

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DO RIO GRANDE DO SUL
CAMPUS RESTINGA**

**SISTEMA DE MONITORAMENTO DE EXTRAVASÃO E
NÍVEL DE RESERVATÓRIOS DE ÁGUA PREDIAIS:
SOLUÇÃO IOT PARA MINIMIZAR PREJUÍZOS POR
EXTRAVASÃO**

MAURÍCIO SOUZA SIMÕES

**Porto Alegre
2025**

MAURÍCIO SOUZA SIMÕES

**SISTEMA DE MONITORAMENTO DE EXTRAVASÃO E
NÍVEL DE RESERVATÓRIOS DE ÁGUA PREDIAIS:
SOLUÇÃO IOT PARA MINIMIZAR PREJUÍZOS POR
EXTRAVASÃO**

Trabalho de Conclusão de Curso apresentado, junto ao Curso de Análise e Desenvolvimento de Sistemas do Instituto Federal Educação, Ciência e Tecnologia do Rio Grande do Sul, como requisito parcial para a obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Roben Castagna Lunardi

**Porto Alegre
2025**

MAURÍCIO SOUZA SIMÕES

**SISTEMA DE MONITORAMENTO DE EXTRAVASÃO E
NÍVEL DE RESERVATÓRIOS DE ÁGUA PREDIAIS:
SOLUÇÃO IOT PARA MINIMIZAR PREJUÍZOS POR
EXTRAVASÃO**

Trabalho de Conclusão de Curso apresentado
como requisito parcial para a obtenção do grau
de Tecnólogo em Análise e Desenvolvimento
de Sistemas.

Orientador: Prof. Roben Castagna Lunardi

Aprovado em __, ____.

Roben Castagna Lunardi - IFRS – Campus Restinga

Membro da Banca - Professor André Marcelo Schneider – Instituto Federal de Educação

Membro da Banca – Professor Rafael Pereira Esteves – IFRS – Campus Restinga

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO GRANDE DO SUL

Reitor: Prof. Júlio Xandro Heck

Pró-Reitora de Ensino: Profa. Clarice Monteiro Escott

Diretor-geral do *Campus* Restinga: Prof. Rudinei Müller

Coordenador do CST em Análise e Desenvolvimento de Sistemas: Rafael Pereira Esteves

Bibliotecária-chefe do *Campus* Restinga: Paula Porto Pedone

Dedico este trabalho ao meu filho, à minha namorada, aos meus irmãos, à minha querida mãe e ao meu falecido pai e às minhas netas, que entenderam minha ausência e que me apoiaram e compreenderam a importância desta conquista.

AGRADECIMENTOS

Agradeço aos meus amigos, familiares e minha namorada pelo apoio incondicional durante a jornada deste TCC. Em especial, meu agradecimento mais profundo vai para o meu filho, cujo suporte técnico foi essencial para a conclusão deste trabalho. Sua ajuda prática, conhecimento e paciência foram fundamentais nos momentos mais desafiadores. Obrigado por estar ao meu lado e por acreditar em mim.

“Acredite que você pode, assim você já está no meio do caminho.”
(Theodore Roosevelt)

RESUMO

Diante da fartura de recursos hídricos em nossa região, em relações a outras regiões do nosso país, ou do mundo, não podemos deixar de minimizar, sempre que possível, o desperdício. Este projeto propõe-se a criar um sistema, baseado em Internet das Coisas (*IoT*), para monitoramento, em tempo integral, de nível de um reservatório de água, com especial atenção à extravasão, visando reduzir significativamente o desperdício de água potável e minimizar possíveis prejuízos financeiros por extravasão. Para tanto o sistema contará, além das informações em tempo real, com alarmes para direcionar a atenção do usuário a estes eventos. O sistema funcionará, coletando os dados com o uso de um módulo de desenvolvimento ESP32, que estará apto a comunicar com um ou mais módulos ESP8266, que atuarão como interface homem-máquina (*HMI*), permitindo a visualização do status do reservatório, resgatando os dados da nuvem, e também um aplicativo, permitindo o monitoramento em tempo real, independentemente da localização.

Palavra-chave: Monitoramento de nível de água; Monitoramento de desperdício de água; Monitoramento em tempo real; Plataforma IoT ThingSpeak; Plataforma NodeMCU; IoT (Internet das coisas).

ABSTRACT

Given the abundance of water resources in our region, compared to other regions of our country or the world, we must always minimize waste whenever possible. This project proposes to create a system, based on the Internet of Things (IoT), for continuous monitoring of the water reservoir level, with special attention to overflow, aiming to significantly reduce potable water waste and minimize potential financial losses due to overflow. To this end, the system will include, in addition to real-time information, alarms to direct the user's attention to these events. The system will operate by collecting data using an ESP32 development module, which will be able to communicate with one or more ESP8266 modules, acting as a human-machine interface (HMI), allowing the visualization of the reservoir status, retrieving data from the cloud, and also an application, allowing real-time monitoring regardless of location.

Keywords: *Water level monitoring; Water waste monitoring; Real-time monitoring; IoT Platform ThingSpeak; NodeMCU Platform; IoT (Internet of Things).*

LISTA DE FIGURAS

FIGURA 1. ESQUEMA DE FUNCIONAMENTO DE CAIXA D'ÁGUA	14
FIGURA 2. DIAGRAMA LC1BD04.....	16
FIGURA 3. RESULTADO QUESTÃO 1.....	25
FIGURA 4. RESULTADO QUESTÃO 2.....	25
FIGURA 5. RESULTADO QUESTÃO 3.....	26
FIGURA 6. RESULTADO QUESTÃO 4.....	26
FIGURA 7. RESULTADO QUESTÃO 5.....	27
FIGURA 8. RESULTADO QUESTÃO 6.....	27
FIGURA 9. MODELAGEM CONCEITUAL.....	28
FIGURA 10. FLUXOGRAMA DO COLETOR DE DADOS	29
FIGURA 11. FLUXOGRAMAS DO DISPLAY.....	29
FIGURA 12. FLUXOGRAMAS DO APLICATIVO DISPLAY	30
FIGURA 13. POSICIONAMENTO DO CONJUNTO DE SENSORES	30
FIGURA 14. DIAGRAMA ESQUEMÁTICO COLETOR DE DADOS.....	32
FIGURA 15. PROTÓTIPO COLETOR DE DADOS.....	32
FIGURA 16. PROTÓTIPO DISPLAY	33
FIGURA 17. DIAGRAMA ESQUEMÁTICO DISPLAY.....	33
FIGURA 18. WORKFLOW COLETOR DE DADOS.....	35
FIGURA 19. WORKFLOW DISPLAY	36
FIGURA 20. WORKFLOW APLICATIVO DISPLAY	37
FIGURA 21. JSON RESGATADO PELO MONITOR.....	38
FIGURA 22. INSTALAÇÃO COLETOR DE DADOS	39
FIGURA 23. INSTALAÇÃO DOS SENSORES	40
FIGURA 24. LEITURA DO DISPLAY	41
FIGURA 25. TELA APLICATIVO DISPLAY	42
FIGURA 26. MONITOR_LEVEL_SENSOR.INO	47
FIGURA 27. FLAG.INO	48
FIGURA 28. MONITOR_LEVEL_TERMINAL(1 DE 2).....	49
FIGURA 29. MONITOR_LEVEL_TERMINAL(2 DE 2).....	50
FIGURA 30. APP LEVELMONITOR.....	51

LISTA DE TABELAS

TABELA 1. TRABALHOS CORRELATOS.....	21
--	-----------

SUMÁRIO

AGRADECIMENTOS.....	11
1. INTRODUÇÃO.....	11
2. FUNDAMENTAÇÃO TEÓRICA.....	13
2.1. DESAFIO DOS SISTEMAS DE ABASTECIMENTO DE ÁGUA.....	13
2.2. EXTRAVASOR/LADRÃO.....	13
2.3. INTERNET DAS COISAS.....	14
2.4. THINGSPEAK.....	14
2.5. REDES DE COMUNICAÇÃO SEM FIO (WIFI).....	15
2.6. MICROCONTROLADOR ESP8266(NODEMCU 0.9 (ESP-12 MODULE)).....	15
2.7. MICROCONTROLADOR ESP32(ESP32 DEV MODULE).....	15
2.8. LC1BD04 – SENSOR DE NÍVEL DE ÁGUA DE 4 DÍGITOS.....	15
2.9. YF-S201–SENSOR DE FLUXO DE ÁGUA HALL.....	16
2.10. BUZZER.....	16
2.11. PROTOCOLO HTTP.....	17
2.12. JSON.....	17
3. TRABALHOS RELACIONADOS.....	18
3.1. MONITORAMENTO DO NÍVEL DE ÁGUA EM RESERVATÓRIOS RESIDENCIAIS UTILIZANDO SENSOR ULTRASSÔNICO DE ALVES (2023).....	19
3.2. SISTEMA DE MONITORAMENTO REMOTO DO NÍVEL DE CAIXAS D’ÁGUA EMPREGANDO A TECNOLOGIA DE BROZOSKI E SILVA (2022).....	20
3.3. DESENVOLVIMENTO DE UMA ESTAÇÃO METEOROLÓGICA AUTOMÁTICA EMBARCADA COM O USO DA PLATAFORMA THINGSPEAK NO MUNICÍPIO DE BELÉM – PA DE ARAÚJO (2022).....	20
3.4. SÍNTESE COMPARATIVA.....	21
4. LEVANTAMENTO DE REQUISITOS.....	24
4.1. ANÁLISE E DISCUSSÃO DA PESQUISA.....	24
4.1.1. <i>Questões da Pesquisa</i>	24
4.1.2. <i>Resultados da Pesquisa</i>	24
5. SOLUÇÃO CONCEITUAL.....	28
6. IMPLEMENTAÇÃO.....	31
6.1. IMPLEMENTAÇÃO DOS MÓDULOS DE COLETA DE DADOS E MONITORAMENTO.....	31
6.1.1. <i>Coletor de dados</i>	31
6.1.2. <i>Display</i>	33
6.2. IMPLEMENTAÇÃO DOS CÓDIGOS DE COLETA DE DADOS E MONITORAMENTO.....	34
6.2.1. <i>Coletor de dados</i>	34
6.2.2. <i>Display</i>	36
6.2.3. <i>Aplicativo Display</i>	37
6.3. JSON EXTRAÍDO DO THINGSPEAK.....	38
6.4. INSTALAÇÃO DO <i>HARDWARE</i> DE COLETA DE DADOS E MONITORAMENTO.....	39
6.5. INSTALAÇÃO E USO DO <i>HARDWARE</i> DO DISPLAY.....	41
6.6. INSTALAÇÃO E USO APLICATIVO DISPLAY.....	42
6.7. TESTE DO SISTEMA.....	42
7. CONCLUSÃO.....	43
REFERÊNCIAS BIBLIOGRÁFICAS.....	44

APÊNDICE A - MONITOR_LEVEL_SENSOR.INO.....	47
APÊNDICE B - FLAG.INO.....	48
APÊNDICE C - MONITOR_LEVEL_TERMINAL.INO.....	49
APÊNDICE D - APP LEVEL MONITOR.....	51

1. INTRODUÇÃO

Os reservatórios de água em edificações possuem um extravasor, comumente chamado de “ladrão”, que permite a saída de água em caso de falha no mecanismo de controle de nível automático. Observando-se a extravasão, o responsável pode tomar as devidas providências para sanar o problema.

Dependendo das características da edificação, o extravasor pode não ser visível, ter algum problema e, mesmo estando em conformidade, as condições climáticas podem misturar a água de extravasão com a água das chuvas, que por vezes podem ser prolongadas, dificultando a percepção do evento, gerando prejuízos como:

Infiltração na estrutura, podendo enfraquecer vigas e chapas por oxidação da ferragem.

Inundação, podendo escorrer pelo alçapão de acesso ao sistema hidráulico, tubulação elétrica ou por infiltração, atingindo os apartamentos abaixo, gerando danos ao mobiliário, pisos em madeira, forro de gesso e equipamentos eletroeletrônicos.

Desperdício de água potável no esgoto pluvial. Geralmente essa situação somente é percebida no momento em que chega a fatura do consumo de água.

Diante do exposto, há a necessidade de informar o responsável sobre o problema o mais rápido possível para que execute o procedimento cabível, evitando as implicações do possível transbordo não sanado.

Este trabalho propõe-se a desenvolver um sistema com o objetivo de monitorar os níveis de água e extravasor de reservatório de água tornando-o notório, o mais rápido possível, permitindo ação rápida e eficaz.

Para atingir este objetivo será feito o uso de tecnologia de Internet das Coisas (IoT) que tem se mostrado extremamente eficaz em aplicações de monitoramento remoto. Sistema será composto por um módulo ESP32 para a coleta contínua. Este dispositivo será responsável pela coleta contínua de dados e envio à nuvem, onde os mesmos ficarão armazenados e disponíveis.

Como complemento do sistema serão utilizados um ou mais módulos ESP8266, que atuarão como display, permitindo a visualização do status do reservatório, resgatando os dados da nuvem, e também um APP, permitindo o monitoramento em tempo real, independentemente da localização.

