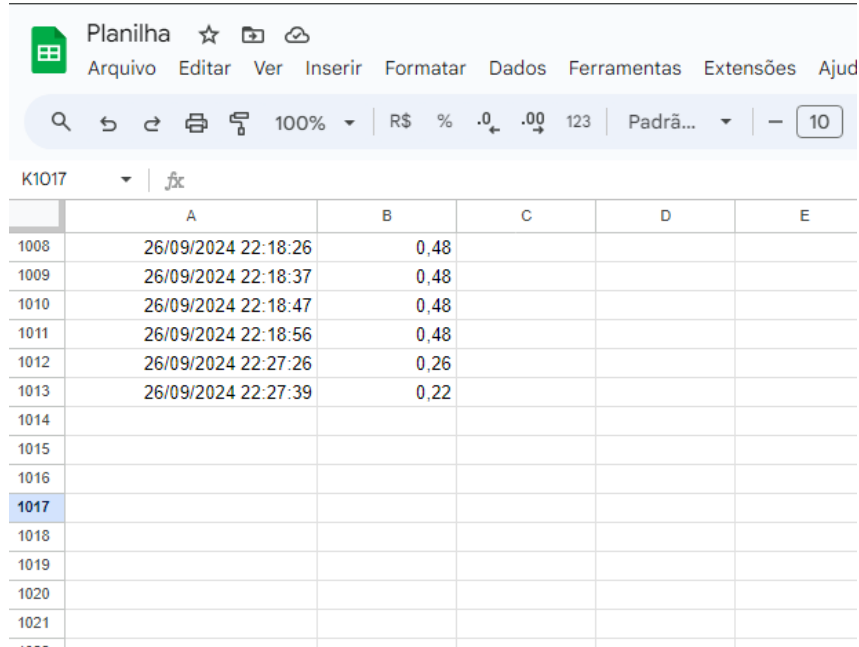
Os resultados indicaram que não houve perda de dados durante a simulação, considerando o fator de correção adotado de 1,077, os 200 pulsos enviados corresponderam a um volume de 0,48 litro. No entanto, foram registrados atrasos ocasionais na atualização dos dados na planilha Google Sheets. Esses atrasos, que duravam alguns segundos, não comprometeram a integridade das informações, já que os dados eram registrados corretamente após um curto intervalo.

Por meio da planilha Google Sheets percebe-se que os dados enviados pela ESP8266 também estão conforme os pulsos simulados.

Foi também testada a capacidade de detectar possíveis vazamentos, levando em conta o tempo reduzido de 5 minutos em uso contínuo, definido na simulação. O resultado foi positivo, como mostra a Figura 24.

Em síntese, os testes simulados foram concluídos com sucesso, validando o desenvolvimento do medidor de água inteligente, mostrado na Figura 25.

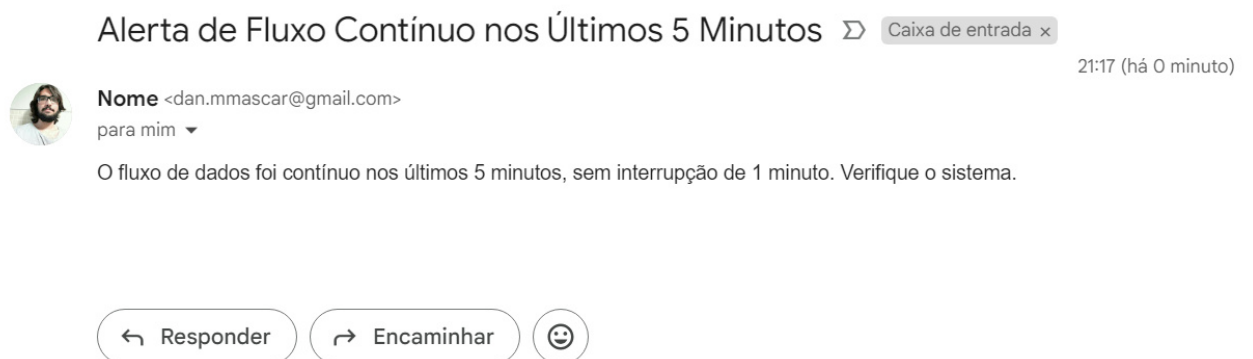
Figura 23 – Testes feitos com simulação de sinais