Este Trabalho de Conclusão de Curso está organizado da seguinte forma: o capítulo 2

apresenta a fundamentação teórica, abordando temas como desperdício de água, extravasores, Internet das Coisas (IoT), e as plataformas ThingSpeak e NodeMCU. O capítulo 3 detalha os trabalhos relacionados, incluindo diversas pesquisas sobre monitoramento de nível de água. O capítulo 4 descreve a solução conceitual proposta, enquanto o capítulo 5 detalha a implementação dos módulos e códigos de coleta de dados e monitoramento. No capítulo 6, são apresentados os resultados da avaliação e análise dos testes realizados. Finalmente, o capítulo 7 traz as conclusões deste trabalho, além das referências bibliográficas e apêndices relevantes.

2. FUNDAMENTAÇÃO TEÓRICA

Os próximos tópicos abordarão alguns temas necessários para a compreensão do trabalho desenvolvido.

2.1. Desafio dos sistemas de abastecimento de água

Conforme a LEI Nº 9.433, DE 8 DE JANEIRO DE 1997, A água é um recurso natural limitado, dotado de valor econômico[...] a gestão dos recursos hídricos deve ser descentralizada e contar com a participação do Poder Público, dos usuários e das comunidades (BRASIL, 1997). Sobre este tema o site Instituto Água Viva diz que, “o Brasil, mesmo com abundantes recursos hídricos, tem dificuldade para distribuir, a água, de maneira consciente e justa. Mesmo com 12% da água doce do mundo, temos regiões de seca e regiões de abundância”. (INSTITUTO ÁGUA VIVA, 2020)

De acordo com a Unesco (2020, p.2), não temos como mensurar a extensão do impacto futuro a nível global causado pelas mudanças climáticas e isso precisa impulsionar o uso de ciência e tecnologia para que possamos fazer algo.

2.2. Extravasor/Ladrão

Resumindo a norma de “Instalação predial e água fria” (NBR, 5626:1998, p. 5), em todos os reservatórios, deve ser instalada tubulação que avise aos usuários quando a torneira de boia ou dispositivo de interrupção do abastecimento do reservatório apresentar falha. Como consequência, ocorrerá a elevação da superfície da água acima do nível máximo previsto, resultando na extravasão do volume de água em excesso do interior do reservatório. Para impedir o transbordamento, a tubulação de aviso deve descarregar imediatamente após a água alcançar o nível de extravasão no reservatório. A água deve ser descarregada em um local facilmente observável e o diâmetro do extravasor precisa ser maior que o da tubulação de entrada, pois não há pressão as extravasão.

Na Figura 1, é ilustrada a instalação de um reservatório de água em conformidade com

as normas, que salienta visualmente a necessária diferença entre os diâmetros dos tubos.

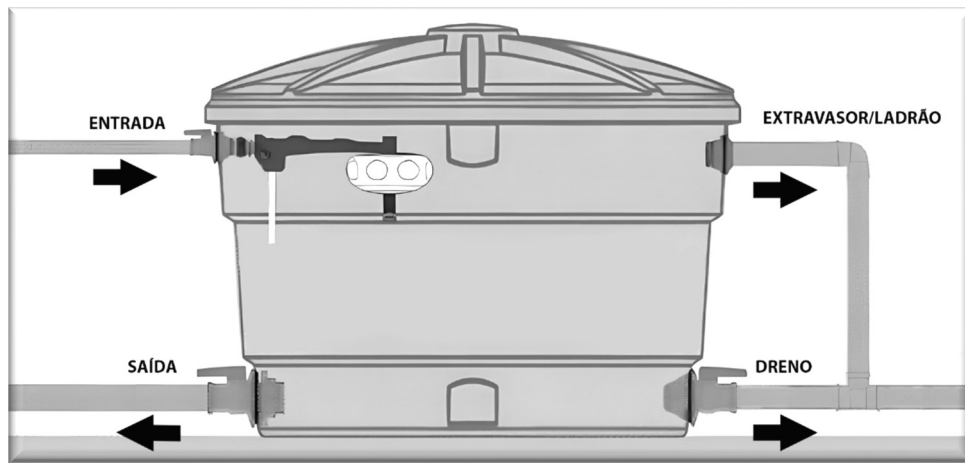


Figura 1. Esquema de funcionamento de Caixa D'água

Fonte: <https://construwiki.blogspot.com/2015/02/>

2.3. Internet das coisas

Uma maneira de expressar o conceito de Internet das Coisas pode ser visto conforme Santaella (2013, p.28), atualmente se entende que Internet das Coisas é a internet usada por dispositivos que efetuam comunicação sem a necessidade de interação humana e segundo Mancini (2020, p.1), ela proporciona aos objetos inteligentes a comunicação, processamento e o uso de sensores.

2.4. ThingSpeak

ThingSpeak é uma plataforma armazenamento e troca de informações em tempo real, voltada para IoT, que mantém algum histórico de informações que permite análise em tempo real (THINGSPEAK, 2025).

2.5. Redes de comunicação sem fio (WIFI)

Conforme Silva (2022, p. 13), o uso do WIFI é a conexão sem fio para a acesso à internet, através de ondas de rádio. Tecnologia usada para a comunicação entre os dispositivos e a estrutura física, substituindo os cabos que podem ser ligados ao modem.

2.6. Microcontrolador ESP8266 (NodeMCU 0.9 (ESP-12 Module))

De acordo com ESP8266 *Technical Reference*(Espressif Systems Inc., 2020), o ESP8266 é uma solução compacta com conexão WiFi de baixíssimo consumo com um processador de 32 bits, com SRAM *on-chip* e um *flash* externo, que podemos gravar o código. Ao ser ligado, executa diretamente o código no *flash* que pode manipular as portas GPIOs e WiFi.

2.7. Microcontrolador ESP32(ESP32 Dev Module)

Segundo Babiuch (2019, 1-6) uma geração superior e posterior ao ESP8266, o ESP32 possui todos os recursos e mais. Sendo considerado uma atualização.

2.8. LC1BD04 – Sensor de nível de água de 4 dígitos

O modulo LC1BD04 é amplamente utilizado para prototipagem, com 4 entradas de alta impedância que detecta a resistência da água entre cada uma das 4 entradas e o referencial de terra(0V), mudando o estado da respectiva saída, conforme diagrama da Figura 2. Mas seu circuito eletrônico é amplamente usado em variados produtos e possui robustez para ser colocado em produção.

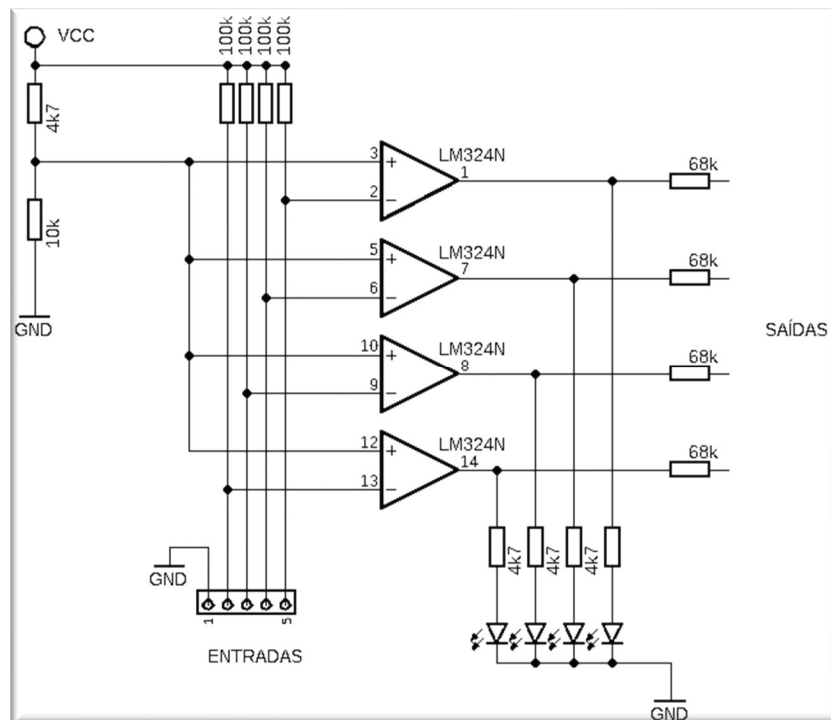


Figura 2. Diagrama LC1BD04

Fonte: Próprio autor.

2.9. YF-S201–Sensor de fluxo de água Hall

O sensor de fluxo de água (ELETROGATE, 2022), consiste em uma roda d'água acoplada a um ímã. Ao movimentar a água, o ímã varia o campo magnético sobre o sensor de efeito Hall próximo. Esta influência magnética altera sua condução elétrica, gerando um pulso. Seu rotor e ímã na porção úmida e componentes eletrônicos no exterior, completamente selados é robusto.

2.10. BUZZER

Buzzer ativo é um componente que gera sinal sonoro. Ao ser alimentado o mesmo emite sinal sonoro de frequência fixa.

2.11. Protocolo HTTP

HTTP é o protocolo base para troca de dados na WEB. Um protocolo cujas trocas de mensagens são iniciadas pelo cliente. Os métodos HTTP são verbos GET, POST, DELETE, PUT, etc ou substantivos OPTIONS ou HEAD que determinam a operação. (Mozilla, 2024)

2.12. JSON

JSON é uma forma de serializar objetos, *arrays*, números, *strings*, booleanos e *null*. Embora utilize a sintaxe do Javascript como base, há diferenças importantes: algumas estruturas do Javascript não são compatíveis com JSON, assim como alguns elementos do JSON não são válidos em Javascript. (Mozilla, 2024)

3. TRABALHOS RELACIONADOS

Realizada uma pesquisa bibliográfica para verificar a existência de algum sistema de monitoramento que já abarcassem os métodos, requisitos e funcionalidades do sistema proposto neste Trabalho de Conclusão de Curso que possuam conteúdo relevante para do desenvolvimento do sistema em questão.

Para tal, foram propostas questões:

- Existem sistemas de informação criados para monitoramento de nível e/ou extravasão em reservatórios?
- Existem sistemas de informação criados para monitoramento de nível e/ou extravasão em reservatórios com uso de IoT?
- Quais plataformas (web, desktop, celular, etc) são empregadas para sistemas de informação criados para o monitoramento de nível e extravasão em reservatórios?
- Qual o hardware utilizado para construção do sistema?
- Quais tecnologias (banco de dados, linguagem de programação, etc) são mais empregados nos sistemas de informação criados para monitoramento de nível e extravasão em reservatórios?
- Qual o público atendido com as soluções implementadas para monitoramento de nível e/ou extravasão em reservatórios?

A partir destas questões foi extraído o seguinte conjunto de palavras chaves para execução da pesquisa: Monitoramento; Nível de água; Sensores; IoT.

Para definir quais documentos serão lidos foram criados os seguintes critérios de seleção:

- Somente documentos em português.
- Documentos que tenham no resumo ao menos uma palavra chave.

Por fim, foram elencados os seguintes critérios para ranquear os documentos:

- Documentos que respondam as questões de pesquisa 1 ou 2;
- Documentos que respondam ao menos três questões de pesquisa;

Com esse planejamento de pesquisa foram encontrados 3 documentos que serão descritos a seguir.

3.1. Monitoramento do nível de água em reservatórios residenciais utilizando sensor ultrassônico de Alves (2023)

O trabalho "Monitoramento do nível de água em reservatórios residenciais utilizando sensor ultrassônico" aborda o desenvolvimento de um sistema eficiente para medir e monitorar os níveis de água em reservatórios domésticos. A proposta central é utilizar sensores ultrassônicos para aferir a altura da coluna de água, permitindo um controle preciso e em tempo real da quantidade de água disponível.

O sistema desenvolvido combina sensores ultrassônicos com um microcontrolador que processa os dados coletados e os transmite para uma interface de usuário, que pode ser acessada remotamente. A utilização de sensores ultrassônicos se dá pela medição de distâncias, tornando-os ideais para aplicações em ambientes variáveis como os reservatórios de água.

Os resultados apresentados no trabalho indicam que o sistema proposto é capaz de fornecer leituras precisas e contínuas dos níveis de água. O monitoramento automatizado elimina a necessidade de verificações manuais frequentes. Dessa forma, o estudo demonstra a viabilidade e a importância de integrar tecnologias avançadas em soluções práticas para desafios cotidianos.

3.2. Sistema de monitoramento remoto do nível de caixas d'água empregando a tecnologia de Brozowski e Silva (2022)

O trabalho "Sistema de monitoramento remoto do nível de caixas d'água empregando a tecnologia LoraWAN" apresenta o desenvolvimento de um sistema para o acompanhamento à distância dos níveis de água em caixas d'água. A tecnologia LoraWAN (Long Range Wide Area Network) é a base desse sistema, proporcionando comunicação sem fio de longo alcance e baixo consumo de energia. Esta tecnologia é ideal para a criação de redes de sensores em áreas amplas, garantindo que as informações sobre o nível de água sejam transmitidas de maneira eficiente e confiável.

A implementação do sistema envolve o uso de sensores para medir o nível de água, conectados a módulos LoraWAN que enviam os dados coletados para uma central de monitoramento. Esta central pode ser acessada remotamente por meio de dispositivos móveis ou computadores, permitindo aos usuários verificar os níveis de água em tempo real. A escolha da tecnologia LoraWAN se deve à sua capacidade de cobrir grandes distâncias e penetrar em ambientes com barreiras físicas, características comuns em contextos residenciais e industriais.

Os resultados do estudo mostram que o sistema de monitoramento remoto desenvolvido é eficaz em fornecer dados precisos e em tempo real sobre os níveis de água em caixas d'água. Isso facilita a gestão dos recursos hídricos, evitando desperdícios e garantindo que as caixas d'água estejam sempre dentro dos níveis ideais. Além disso, o uso da tecnologia LoraWAN se mostrou uma solução viável e econômica para a implementação de redes de sensores, podendo ser replicada em diversas outras aplicações de monitoramento remoto.

3.3. Desenvolvimento de uma estação meteorológica automática embarcada com o uso da plataforma Thingspeak no município de Belém – PA de Araújo (2022)

O trabalho "Desenvolvimento de uma estação meteorológica automática embarcada com o uso da plataforma Thingspeak no município de Belém - PA" detalha a criação e implementação de uma estação meteorológica automática para monitorar variáveis climáticas importantes. Utilizando a plataforma Thingspeak, o sistema desenvolvido permite a coleta, análise e visualização dos dados climáticos em tempo real. A estação foi projetada para fornecer informações precisas sobre o clima, essenciais para diversas aplicações, desde a agricultura até a pesquisa científica.

A estação meteorológica é equipada com uma série de sensores que medem parâmetros como temperatura, umidade, pressão atmosférica, velocidade e direção do vento, além de precipitação. Os dados coletados pelos sensores são processados por um microcontrolador e enviados para a plataforma Thingspeak, onde são armazenados e disponibilizados para acesso remoto. Essa abordagem permite a integração dos dados em sistemas de monitoramento mais amplos e facilita a análise por meio de gráficos e outras ferramentas visuais oferecidas pela plataforma.

Os resultados obtidos com a estação meteorológica automática demonstram a eficácia do sistema em fornecer dados climáticos confiáveis e em tempo real. A utilização da plataforma Thingspeak não só facilita a visualização e análise dos dados, mas também possibilita o compartilhamento das informações. A implementação desse sistema em Belém - PA evidencia a viabilidade de soluções tecnológicas acessíveis para o monitoramento ambiental.

3.4. Síntese Comparativa

Tabela 1. Trabalhos correlatos

	Monitoramento do nível de água em reservatórios residenciais utilizando sensor ultrassônico	Sistema de monitoramento remoto do nível de caixas d'água empregando a tecnologia LoraWAN	Desenvolvimento de uma estação meteorológica automática embarcada com o uso da plataforma Thingspeak no município de Belém - PA.
Hardware	Sensor Ultrassônico (HC-SR04), Microcontrolador (Arduino ou ESP8266/ESP32), Módulo de Comunicação Wi-Fi (ESP8266/ESP32)	Sensor de Nível (Ultrassônico ou de Pressão como MPX5010), Microcontrolador compatível com LoraWAN (Arduino MKR WAN 1300 ou ESP32 com módulo Lora), Módulo LoraWAN (RFM95W ou Dragino Lora Shield), Gateway LoraWAN (The Things Gateway)	Sensores Meteorológicos (DHT22, BMP180, anemômetro, vane, pluviômetro), Microcontrolador (Arduino ou ESP8266/ESP32), Módulo de Comunicação Wi-Fi (ESP8266/ESP32)

Plataformas	React, Angular ou Vue.js para a web	A plataforma envolve a utilização de uma rede LoraWAN .	A plataforma principal é o ThingSpeak .
Linguagens	C/C++ para programação do microcontrolador, e JavaScript, HTML e CSS para o desenvolvimento da interface web. Para dispositivos móveis, as linguagens podem incluir Dart (usado pelo Flutter) ou JavaScript/TypeScript (usado pelo React Native).	As linguagens utilizadas incluem C/C++ para programação dos módulos LoraWAN, e JavaScript, HTML e CSS para desenvolvimento da interface web.	As linguagens incluem C/C++ para a programação dos microcontroladores que coletam os dados dos sensores, e MATLAB ou JavaScript para processamento e visualização dos dados na plataforma Thingspeak.
Público atendido	O público-alvo são proprietários de residências que desejam monitorar e gerir o nível de água em seus reservatórios.	O público-alvo são tanto usuários residenciais quanto comerciais que necessitam monitorar os níveis de água.	O público-alvo são pesquisadores e acadêmicos interessados em monitorar e estudar o clima local, agricultores que necessitam de dados meteorológicos para planejamento agrícola, e autoridades públicas.

Fonte: próprio autor.

A tabela Tabela 1 resume os principais componentes de hardware e tecnologias utilizadas em três diferentes sistemas de monitoramento de recursos hídricos. Todos os projetos utilizam microcontroladores ESP32 ou ESP8266, conhecidos por suas capacidades de comunicação sem fio integradas. Os sistemas abordam o monitoramento do nível de água em reservatórios e caixas d'água, além de uma estação meteorológica automática. Os sensores variam de ultrassônicos a sensores de pressão e meteorológicos, enquanto a plataforma ThingSpeak é muito boa para a coleta, análise e visualização de dados. Esses sistemas são desenvolvidos utilizando linguagens de programação C/C++ e são voltados para o público que necessita de soluções eficientes e acessíveis para a gestão de recursos e monitoramento.

Além do monitoramento, o sistema proposto pretende gerar alertas em caso de extravasão e monitorar a integridade do próprio sistema, seja através de hardware dedicado ou

por meio de um aplicativo com suporte à tecnologia IoT, utilizando ESP32 e ESP8266. No entanto, sem usar os sensores por eles utilizados, o que implica em ajuste nas tabelas, de acordo com cada reservatório a ser instalado.

4. LEVANTAMENTO DE REQUISITOS

Realizada pesquisa direcionada aos responsáveis por edificações com caixas d'água, síndicos, proprietários de imóveis e zeladores, para evidenciar a relevância do sistema proposto e suas possíveis aplicações futuras. A pesquisa incluiu seis perguntas, das quais três abordaram eventos ocorridos e as outras três focaram em possíveis eventos futuros e as expectativas dos respondentes.

4.1. Análise e Discussão da Pesquisa

Nesta seção, apresentamos os resultados da pesquisa realizada para entender a percepção dos entrevistados sobre problemas de extravasão nas edificações. A pesquisa foi composta por seis questões principais, cujas respostas são discutidas a seguir.

4.1.1. Questões da Pesquisa

As questões abordadas na pesquisa foram as seguintes:

1. Já teve problema de extravasão na edificação?
2. Ficou ciente imediatamente?
3. O evento de extravasão causou prejuízos?
4. Tem conhecimento de quanto tempo levaria para ficar ciente de uma extravasão?
5. Considera necessário saber o mais rápido o possível sobre um evento de extravasão?
6. Se houver problema de extravasão, quanto tempo seria adequado para conhecimento do responsável? (minutos)

4.1.2. Resultados da Pesquisa

Pesquisa realizada com a ferramenta Google Forms e enviada via email e WhatsApp. Vinte e dois indivíduos responderam a esta pesquisa e aqui apresentamos os resultados obtidos para cada questão, analisando as respostas e discutindo as implicações dos dados coletados. A seguir, discutimos detalhadamente os resultados, Figura 3, Figura 4, Figura 5, Figura 6, Figura 7 e Figura 8:

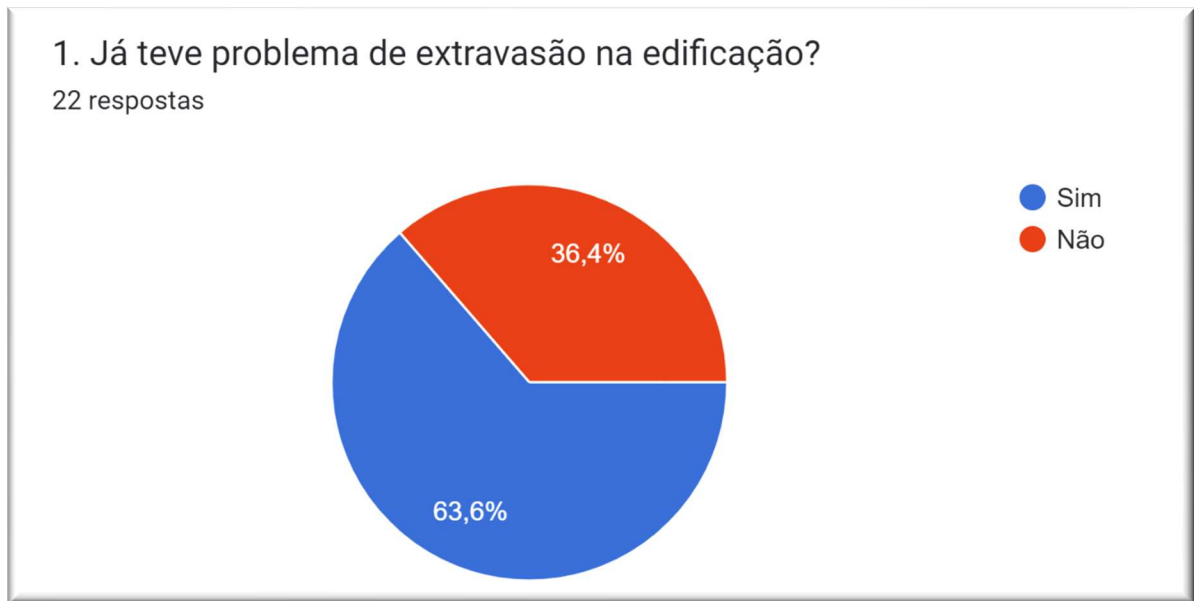


Figura 3. Resultado Questão 1
Fonte: Próprio autor.

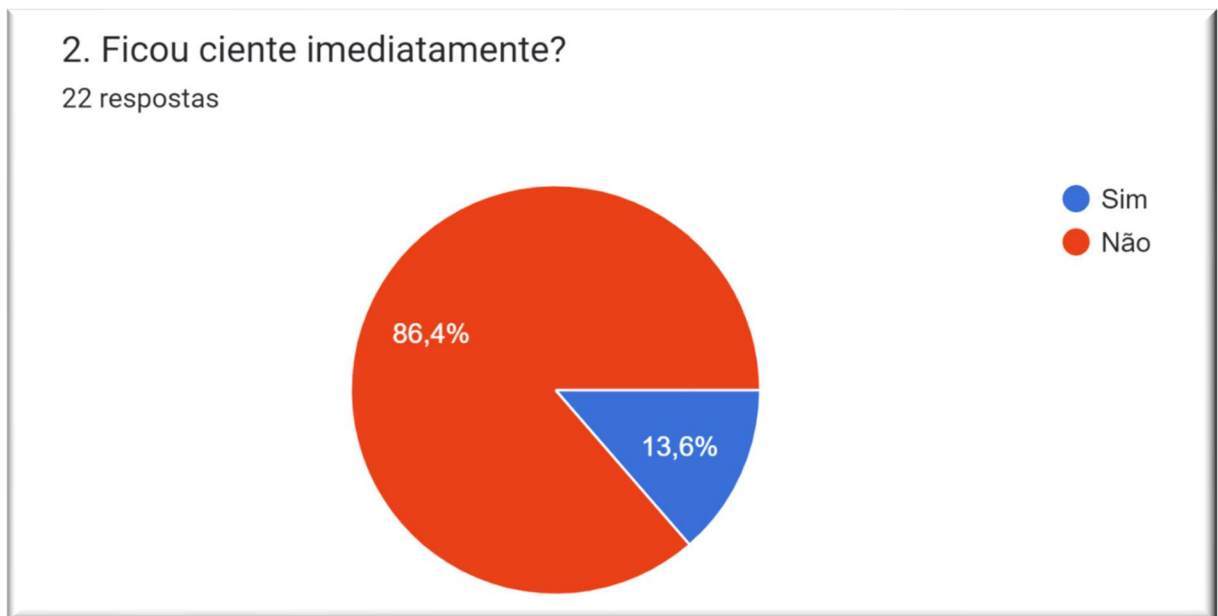


Figura 4. Resultado Questão 2
Fonte: Próprio autor.



Figura 5. Resultado Questão 3
Fonte: Próprio autor.



Figura 6. Resultado Questão 4
Fonte: Próprio autor.