	A	B	C	D	E
1008	26/09/2024 22:18:26	0,48			
1009	26/09/2024 22:18:37	0,48			
1010	26/09/2024 22:18:47	0,48			
1011	26/09/2024 22:18:56	0,48			
1012	26/09/2024 22:27:26	0,26			
1013	26/09/2024 22:27:39	0,22			
1014					
1015					
1016					
1017					
1018					
1019					
1020					
1021					

Fonte: Próprio autor

Figura 24 – Email de alerta de fluxo constante durante simulação



Fonte: Próprio autor

Figura 25 – Medido de água instalado



Fonte: Próprio autor

## 5 CONSIDERAÇÕES FINAIS

O medidor de água desenvolvido neste trabalho atendeu aos objetivos propostos ao registrar adequadamente a vazão e o consumo de água, enviar e-mail com alertas de fluxo contínuo de água, além de disponibilizar informações remotamente em tempo real, por meio de gráficos de fácil compreensão.

Além disso, destaca-se por sua versatilidade e facilidade de instalação, podendo ser instalado em qualquer ponto de água na residência, no jardim ou na entrada da caixa d'água, sem a necessidade de quebrar paredes ou realizar grandes reformas. Também, apresenta uma interface gráfica que fornece aos usuários uma visão clara e detalhada do consumo de água. É possível visualizar o consumo em tempo real, dados históricos do dia, de uma semana e do mês, além de previsão de consumo. Com base nessas informações, o usuário pode adotar medidas para otimizar o uso da água, como a detecção de vazamentos e a implementação de hábitos mais conscientes.

É importante destacar que o objetivo do medidor de água desenvolvido neste trabalho não é substituir o hidrômetro instalado pela concessionária, mas sim fornecer ao usuário uma ferramenta adicional para monitorar o consumo de água de forma mais eficiente. Diferentemente do hidrômetro da concessionária, que não permite o acompanhamento remoto, o medidor proposto neste trabalho oferece essa funcionalidade, além do acompanhamento em tempo real do consumo, por meio de uma interface intuitiva. Com isso, elimina a necessidade do usuário se deslocar até o equipamento para verificar o consumo ou de manter registros manuais, proporcionando maior comodidade e precisão no controle do consumo de água.

Apesar de satisfatório, o medidor de água apresentou algumas limitações: de ordem energética, de transmissão de dados e de interação com os usuários. Em termos de energia, foi inicialmente utilizada uma bateria de 9 volts, que não foi suficiente para suportar o uso do dispositivo, sendo necessário utilizar uma fonte de energia elétrica durante todo o processo. No que diz respeito à transmissão de dados, foi observado que as oscilações da internet podem causar instabilidade no medidor de água, interrompendo a transmissão de dados de consumo. Em alguns casos, o dispositivo pode se recuperar sozinho após a estabilização da conexão, mas em outros, é necessário reiniciá-lo. Com relação às interações com o usuário, o medidor de água utiliza o Power BI Desktop para disponibilizar os gráficos, que pode ser utilizado gratuitamente no computador. Caso o usuário deseje disponibilizar um *link* do dashboard na internet, é preciso adquirir o Power BI Service que é uma versão paga.

Em suma, o medidor de água inteligente desenvolvido neste projeto demonstra potencial para auxiliar os usuários no controle eficiente do consumo de água residencial. Ao fornecer dados remotamente e em tempo real, o dispositivo permite a identificação de vazamentos, a otimização do uso da água e a adoção de hábitos mais conscientes, contribuindo para a economia doméstica.