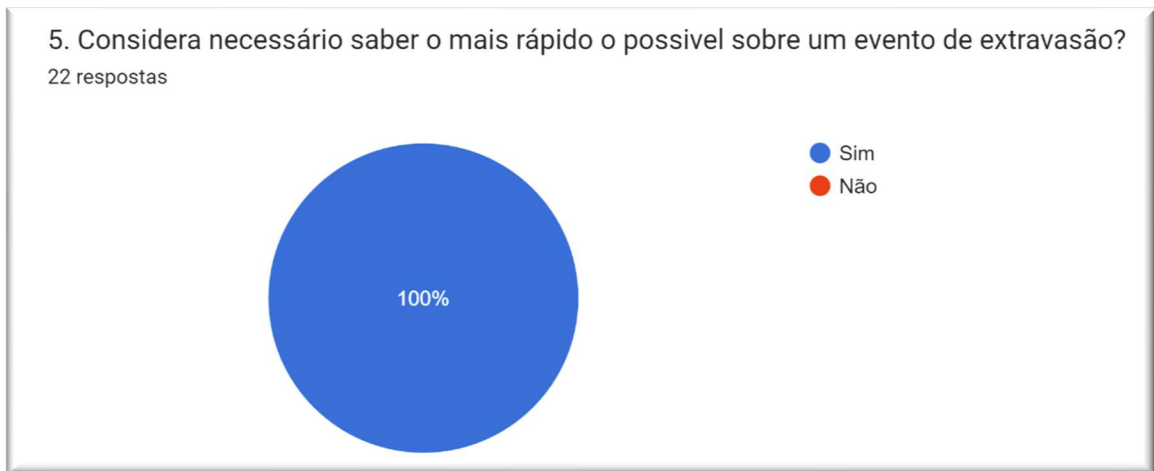


Figura 7. Resultado Questão 5

Fonte: Próprio autor.

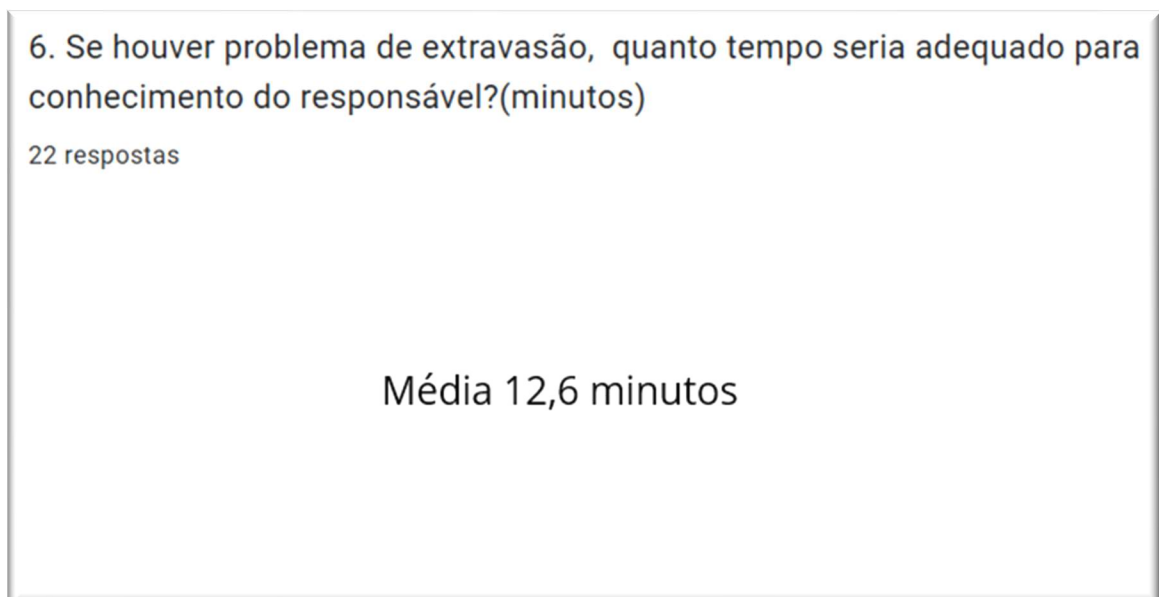


Figura 8. Resultado Questão 6

Fonte: Próprio autor.

Com base na análise dos resultados evidenciamos que a proposta deste sistema se faz relevante, pois dos 40,9% dos entrevistados que enfrentaram problemas de extravasão, nenhum deles ficou ciente imediatamente, 71,43% não tem conhecimento de quanto tempo pode passar até que fique ciente em caso de problema futuro e consideram 16 minutos e 15 segundos em média, aceitável para tal. O presente sistema se mostrou eficiente, com tempo de alerta de até 2 minutos.

5. SOLUÇÃO CONCEITUAL

Após a análise dos trabalhos relacionados e dos dados levantados na pesquisa, foi desenvolvido o sistema aqui proposto para monitorar constantemente o status do reservatório de água. A cada 20 segundos, nosso coletor de dados composto pelo ESP32 lê o status do sensor de nível e envia para a nuvem. Em caso de extravasão, o intervalo será ignorado e o status será imediatamente enviado para o servidor ThingSpeak, onde estará disponível e poderá ser recuperado por um ou mais ESP8266 para exibição visual e/ou sonora do status, como por aplicativo em dispositivo móvel com acesso à internet, conforme ilustrado na Figura 9.

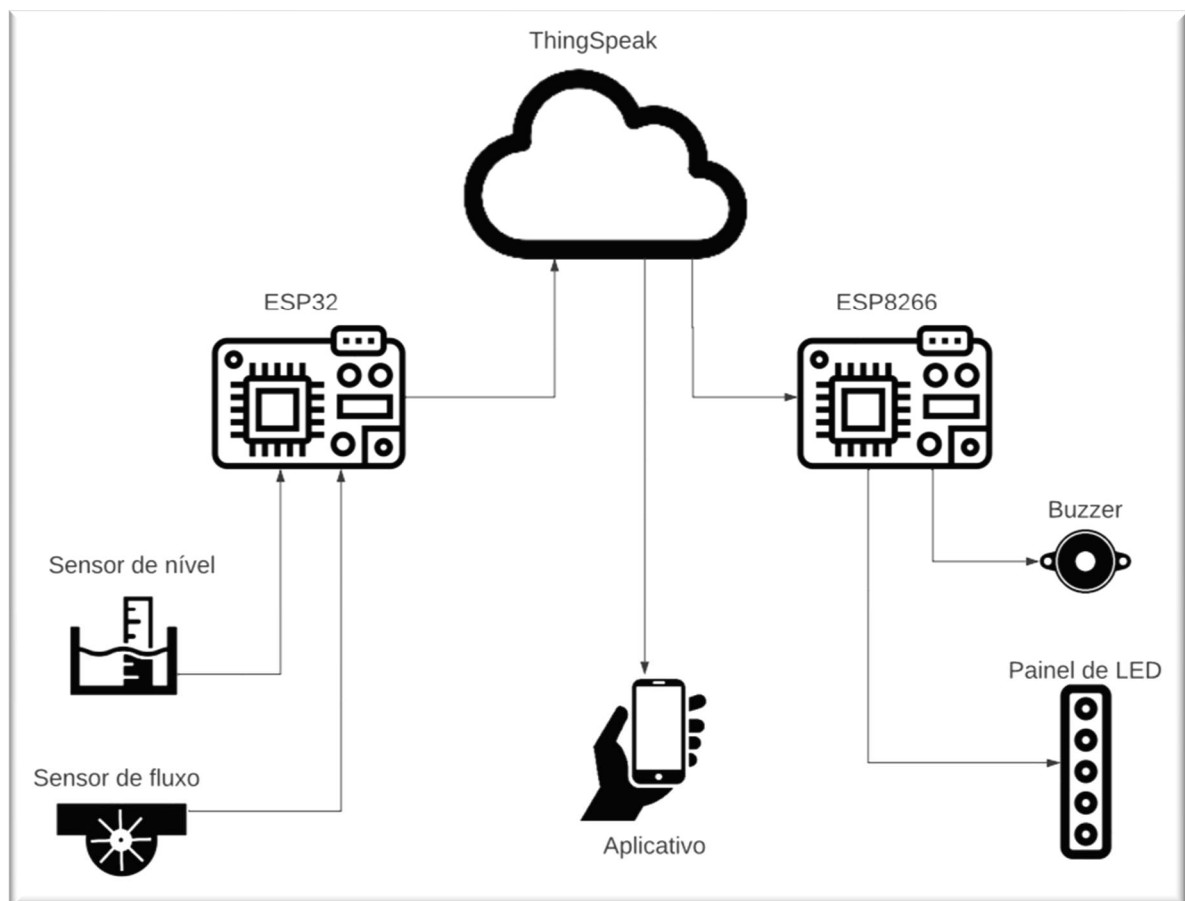


Figura 9. Modelagem conceitual

Fonte: Próprio autor.

Na Figura 10, Figura 11 e Figura 12 o fluxograma de ambos os dispositivos e aplicativo display. O coletor de dados que envia a leitura dos sensores para a nuvem e display, que pode ser um ou mais dispositivos ou um ou mais aplicativos móveis, que resgatam os dados da nuvem, convertendo em sinalização para os usuários.

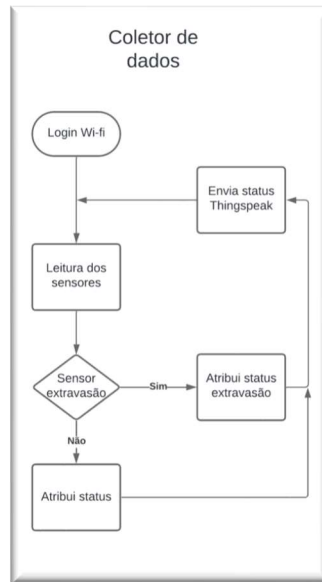


Figura 10. Fluxograma do Coletor de dados

Fonte: Próprio autor.

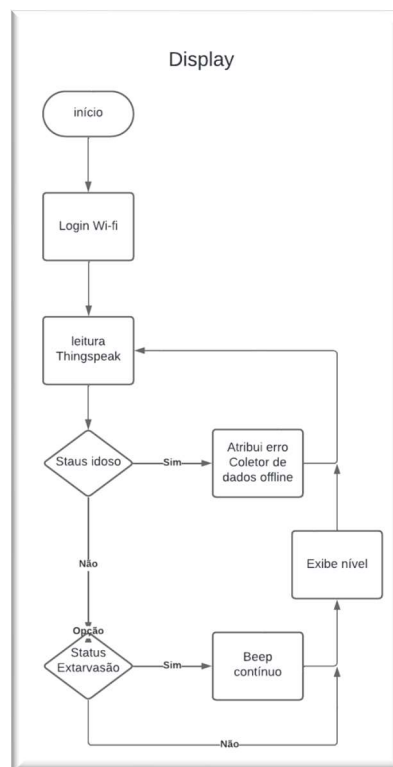


Figura 11. Fluxogramas do Display

Fonte: Próprio autor.

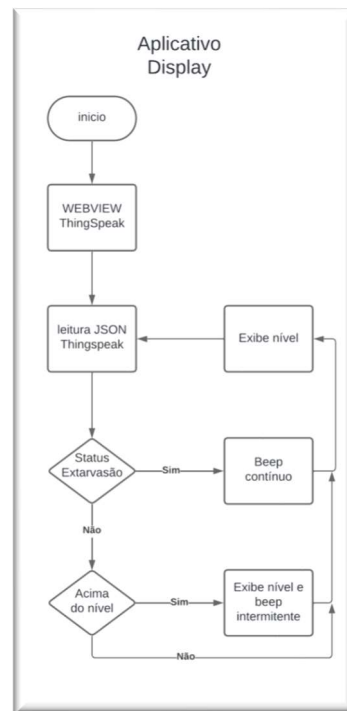


Figura 12. Fluxogramas do aplicativo Display
 Fonte: Próprio autor.

O posicionamento dos sensores é feito de acordo com o reservatório a ser instalado, conforme ilustrado na Figura 13.

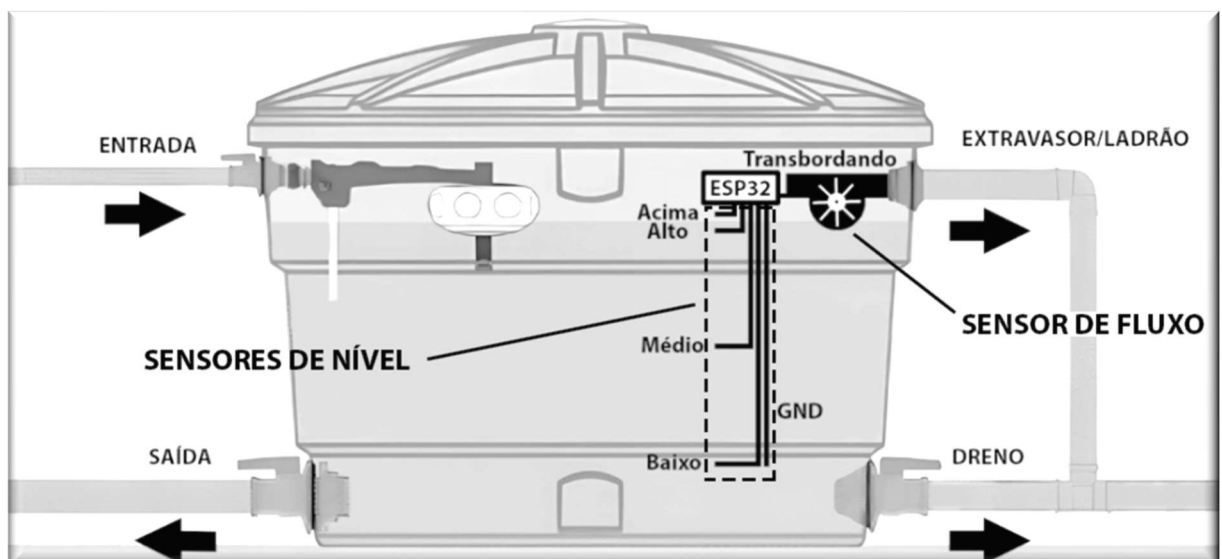


Figura 13. Posicionamento do conjunto de sensores
 Fonte: Adaptação de <https://construwiki.blogspot.com/2015/02/>

6. IMPLEMENTAÇÃO

No presente capítulo, abordamos a implementação dos módulos desenvolvidos para este trabalho de conclusão de curso, que consistem em dois componentes principais: o coletor de dados e o monitor. Ambos os módulos foram construídos utilizando o microcontrolador ESP32 e ESP8266, respectivamente e posteriormente os códigos para eles criados, assim como aplicativo para dispositivos Android.

O coletor de dados tem como função principal a captura e transmissão de informações relevantes, enquanto o monitor é responsável pela recepção e análise e exposição dessas informações. A comunicação entre os módulos é realizada através da plataforma ThingSpeak, que permite a troca de mensagens de forma eficiente e segura via internet.

6.1. Implementação dos Módulos de Coleta de Dados e Monitoramento

Nesta seção, detalharemos o processo de construção e configuração de cada módulo, explicando as etapas envolvidas.

6.1.1. Coletor de dados

O diagrama apresentado na Figura 14 descreve a montagem do módulo Figura 15 utilizado um regulador de 5V, para possibilitar o uso de fontes de alimentação mais populares e baratas. O módulo LC1BD04 com suas entradas no conector J1 e suas saídas nas portas IO 13, 14, 26 e 27. O sensor de fluxo YF-S201 conectado a placa no conector J2 entrega o sinal na porta IO 4. As portas IO 25 e 33 estão ligadas ao LED Bicolor vermelho e verde.

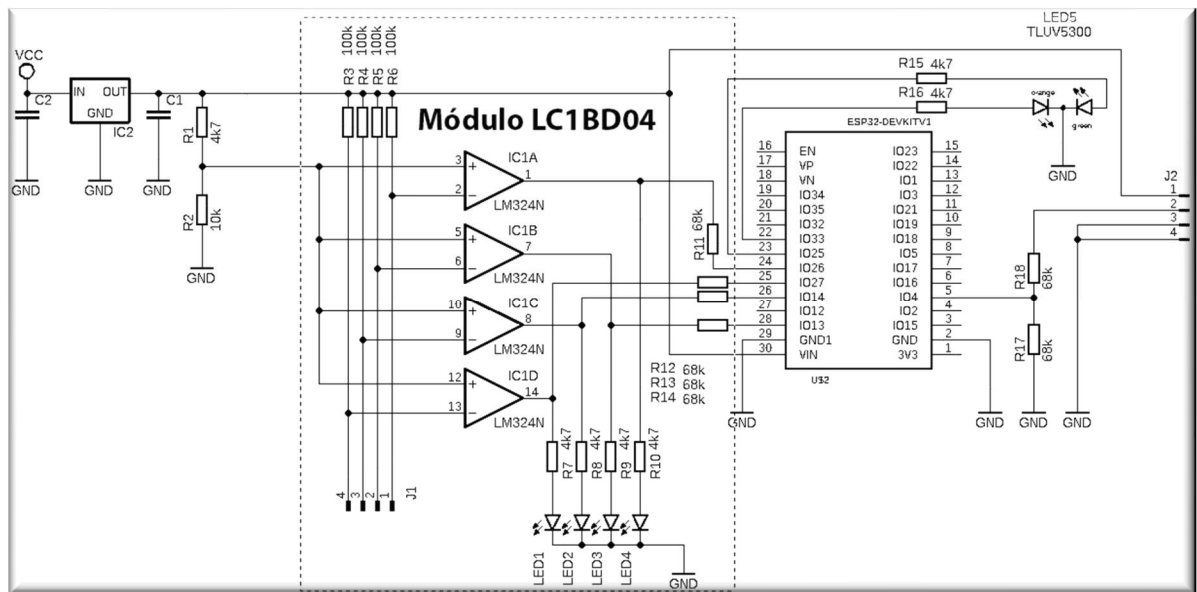


Figura 14. Diagrama esquemático Coletor de dados

Fonte: Próprio autor.

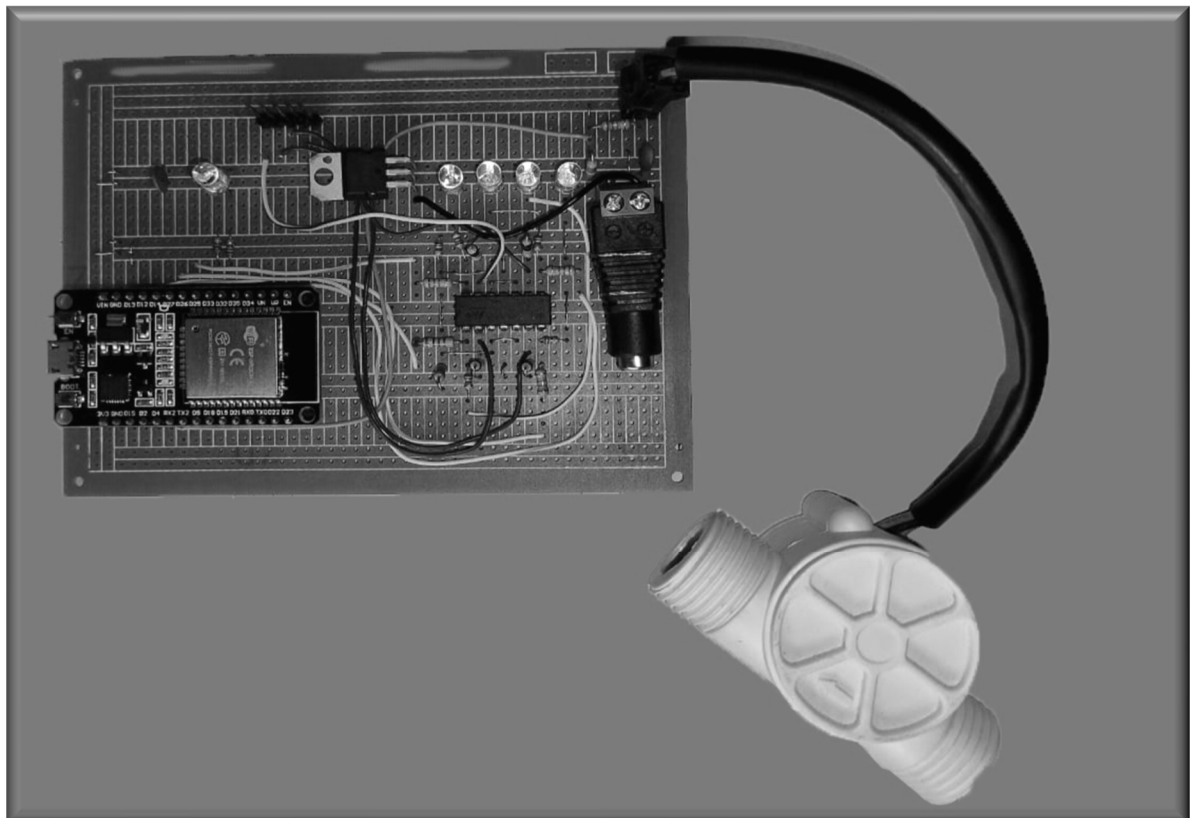


Figura 15. Protótipo Coletor de dados

Fonte: Próprio autor.

6.1.2. Display

Mostrado na Figura 16 a placa do display, que foi montada conforme diagrama da Figura 17, usamos um regulador de 5V para controlar a tensão e possibilitar o uso de fontes genéricas de 12V. Conectamos LEDs de nível nas portas D1, D2, D6 e D7. Já na porta D5, conectamos um LED e um Buzzer, que é um dispositivo que emite sinal sonoro.

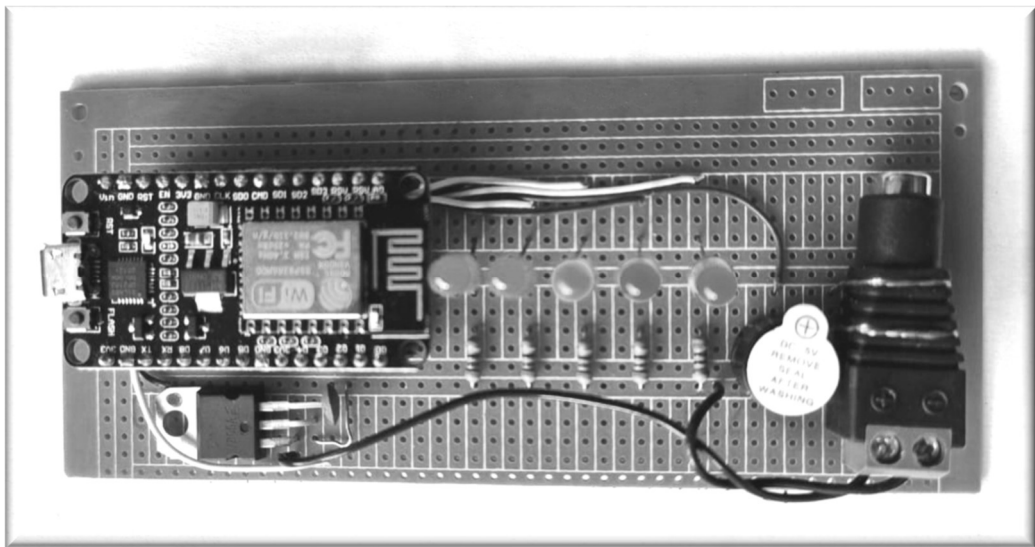


Figura 16. Protótipo Display

Fonte: Próprio autor.

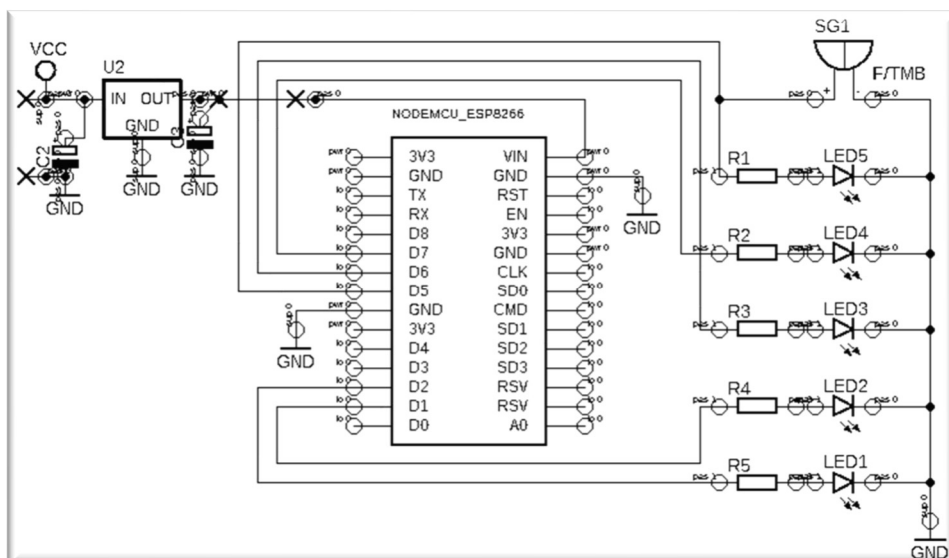


Figura 17. Diagrama esquemático Display

Fonte: Próprio autor.

6.2. Implementação dos códigos de Coleta de Dados e Monitoramento

No desenvolvimento deste projeto, para criar uma solução de Internet das Coisas (IoT). O código-fonte foi desenvolvido para cada dispositivo, considerando suas características únicas e capacidades de conectividade Wi-Fi. Foi utilizado o ambiente de programação do Arduino IDE para escrever código, integrando os sensores. Paralelamente, foi desenvolvido um aplicativo móvel utilizando o MIT App Inventor, uma plataforma visual de programação que facilita a criação de aplicativos Android. O app serve como interface de usuário, permitindo o monitoramento, assim como o módulo que denominamos “Display”. A comunicação entre o app e os microcontroladores é realizada através de protocolo HTTP, utilizando requisições GET e POST para trocar informações entre o Coletor de dados e os displays.