## REFERÊNCIAS

- AL-FUQAHA, A.; GUIZANI, M.; MOHAMMADI, M.; ALEDHARI, M.; AYYASH, M. Internet of things: A survey on enabling technologies, protocols, and applications. **IEEE communications surveys & tutorials**, IEEE, v. 17, n. 4, p. 2347–2376, 2015.
- ARDUINO. **Introduction**. 2018. <https://www.arduino.cc/en/Guide/Introduction>. Acesso em: 17 maio 2023.
- BRANDÃO, I. A. d. P. *et al.* Sistema inteligente para automação do consumo de água do campus da ufcg de campina grande. Universidade Federal de Campina Grande, 2019.
- CAGECE. **Produtos e serviços**. 2024. <https://www.cagece.com.br/produtos-e-servicos/agua/>. Acesso em: 17 setembro 2024.
- COSTALONGA, T. O. Técnica hardware-in-the-loop no desenvolvimento de sistemas de controle e automação. **Revista Marítima Brasileira**, v. 140, n. 07/09, p. 190–205, 2020.
- CÔRTE, P. A. B. da. **Sistema de gerenciamento autônomo de consumo de água para casas inteligentes utilizando IoT**. 93 p. Monografia (Graduação) – Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Graduação em Ciência da Computação, Campina Grande - PB, 2020.
- FOWLER, M. **UML Essencial**. E-book. Porto Alegre: Grupo A, 2011. ISBN 9788560031382.
- IDEALI, W. **Conectividade em Automação e IoT: Protocolos I2C, SPI, USB, TCP-IP entre outros. Funcionalidade e interligação para automação e ToT**. [S. l.]: Alta Books, 2021.
- INTEL-CORPORATION. **Legacy Intel® Wireless Products**. 2024. <https://www.intel.com/content/www/us/en/support/articles/000005725/wireless/legacy-intel-wireless-products.html>. Acesso em: 21 setembro 2024.
- JÚNIOR, S. L. S.; FARINELLI, F. A. **Domótica - Automação Residencial e Casas Inteligentes com Arduino e ESP826**. Rio de Janeiro: Érica, 2018. E-book. ISBN 9788536530055.
- LOPES, A. M. R. **Aplicação da Simulação Hardware in the Loop para testes e desenvolvimento de suspensões veiculares**. Dissertação (Dissertação (mestrado)) – Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica, 2017. 1 recurso online (123 p.). Disponível em: <https://hdl.handle.net/20.500.12733/1631648>.
- MAGRANI, E. A internet das coisas no brasil. **Rio de Janeiro: FGV Editora**, 2018.
- MAIA, L. P. **Arquitetura de Redes de Computadores**. [S. l.]: Grupo GEN, 2013. ISBN 978-85-216-2436-3.
- MASCHIETTO, L.; VIEIRA, A. L. N.; TORRES, F. E.; OUTROS. **Arquitetura e Infraestrutura de IoT**. **Sagah Educação: Grupo A**, 2021.
- MATTOS, D. P. de; CONCEIÇÃO, N. L. da. Desenvolvimento de um hidrômetro inteligente para um consumo consciente. **Revista SIMEP**, v. 2, n. 1, p. 209–223, 2022.
- MEDEIROS, L. E. L. de; BARBOSA, D. L.; RODRIGUES, A. C. L.; BRANDÃO, I. A. de P.; SOARES, A. L. F. Desenvolvimento de um protótipo inteligente utilizando sensor turbina para detecção de vazamentos em edificações. **Revista Ibero-Americana de Ciências Ambientais**, v. 12, n. 11, p. 126–134, 2021.