6.2.1. Coletor de dados

Para a elaboração do Código, fizemos a implementação dos recursos disponíveis no ESP32, conforme a Figura 18, na inicialização do sistema, são configurados os sensores, estabelecida a conexão WIFI e inicializado o ThingSpeak. No *loop* principal são lidos os GPIOs referentes sensores e esta informação é enviada ao ThingSpeak, atribuindo também cor para o LED indicador de status. Com o tratamento de interrupção é definida uma *flag* para transbordo, que usa as funções auxiliares para definir o valor da *flag* de transbordo inicial, como *false* e verifica a *flag* de transbordo, quando houver sinal na GPIO do sensor de transbordo, mudará seu estado para *true*.

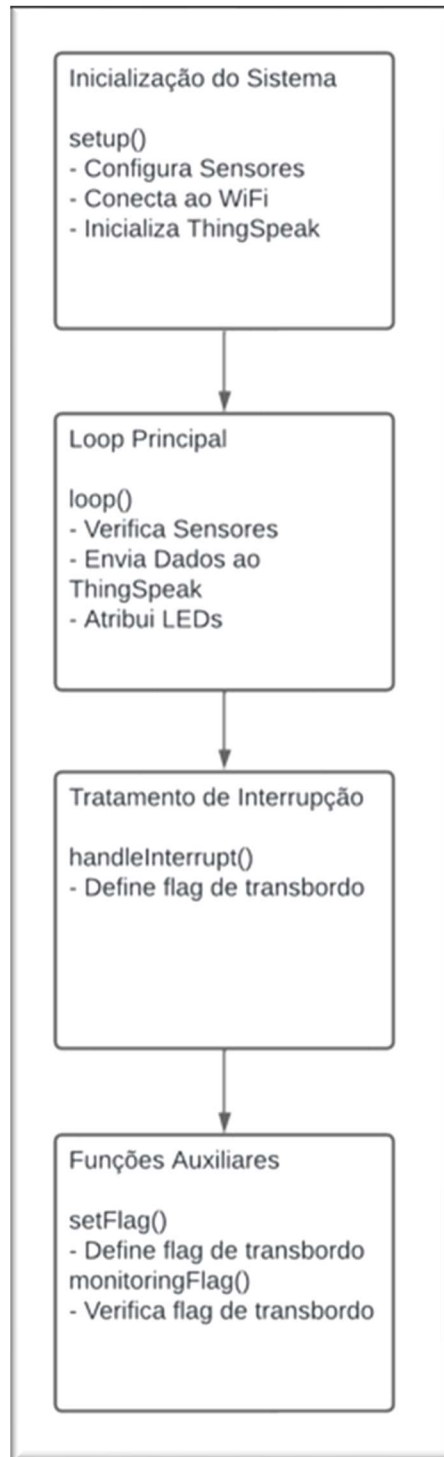


Figura 18. Workflow Coletor de dados
Fonte: Próprio autor.

6.2.2. Display

Este Código foi elaborado para o módulo ESP8266. Ao inicializar o sistema são definidos os LEDs e efetuada a conexão WIFI. Entrando no *Loop* é solicitado o JSON ao ThingSpeak, feita a extração dos dados pertinentes, lido o valor de field1, que recebe o status do reservatório, monitorada a atividade, para identificar se o coletor de dados atualizou o status e controla os LEDs e *Buzzer* para sinalizar o status ao usuário Figura 19.

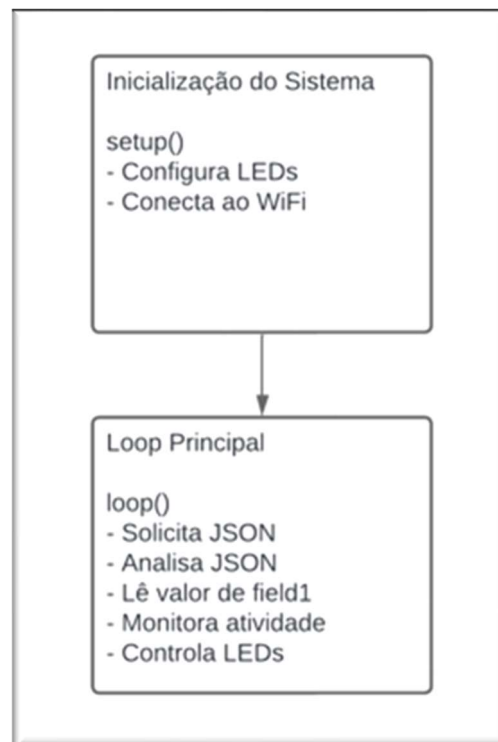


Figura 19. Workflow Display
Fonte: Próprio autor.

6.2.3. Aplicativo Display

APP Display desenvolvido no MIT APP Inventor com o uso de blocos.

Na Figura 20, mostramos o fluxo de trabalho. Ao inicializar o aplicativo é criado um WebView para exibir um gráfico gerado no ThingSpeak. No *loop* principal o JSON é baixado do ThingSpeak, analisado e seu status é exibido.

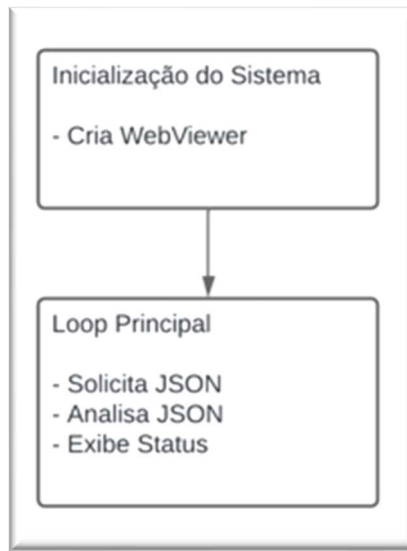


Figura 20. Workflow aplicativo Display
Fonte: Próprio autor.

6.3. JSON extraído do ThingSpeak

JSON retornado pelo ThinSpeak o qual serão extraídos os valores dos campos “entry_id”, e “field” que irão atualizar as variáveis de entrada do Display, para atualizar respectivamente as variáveis id_temp e Count, Figura 21:

```
1 {
2   "channel": {
3     "id": 2273092,
4     "name": "LevelMonitor",
5     "description": "Monitor de nível com alerta de transbordo.",
6     "latitude": "0.0",
7     "longitude": "0.0",
8     "field1": "Level",
9     "created_at": "2023-09-17T16:09:24Z",
10    "updated_at": "2023-09-18T06:39:11Z",
11    "last_entry_id": 6725
12  },
13  "feeds": [
14    {
15      "created_at": "2023-10-01T21:47:55Z",
16      "entry_id": 6725,
17      "field1": "1"
18    }
19  ]
20 }
```

Figura 21. JSON resgatado pelo monitor

Fonte: Próprio autor.

6.4. Instalação do *hardware* de Coleta de Dados e Monitoramento

A instalação foi feita em local próximo ao reservatório de água, observando a cobertura de sinal do WI-FI e um posicionamento adequado para futuros testes ou manutenção, Figura 22.

Ao instalar pode ser observado no coletor de dados os 4 LEDs indicadores de nível, cada led corresponde a um terminal do sensor de nível e um LED bicolor que ficará VERMELHO ao ligar o equipamento, indicando que não há conexão com a internet. Assim que a conexão com a rede WI-FI é estabelecida, o LED fará a transição para verde, indicando que foi enviado com sucesso o status do coletor. Se a conexão com WI-FI for estabelecida e não for bem sucedido o envio dos dados o LED bicolor vai exibir cor amarela.

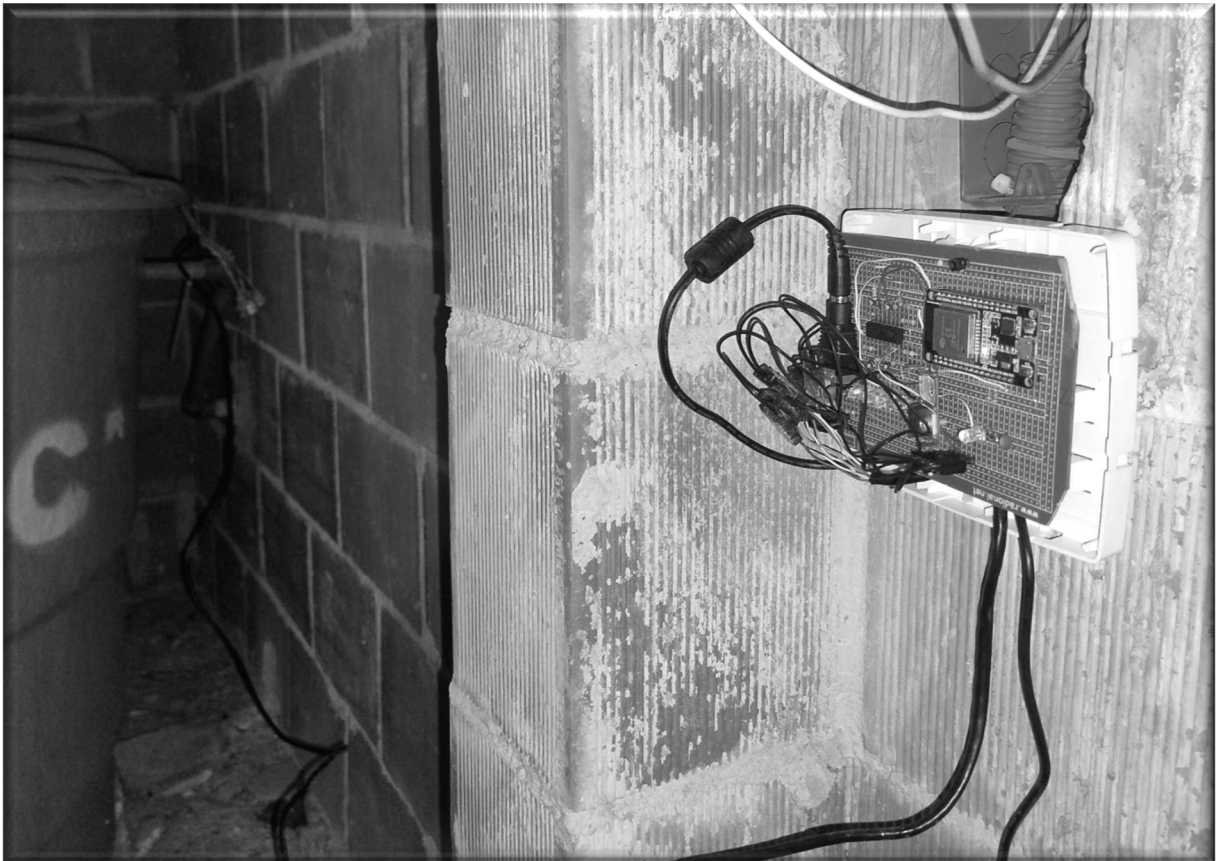


Figura 22. Instalação coletor de dados
Fonte: Próprio autor.

O conjunto de sensores foi instalado junto ao extravasor. Para não reduzir a vazão de saída, foi aberto um orifício na porção superior do redutor usado para a conexão e a detecção de nível feita por cabos de aço inoxidável revestidos de silicone, com suas extremidades expostas, Figura 23.

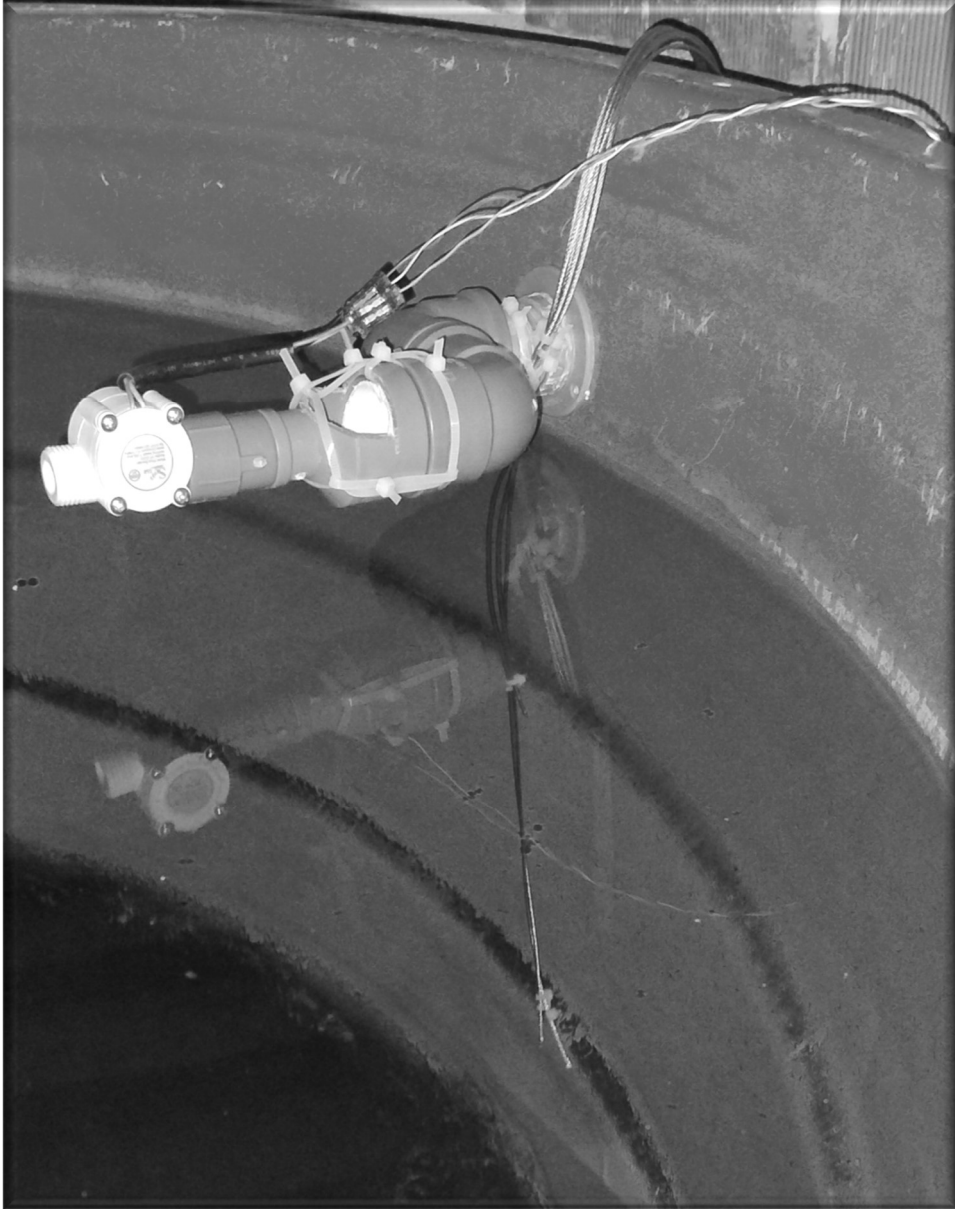


Figura 23. Instalação dos sensores
Fonte: Próprio autor.