- MONDAL, R.; ZULFI, T. Internet of things and wireless sensor network for smart cities. **International Journal of Computer Science Issues (IJCSI)**, International Journal of Computer Science Issues (IJCSI), v. 14, n. 5, p. 50–55, 2017.
- MONK, S. **30 Projetos com Arduino (Tekne)**. Porto Alegre: Bookman, 2014. E-book. ISBN 9788582601631.
- MORAES, C. d.; ALMEIDA, R. d.; SERAPHIN, T. **Programação de sistemas embarcados: desenvolvendo software para microcontroladores em linguagem C**. [S. l.]: Sao Paulo: Editora Elsevier Ltda, 2016.
- MORALES, F. **Conheça alguns dos prédios mais 'inteligentes' do Brasil**. 2012. <https://arquivo.canaltech.com.br/curiosidades/Predios-inteligentes-sao-a-grande-tendencia-para-o-setor-da-construcao-civil/>. Acesso em: 31 maio 2023.
- NATURE. **Água doce para a vida na Terra**. 2023. <https://www.nature.org/en-us/what-we-do/our-priorities/protect-water-and-land/land-and-water-stories/fresh-water-conservation/>. Acesso em: 30 junho 2023.
- NETO, A.; OLIVEIRA, Y. de. **Eletrônica Analógica e Digital Aplicada À IOT: Aprenda de maneira descomplicada**. [S. l.]: Alta Books, 2017.
- OLIVEIRA, A. S. de; ANDRADE, F. S. de. **Sistemas embarcados: hardware e firmware na prática**. [S. l.]: Editora Érica Ltda, 2010.
- OLIVEIRA, C. L. V.; ZANETTI, H. A. P. **Arduino Descomplicado-Como Elaborar Projetos de Eletrônica**. [S. l.]: Saraiva Educação SA, 2015.
- ORACLE. **O que é Internet of Things (IoT)**. 2023. <https://www.oracle.com/br/internet-of-things/>. Acesso em: 31 maio 2023.
- Power-BI. **Describing the Forecasting Models in Power View**. 2023. <https://powerbi.microsoft.com/en-us/blog/describing-the-forecasting-models-in-power-view/#which>. Acesso em: 27 setembro 2024.
- REINEHR, S. **Engenharia de requisitos**. [S. l.]: Grupo A, 2020. Disponível em: Minha Biblioteca.
- RUBIN, K. S. **Scrum Essencial: Um Guia Prático para o Processo Mais Popular Ágil**. [S. l.]: Editora Alta Livros, 2017. E-book. ISBN 9788550804118.
- SANTOS, B. P.; SILVA, L. A.; CELES, C.; BORGES, J. B.; NETO, B. S. P.; VIEIRA, M. A. M.; VIEIRA, L. F. M.; GOUSSEVSKAIA, O. N.; LOUREIRO, A. Internet das coisas: da teoria à prática. **Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**, v. 31, p. 16, 2016.
- SANTOS, M. da Silva dos; PADILHA, J.; CHAGAS, A. B. **Desenvolvimento Orientado a Reúso de Software**. Porto Alegre: Grupo A, 2021. E-book. ISBN 9786556902227.
- SCHACH, S. R. **Engenharia de Software-: Os Paradigmas Clássico e Orientado a Objetos**. [S. l.]: AMGH Editora, 2009.
- SILVA, F. R. e. a. **Redes sem fio**. [S. l.]: Grupo A, 2021. ISBN 9786556901374.

SILVA, R. A.; JUNIOR, S. L. S. **Automação e Instrumentação Industrial com Arduino-Teoria e Projetos**. [S. l.]: Saraiva Educação SA, 2015.

SOUZA, F. **O que são sistemas embarcados?** 2023. <https://embarcados.com.br/o-que-sao-sistemas-embarcados/> Acesso em: 30 julho 2023.

TRATA-BRASIL. **TRATA BRASIL**. 2023. <https://tratabrasil.org.br/principais-estatisticas/agua/> Acesso em: 30 junho 2023.

ZHAO, F.; GUIBAS, L. J. **Wireless sensor networks: an information processing approach**. [S. l.]: Morgan Kaufmann, 2004.