6.5. Instalação e uso do *hardware* do Display

O display instalado em local conveniente para o usuário, podendo ser instalados quantos forem necessários, observando o status conforme Figura 24. Com o reservatório vazio irá acender somente o LED 4, os LEDs 1, 2 e 3, indicam nível baixo, médio e alto. Em caso de falha no controle de nível, inicialmente acenderá dos LEDs de 1 a 4 e quando sair água pelo extravasor, todos os 5 LED ficarão acesos e o *Buzzer* será acionado continuamente.

DISPLAY						
	VAZIO	BAIXO	MÉDIO	ALTO	ACIMA	TRANSBORDANDO
BEEP/LED	OFF	OFF	OFF	OFF	OFF	ON
LED 4	ON	OFF	OFF	OFF	ON	ON
LED 3	OFF	OFF	OFF	ON	ON	ON
LED 2	OFF	OFF	ON	ON	ON	ON
LED 1	OFF	ON	ON	ON	ON	ON

Figura 24. Leitura do display

Fonte: Próprio autor.

6.6. Instalação e uso aplicativo Display

Instalado o aplicativo e assim que executado, imediatamente busca atualizações do status continuamente, Figura 25, exibindo o nível do reservatório, VAZIO, BAIXO, MÉDIO, ALTO, quando ACIMA será emitido alarme sonoro intermitente e quando saindo água pelo extravasor, indicará TRANSBORDANDO e o alerta sonoro será contínuo.

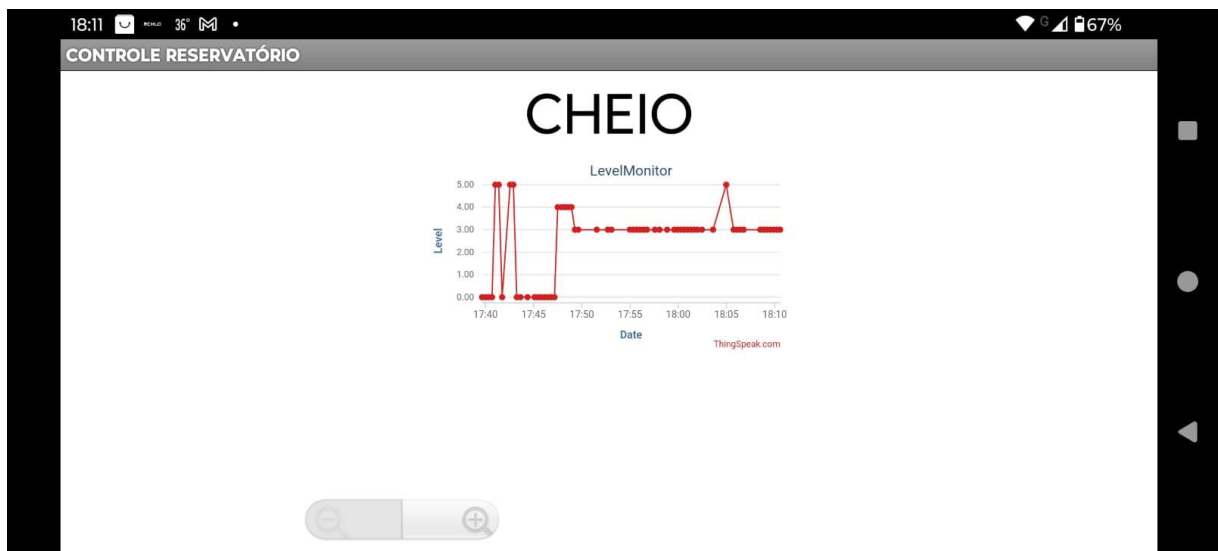


Figura 25. Tela aplicativo display
Fonte: Próprio autor.

6.7. Teste do sistema

Feita simulação alterando a profundidade dos sensores de níveis de status de VAZIO à ACIMA DO NÍVEL foram exibidos corretamente. Para teste do sensor de extravasão, foi derramada uma pequena quantidade de água na entrada do sensor de fluxo e o disparo do alarme tanto no display quanto no aplicativo ocorreu dentro do esperado, com tempos de latência inferiores a 2 minutos.

7. CONCLUSÃO

Com base nos estudos realizados e nos resultados obtidos durante o desenvolvimento, instalação e testes deste projeto, concluímos que o sistema de monitoramento de nível de reservatório de água com alarme de extravasão, utilizando a tecnologia de Internet das Coisas (IoT), é uma solução eficaz para minimizar o desperdício de água potável e reduzir prejuízos financeiros causados por extravasamentos.

Os protótipos foram construídos com componentes confiáveis e aptos para serem colocados em produção, e, futuramente, serão feitas placas de circuito impresso personalizadas. A implementação do módulo ESP32 para coleta contínua de dados, juntamente com os módulos ESP8266 como interfaces homem-máquina (HMI) e o aplicativo de monitoramento em tempo real, resultou em um sistema robusto e confiável.

Os testes confirmaram que o sistema consegue detectar e informar em menos de 2 minutos qualquer evento de extravasão, permitindo a ação imediata do responsável. Dessa forma, o sistema contribui não apenas para a conservação dos recursos hídricos, mas também para a integridade das estruturas das edificações e a prevenção de danos materiais.

Além disso, a utilização da plataforma ThingSpeak mostrou-se adequada para o armazenamento e processamento dos dados, oferecendo uma interface de fácil acesso e uso para os usuários. A integração do sensor de nível de água e do sensor de fluxo também demonstrou precisão e confiabilidade nos dados coletados.

Para o futuro, pretende-se implementar mais funcionalidades no sistema, como o controle de eletroválvulas e motobombas para a automatização de sistemas prediais, e incluir comunicação de rádio direta entre os dispositivos. Esta comunicação, já consagrada na indústria, garantirá o funcionamento do sistema mesmo quando a conexão com a internet não for possível, diferenciando-o das soluções de IoT prontas no mercado. Além disso, a escalabilidade do sistema permite sua aplicação em diversos tipos de edificações, tanto residenciais quanto comerciais.

Este trabalho evidencia a importância da adoção de tecnologias inovadoras e sustentáveis no gerenciamento de recursos naturais, contribuindo para um futuro mais consciente e eficiente no uso da água.

REFERÊNCIAS BIBLIOGRÁFICAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 5626: INSTALAÇÃO PREDIAL DE ÁGUA FRIA: ELABORAÇÃO. RIO DE JANEIRO: ABNT, 1998.

ALVES, Gabriel Belizário. **Monitoramento do Nível de água em reservatórios Residenciais utilizando sensor Ultrassônico**. Orientador: Thiago de Carvalho Batista. 2023. 73f. TCC(Graduação em Engenharia Elétrica) – Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, João Pessoa, 2023.

ARAÚJO, João Luiz Pontes de. **Desenvolvimento de uma estação meteorológica automática embarcada com o uso da plataforma THINGSPEAK no município de Belém – PA**. Orientador: Antônio Thiago Madeira. 2022. 79f. TCC(Graduação em Engenharia Ambiental e Engenharias Renováveis) – Universidade Federal Rural da Amazônia, Belém, 2022.

BABIUCH, M.; FOLTYNEK, P.; SMUTN ` Y, P. Using the esp32 microcontroller for data processing. ` In: IEEE. 2019 20th International Carpathian Control Conference (ICCC). [S.l.], 2019. p. 1–6.

BRASIL. Lei nº 9.433, de 8 de janeiro de 1997. Institui a Política Nacional de Recursos Hídricos, cria o Sistema Nacional de Gerenciamento de Recursos Hídricos, regulamenta o inciso XIX do art. 21 da Constituição Federal, e altera o art. 1º da Lei nº 8.001, de 13 de março de 1990, que modificou a Lei nº 7.990, de 28 de dezembro de 1989. Diário Oficial da União: seção 1, Brasília, DF, 9 jan. 1997. Disponível em: https://www.planalto.gov.br/ccivil_03/leis/19433.htm. Acesso em: 5 nov. 2024.

BROZOSKI, Franciele de Lima; SILVA, Leonaldo Barbosa da Júnior. **Sistema de monitoramento remoto do nível de caixas d´água empregando a tecnologia LORAWAN**. Orientador: Simone Crosetti. 2022. 107f. TCC(Graduação em Engenharia Eletrônica) – Universidade Federal do Paraná, Curitiba, 2022.

ELETROGATE COMPONENTES ELETRÔNICOS.

Disponível em: <https://blog.eletrogate.com/sensor-de-fluxo-de-agua/#:~:text=O%20sensor%20de%20fluxo%20de,um%20sinal%20de%20pulso%20corresponde>. Acesso em: 25 jan. 2025.

ESPRESSIF SYSTEMS. ESP32 Technical Reference.

Disponível em: https://www.espressif.com/sites/default/files/documentation/esp8266-technical_reference_en.pdf. Acesso em: 15 dez. 2024.

INSTITUTO ÁGUA VIVA. A importância da preservação da água: um chamado à consciência.

Disponível em: <https://www.institutoaguaviva.org.br/post/a-importancia-da-preservacao-da-agua-um-chamado-a-consciencia>. Acesso em: 5 nov. 2024.

MANCINI, Mônica. Internet das Coisas: História, Conceitos, Aplicações e Desafios1.

Disponível em: <http://mmproject.com.br/wp-content/uploads/2020/02/artigo-iot-monicamancini-v1.pdf>. Acesso em: 05 nov. 2024.

MOZILLA. MDN Web Docs: JavaScript Object Notation.

Disponível em: https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/JSON?form=MG0AV3. Acesso em: 22 jan. 2025.

MOZILLA. MDN Web Docs: Uma visão geral do HTTP.

Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Overview?form=MG0AV3>. Acesso em: 22 jan. 2025.

Programme, U. et al., 2020. *Relatório mundial das Nações Unidas sobre desenvolvimento dos recursos hídricos 2020: água e mudança climática, fatos e dados*, UNESCO: United Nations Educational, Scientific and Cultural Organisation. Itália. Disponível

em: <https://coilink.org/20.500.12592/jsv83b> Acesso em 05 nov. 2024. COI: 20.500.12592/jsv83b.

SANTAELLA, L.; GALA, A.; POLICARPO, C.; GAZONI, R. Desvelando a internet das coisas. Revista GEMInIS, v. 4, n. 2, p. 19-32, 2013.

SILVA, Fábio de Souza. **Qualidade de serviço em um arede wifi definida por software.** Orientador: Augusto Foronda. 2022. 42f. TCC(Graduação em Ciência da Computação do Departamento Acadêmico de informática) – Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2022.

Saiba mais sobre o ThingSpeak. THISGSPEAK, 2025. Disponível em:

https://thingspeak.mathworks.com/pages/learn_more. Acesso em: 12 jan. 2025.