## 6 CÓDIGOS-FONTES UTILIZADOS

### 6.1 Código Arduino

```
1 // Defina o pino de entrada do sensor de fluxo
2 const int sensorDeFluxo = 2; // 0 pino digital 2
3 volatile int ContadorPulso = 0; // Contador de pulsos
4 float totalLitros = 0.0; // Total de litros contados
5 unsigned long temporizador = 0; // Tempo da última leitura
6 const unsigned long intervalo = 10000; // Intervalo de tempo
   para cálculos (em milissegundos, 10 segundos)
7 const float pulsosLitro = 450; // Pulsos por litro para o sensor
   YF-S201
8 float alpha = 1.077;
9
10 void setup() {
11   // Inicializa o pino do sensor como entrada
12   pinMode(sensorDeFluxo, INPUT);
13   // Configura a interrupção para contar pulsos
14   attachInterrupt(digitalPinToInterrupt(sensorDeFluxo),
   contarPulsos, FALLING);
15   // Inicializa a comunicação serial a 115200 baud rate
16   Serial.begin(115200);
17 }
18
19 void loop() {
20   unsigned long currentTime = millis();
21
22   // Se passaram 10 segundos
23   if (currentTime - temporizador >= intervalo) {
24     if (ContadorPulso > 0) { // Somente envia dados se houver
   fluxo
25       // Calcula o total de litros nos últimos 10 segundos
26       totalLitros = (ContadorPulso / pulsosLitro) * alpha;
27
```

```

28     // Converte o valor para string e substitui o ponto por ví
        rgula
29     String totalLitrosStr = String(totalLitros);
30     totalLitrosStr.replace('.', ',');
31
32     // Envia apenas os números para o monitor serial
33     Serial.println(totalLitrosStr);
34 }
35
36 // Reseta o contador de pulsos e o tempo da última leitura
37 ContadorPulso = 0;
38 temporizador = currentTime;
39 }
40 }
41
42 // Função chamada pela interrupção para contar pulsos
43 void contarPulsos() {
44     ContadorPulso++;
45 }

```

Código-fonte 1 – Código para monitoramento do fluxo de água

## 6.2 Código ESP8266

```

1
2 #include <ESP8266WiFi.h>
3 #include <ESP8266HTTPClient.h>
4 #include <WiFiClientSecure.h>
5 #include <ESP_Mail_Client.h> // Biblioteca para envio de e-mail
6
7 const char* ssid = "Wi-fi";
8 const char* password = "password";
9

```

```
10 const char* serverName = "https://script.google.com/macros/s/  
    insertSheetCode/exec";  
11  
12 // Configuração do SMTP  
13 const char* emailSender = "email@gmail.com";  
14 const char* emailSenderPassword = "emailPassword";  
15 const char* emailRecipient = "email.mmascar@gmail.com"; // Mesma  
    conta para enviar e receber  
16 const char* smtpServer = "smtp.gmail.com";  
17 const int smtpPort = 465;  
18  
19 SMTPSession smtp;  
20 ESP_Mail_Session mailSession;  
21 SMTP_Message message;  
22  
23 unsigned long ultimaDeteccaoFluxo = 0;  
24 unsigned long ultimoEmailEnviado = 0;  
25 const unsigned long umMinutoEmMilissegundos = 60000; // 1 minuto  
    em milissegundos  
26 const unsigned long umDiaEmMilissegundos = 86400000; // 24 horas  
    em milissegundos  
27 const unsigned long horaEmMilissegundos = 3600000; // 1 hora em  
    milissegundos  
28 bool fluxoParado = false;  
29 bool emailSent = false; // Flag para verificar se o e-mail foi  
    enviado  
30  
31 void setup() {  
32     Serial.begin(115200);  
33     WiFi.begin(ssid, password);  
34  
35     while (WiFi.status() != WL_CONNECTED) {  
36         delay(1000);  
37         Serial.println("Conectando ao WiFi...");
```

```
38     }
39     Serial.println("WiFi conectado!");
40
41     // Configurar a sessão SMTP
42     mailSession.server.host_name = smtpServer;
43     mailSession.server.port = smtpPort;
44     mailSession.login.email = emailSender;
45     mailSession.login.password = emailSenderPassword;
46     mailSession.login.user_domain = "";
47
48     smtp.debug(1);
49 }
50
51 void loop() {
52     unsigned long currentTime = millis();
53
54     if (WiFi.status() == WL_CONNECTED) {
55         if (Serial.available() > 0) {
56             String valor = Serial.readStringUntil('\n'); // Lê o
                    valor da serial
57             valor.trim(); // Remove espaços em branco extras
58
59             WiFiClientSecure client;
60             client.setInsecure(); // Ignora a verificação do
                    certificado SSL
61
62             HTTPClient http;
63
64             String url = serverName;
65             url += "?valor=" + valor; // Adiciona o valor à URL
66
67             http.begin(client, url); // Inicia a conexão HTTPS
68
```

```
69     int httpResponseCode = http.GET(); // Envia a requisição
70         GET
71
72     if (httpResponseCode > 0) {
73         Serial.println("Dado enviado com sucesso!"); //
74             Confirma que o dado foi enviado
75     } else {
76         Serial.println("Erro ao enviar o dado."); // Indica que
77             houve um erro no envio
78     }
79
80     http.end(); // Encerra a conexão
81
82     // Atualiza o tempo do último fluxo detectado
83     ultimaDeteccaoFluxo = currentTime;
84     fluxoParado = false; // Indica que o fluxo está ativo
85 }
86
87 } else {
88     Serial.println("WiFi desconectado");
89 }
90
91 // Verifica se o fluxo parou por 1 minuto
92 if ((currentTime - ultimaDeteccaoFluxo >=
93     umMinutoEmMilisegundos) && !fluxoParado) {
94     fluxoParado = true; // Marca que o fluxo foi interrompido
95         por pelo menos 1 minuto
96 }
97
98 // Verifica se se passaram 24 horas desde o último e-mail
99 if (currentTime - ultimoEmailEnviado >= umDiaEmMilisegundos) {
100     // Se o fluxo foi contínuo nas últimas 24 horas (ou seja, não
101         houve interrupção de 1 minuto)
102     if (!fluxoParado) {
103         sendEmail();