APÊNDICE A - MONITOR_LEVEL_SENSOR.INO

Código implementado no coletor de dados, Figura 26.

```

1  #include <Arduino.h>
2  #include <WiFi.h>
3  #include <ThingSpeak.h>
4
5  #define SECRET_SSID "SSID da rede WIFI"
6  #define SECRET_PASS "SENHA"
7
8  #define SECRET_CH_ID 69696969
9  #define SECRET_WRITE_APIKEY "CHAVE SECRETA"
10
11 char ssid[] = SECRET_SSID;
12 char pass[] = SECRET_PASS;
13 WiFiClient client;
14
15 unsigned long myChannelNumber = SECRET_CH_ID;
16 const char * myWriteAPIKey = SECRET_WRITE_APIKEY;
17
18 const int sensorPin0 = 26; // baixo
19 const int sensorPin1 = 13; // médio
20 const int sensorPin2 = 14; // alto
21 const int sensorPin3 = 27; // acima
22 const int flowSensorPin4 = 4; // transbordando (sensor de fluxo)
23 const int redPin = 25; // vermelho
24 const int greenPin = 33; // verde
25
26 int number = 0;
27
28 void setup() {
29   Serial.begin(115200);
30   pinMode(sensorPin0, INPUT);
31   pinMode(sensorPin1, INPUT);
32   pinMode(sensorPin2, INPUT);
33   pinMode(sensorPin3, INPUT);
34   pinMode(flowSensorPin4, INPUT);
35   pinMode(redPin, OUTPUT);
36   pinMode(greenPin, OUTPUT);
37   attachInterrupt(flowSensorPin4, handleInterrupt, RISING); //cria interrupção - sensor de fluxo
38   digitalWrite(redPin, HIGH); // inicia com LED vermelho
39
40   WiFi.begin(ssid, pass); // Connect to WiFi network
41   while(WiFi.status() != WL_CONNECTED) {
42     delay(5000);
43   }
44   digitalWrite(redPin, LOW); // se conectado no WI-FI desliga o LED vermelho
45   ThingSpeak.begin(client); // Initialise ThingSpeak
46 }
47
48 void loop() {
49   digitalWrite(greenPin, LOW);
50   digitalWrite(redPin, LOW);
51
52   if(monitoringFlag()) number = 5;
53   else if(digitalRead(sensorPin3) == 1) number = 4;
54   else if(digitalRead(sensorPin2) == 1) number = 3;
55   else if(digitalRead(sensorPin1) == 1) number = 2;
56   else if(digitalRead(sensorPin0) == 1) number = 1;
57   else number = 0;
58
59   setFlag(false); // seta flag de transbordo para FALSE depois setar transbordo para enviar para nuvem
60
61   int x = ThingSpeak.writeField(myChannelNumber, 1, number, myWriteAPIKey);
62   if(x == 200) {
63     digitalWrite(greenPin, HIGH); // se enviado, LED fica verde
64   }
65   else {
66     digitalWrite(redPin, HIGH);
67     digitalWrite(greenPin, HIGH);
68   }
69   delay(20000);
70 }
71
72 void handleInterrupt() {
73   setFlag(true); // Definir a flag como verdadeira
74 }
75

```

Figura 26. MONITOR_LEVEL_SENSOR.INO

Fonte: Próprio autor.

APÊNDICE B - FLAG.INO

Código implementado no coletor de dados, Figura 27.

```
1  boolean flag_transbordo = false;
2
3  void setFlag(boolean status){
4      flag_transbordo = status;
5  }
6
7  boolean getFlag(){
8      return flag_transbordo;
9  }
10
11 bool monitoringFlag(){
12     if(getFlag()){
13         return true;
14     } else{
15         return false;
16     }
17 }
18
19
```

Figura 27. FLAG.INO
Fonte: Próprio autor.

APÊNDICE C - MONITOR_LEVEL_TERMINAL.INO

Código implementado no coletor de dados, Figura 28 e Figura 29.

```

1  #include <Arduino.h>
2  #include <ESP8266WiFi.h>
3  #include <ArduinoJson.h>
4  #include <ESP8266HTTPClient.h>
5
6
7  const char* ssid = "WIFI";
8  const char* password = "4444333221";
9
10 const char* jsonUrl = "http://api.thingspeak.com/channels/2273092/fields/1.json?api_key=69696969&results=1";
11
12 const int ledGreenPin0 = 4; // GPIO 4
13 const int ledGreenPin1 = 5; // GPIO 5
14 const int ledGreenPin2 = 12; // GPIO 12
15 const int ledRedPin3 = 13; // GPIO 13
16 const int ledRedBusserPin4 = 14; // GPIO 14
17
18 int id_temp ;
19 int count = 0 ;
20
21 void setup() {
22   pinMode(ledGreenPin0, OUTPUT);
23   pinMode(ledGreenPin1, OUTPUT);
24   pinMode(ledGreenPin2, OUTPUT);
25   pinMode(ledRedPin3, OUTPUT);
26   pinMode(ledRedBusserPin4, OUTPUT);
27
28   // Conectar à rede Wi-Fi
29   Serial.begin(115200);
30   WiFi.begin(ssid, password);
31   while (WiFi.status() != WL_CONNECTED) {
32     delay(5000);
33     Serial.println("Conectando ao WiFi...");
34   }
35   delay(5000);
36 }
37
38 void loop() {
39   WiFiClient wifiClient; // Criar uma instância WiFiClient
40   HTTPClient http;
41   if (http.begin(wifiClient, jsonUrl)) {
42     int httpCode = http.GET();
43     if (httpCode == HTTP_CODE_OK) {
44       String response = http.getString();
45       http.end(); // Liberar a conexão HTTP
46
47       // Analisar o JSON
48       DynamicJsonDocument doc(1024);
49       DeserializationError error = deserializeJson(doc, response);
50       if (error) {
51         Serial.print("Erro ao analisar JSON: ");
52         Serial.println(error.c_str());
53         return;
54       }
55
56       // Ler o valor de "field1" do JSON como uma string
57       String field1 = doc["feeds"][0]["field1"].as<String>();
58       String entry_id = doc["feeds"][0]["entry_id"].as<String>();
59
60
61       // Converter a string em um inteiro
62       int valor = field1.toInt();
63       int idMsg = entry_id.toInt();
64
65       // Monitora atividade
66       if ( id_temp != idMsg ){
67         id_temp = idMsg;
68         count = 0;
69       }
70       else count++;
71       if ( count > 3 ) valor = 6;
72       Serial.println(count);
73

```

Figura 28. MONITOR_LEVEL_TERMINAL(1 DE 2)

Fonte: Próprio autor.

```

74 // Controlar os LEDs com base no valor lido
75 if (valor == 0) {
76   digitalWrite(ledGreenPin0, LOW);
77   digitalWrite(ledGreenPin1, LOW);
78   digitalWrite(ledGreenPin2, LOW);
79   digitalWrite(ledRedPin3, HIGH);
80   digitalWrite(ledRedBusserPin4, LOW);
81   delay(100);
82 } else if (valor == 1) {
83   digitalWrite(ledGreenPin0, HIGH);
84   digitalWrite(ledGreenPin1, LOW);
85   digitalWrite(ledGreenPin2, LOW);
86   digitalWrite(ledRedPin3, LOW);
87   digitalWrite(ledRedBusserPin4, LOW);
88   delay(1000);
89 } else if (valor == 2) {
90   digitalWrite(ledGreenPin0, HIGH);
91   digitalWrite(ledGreenPin1, HIGH);
92   digitalWrite(ledGreenPin2, LOW);
93   digitalWrite(ledRedPin3, LOW);
94   digitalWrite(ledRedBusserPin4, LOW);
95   delay(1000);
96 } else if (valor == 3) {
97   digitalWrite(ledGreenPin0, HIGH);
98   digitalWrite(ledGreenPin1, HIGH);
99   digitalWrite(ledGreenPin2, HIGH);
100  digitalWrite(ledRedPin3, LOW);
101  digitalWrite(ledRedBusserPin4, LOW);
102  delay(1000);
103 } else if (valor == 4) {
104  digitalWrite(ledGreenPin0, HIGH);
105  digitalWrite(ledGreenPin1, HIGH);
106  digitalWrite(ledGreenPin2, HIGH);
107  digitalWrite(ledRedPin3, HIGH);
108  digitalWrite(ledRedBusserPin4, LOW);
109  delay(1000);
110 } else if (valor == 5) {
111  digitalWrite(ledGreenPin0, HIGH);
112  digitalWrite(ledGreenPin1, HIGH);
113  digitalWrite(ledGreenPin2, HIGH);
114  digitalWrite(ledRedPin3, HIGH);
115  digitalWrite(ledRedBusserPin4, HIGH);
116  delay(1000);
117 } else if (valor == 6) {
118  digitalWrite(ledGreenPin0, HIGH);
119  digitalWrite(ledGreenPin1, HIGH);
120  digitalWrite(ledGreenPin2, HIGH);
121  digitalWrite(ledRedPin3, HIGH);
122  delay(10000);
123  digitalWrite(ledGreenPin0, LOW);
124  digitalWrite(ledGreenPin1, LOW);
125  digitalWrite(ledGreenPin2, LOW);
126  digitalWrite(ledRedPin3, LOW);
127  digitalWrite(ledRedBusserPin4, LOW);
128 }
129 } else {
130   http_end(); // Liberar a conexão HTTP
131 }
132 } else {
133   Serial.println("Falha ao iniciar a conexão HTTP");
134 }
135
136 // Aguardar um tempo antes de fazer a próxima solicitação
137 delay(10000); // Intervalo de 10 segundos (ou ajuste conforme necessário)
138 }
139

```

Figura 29. MONITOR_LEVEL_TERMINAL(2 DE 2)

Fonte: Próprio autor.

APÊNDICE D – APP LEVEL MONITOR

APP LEVELMONITOR em blocos, Figura 30.

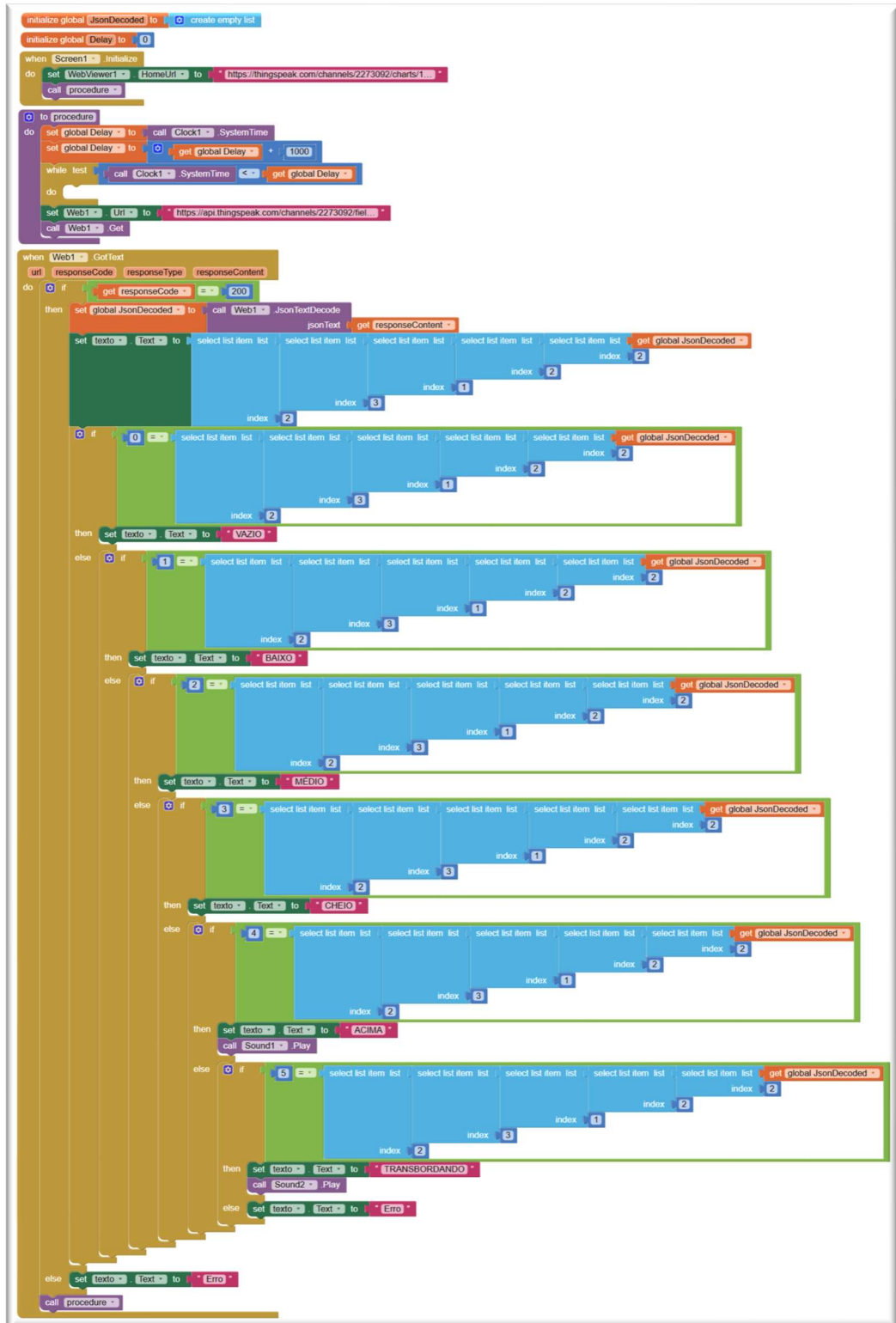


Figura 30. APP LEVELMONITOR

Fonte: Próprio autor.