```



```
97     ultimoEmailEnviado = currentTime;
98     emailSent = true;
99
100    // Atualiza o tempo do último fluxo detectado para
        garantir que a condição seja verificada novamente nas
        próximas 24 horas
101    ultimaDeteccaoFluxo = currentTime;
102 }
103 }
104 // Se um e-mail foi enviado e 1 hora se passou desde o último
        e-mail
105 else if (emailSent && currentTime - ultimoEmailEnviado >=
        horaEmMilisegundos) {
106     if (!fluxoParado) {
107         sendEmail();
108         ultimoEmailEnviado = currentTime;
109     }
110 }
111 }
112
113 void sendEmail() {
114     // Configurar a mensagem de e-mail
115     message.sender.name = "Sensor de Fluxo";
116     message.sender.email = emailSender;
117     message.subject = "Fluxo Anormal Detectado";
118     message.addRecipient("Destinatário", emailRecipient);
119     message.text.content = "Nas últimas 24 horas, foi detectado um
        fluxo contínuo de água sem interrupção de 1 minuto. Existe
        um alerta de possível vazamento. Verifique o sistema.";
120
121     // Conectar ao servidor SMTP e enviar o e-mail
122     if (!smtp.connect(&mailSession)) {
123         Serial.println("Erro ao conectar ao servidor SMTP.");
124         return;
```

```

125 }
126
127 if (!MailClient.sendMail(&smtp, &message)) {
128     Serial.println("Erro ao enviar o e-mail.");
129     return;
130 } else {
131     Serial.println("E-mail enviado com sucesso.");
132 }
133
134 smtp.closeSession();
135 }

```

Código-fonte 2 – Código para monitoramento de fluxo de água com envio de dados e alertas por e-mail

### 6.3 Script para Google Sheets

```

1 function doGet(e) {
2     var valor = e.parameter.valor;
3     var ss = SpreadsheetApp.openById('codigoDaTabela');
4     var sheet = ss.getSheetByName('Sheet1'); // substitua pelo
           nome correto da aba
5
6     // Formatar a data e hora
7     var now = new Date();
8     var formattedDate = Utilities.formatDate(now, Session.
           getScriptTimeZone(), "dd/MM/yyyy HH:mm:ss");
9
10    sheet.appendRow([formattedDate, valor]);
11    return ContentService.createTextOutput('Dados recebidos com
           sucesso');
12 }

```

Código-fonte 3 – Código do Google Apps Script para salvar dados em Google Sheets

Códigos podem ser acessados [aqui](#).