



FACULDADES
DOM BOSCO

LUCAS BARBOSA E SILVA
LUIS GUILHERME DE ANDRADE SESSO

MONITORAMENTO DO FLUXO E NÍVEL DE ÁGUA EM NASCENTES

Resende - RJ
2024

**ASSOCIAÇÃO EDUCACIONAL DOM BOSCO
FACULDADE DE ENGENHARIA DE RESENDE**

**Lucas Barbosa e Silva
Luis Guilherme de Andrade Sesso**

MONITORAMENTO DO FLUXO E NÍVEL DE ÁGUA EM NASCENTES

Trabalho de Graduação apresentado à Associação Educacional Dom Bosco, Faculdade de Engenharia de Resende, Curso de Engenharia Elétrica com ênfase em eletrônica, como requisito parcial para obtenção do diploma de Bacharel em Engenharia Elétrica.

Resende - RJ
2024

Catálogo na fonte
Biblioteca Central da Associação Educacional Dom Bosco – Resende-RJ

S586 Silva, Lucas Barbosa e
Monitoramento do fluxo e nível de água em nascentes / Lucas
Barbosa e Silva; Luis Guilherme de Andrade Sesso - 2024.
138f.

Orientador: Tiago Duarte Amorim

Trabalho de conclusão de curso apresentado como requisito parcial à
finalização do curso de Engenharia Elétrica da Faculdade de Engenharia
de Resende da Associação Educacional Dom Bosco.

1. Engenharia elétrica. 2. Sensores. 3. Sistema de monitoramento. 4.
Cabeça d'água. 5. Protótipo. I. Sesso, Luis Guilherme de Andrade. II.
Amorim, Tiago Duarte. III. Faculdade de Engenharia de Resende. IV.
Associação Educacional Dom Bosco. V. Título.

CDU 621.38(043)



**LUCAS BARBOSA E SILVA
LUIS GUILHERME DE ANDRADE SESSO**

ESTE TRABALHO DE GRADUAÇÃO FOI JULGADO ADEQUADO COMO
PARTE DO REQUISITO PARA A OBTENÇÃO DO DIPLOMA DE
“GRADUADO EM ENGENHARIA ELÉTRICA”

APROVADO EM SUA FORMA FINAL PELA BANCA EXAMINADORA

BANCA EXAMINADORA:

Prof. (a): TIAGO DUARTE AMORIM
Orientador

Prof. (a): LUIZ FERNANDO RIBAS
MONTEIRO
Membro da Banca

Prof. (a): BIANCA AZEVEDO SALGADO
Membro da Banca

Junho de 2024

AGRADECIMENTOS

Por Lucas Barbosa e Silva

Primeiramente, agradeço a Deus por me guiar e fortalecer durante toda a jornada acadêmica. Sem Sua presença em minha vida, nada disso seria possível.

Agradeço profundamente aos meus pais *Maria de Jesus Pereira Barbosa e Silva* e *Vanderley de Oliveira e Silva*, cujo amor incondicional, apoio e incentivos constantes foram fundamentais para que eu pudesse chegar até aqui. Eles sempre acreditaram em mim e me proporcionaram todas as oportunidades para que eu pudesse crescer e me desenvolver.

À minha namorada, *Ludimilla Luisa de Souza*, por sua paciência, compreensão e incentivo constantes. Seu amor e apoio foram fundamentais nos momentos mais desafiadores.

Um agradecimento especial aos meus professores, que ao longo dos anos compartilharam seu conhecimento e dedicação, contribuindo significativamente para a minha formação. Sou grato por suas orientações e ensinamentos que me ajudaram a alcançar este marco em minha vida.

Quero expressar minha gratidão ao meu orientador de TCC, *Prof. Tiago Amorim*, por sua paciência, orientação e conselhos valiosos ao longo do desenvolvimento deste trabalho. Sua experiência e apoio foram essenciais para a conclusão deste projeto.

Agradeço também ao meu amigo e parceiro de TCC, *Luis Guilherme de Andrade Sesso*, por sua colaboração, amizade e compromisso com o nosso projeto. Trabalhar em conjunto foi uma experiência enriquecedora, e nossa parceria foi crucial para o sucesso deste trabalho e formação acadêmica.

Por fim, sou eternamente grato aos meus amigos de longa data, cujo apoio e amizade foram uma fonte constante de motivação e alegria. Eles me acompanharam em todos os momentos, tanto nos desafios quanto nas conquistas, e por isso, tenho muito a agradecer.

Por Luis Guilherme de Andrade Sesso

A realização deste Trabalho de Conclusão de Curso representa o resultado de uma jornada de aprendizado e crescimento pessoal. Gostaria de expressar minha gratidão a todos que contribuíram para que este momento fosse possível.

Primeiramente, agradeço a Deus e aos meus guias espirituais, pela saúde, força e determinação que me permitiram chegar até aqui.

A minha mãe *Ângela Aparecida de Andrade* e ao meu pai *Luiz Carlos Sesso*, pela educação, apoio incondicional e incentivo constante. Vocês são a base de tudo e me ensinaram o valor do esforço e da dedicação.

A minha irmã *Carla Ferreira dos Santos Sesso* e ao meu primo *Thiago Bertocchi Sesso*, pelo incentivo e apoio durante esta jornada, suas palavras de encorajamento e suporte foram essenciais para a conclusão deste trabalho.

Ao meu orientador, *Tiago Duarte Amorim* pela paciência, orientação e por compartilhar seu conhecimento e experiência. Sem suas valiosas contribuições, este trabalho não teria alcançado o mesmo nível de qualidade.

Agradeço em especial ao meu amigo e parceiro de TCC *Lucas Barbosa e Silva* pela colaboração, motivação, sinergia e pela troca de ideias que enriqueceram este projeto. Aos meus colegas e amigos, pelo companheirismo, pelas discussões construtivas e pelo suporte nos momentos mais difíceis.

À minha namorada, *Mariana Takada de Aguiar* pois seu incentivo e apoio foram fundamentais para que eu pudesse superar os desafios e concluir este trabalho.

À instituição de ensino e aos professores, por proporcionarem um ambiente acadêmico desafiador e estimulante, essencial para o desenvolvimento deste trabalho.

Por fim, agradeço a todos que, direta ou indiretamente, contribuíram para a realização deste TCC. Este trabalho é resultado do esforço coletivo e do apoio de todos vocês. Muito obrigado!

“Que todos os nossos esforços estejam sempre focados no desafio à impossibilidade. Todas as grandes conquistas humanas vieram daquilo que parecia impossível.”

Charles Chaplin

RESUMO

Historicamente são registrados inúmeros acidentes envolvendo fenômenos naturais como por exemplo, trombas d'água e enchentes. Estes desastres naturais trazem uma série de impactos negativos aos locais onde ocorrem, desde perdas materiais até danos irreversíveis como a perda de vidas, que poderiam ser mitigados caso houvesse um aviso prévio referente as variações do nível e fluxo da água das nascentes dos rios por exemplo.

A partir disso, este estudo tem como objetivo desenvolver e implementar um sistema que seja capaz de monitorar o fluxo e nível da água das nascentes dos rios em tempo real, visando mitigar os impactos causados e reduzir os riscos associados a desastres naturais, como as cabeças d'água, e assegurar a segurança das pessoas que estejam expostas aos riscos e as comunidades locais. Para garantir a efetividade do sistema e assegurar o funcionamento preciso, serão utilizados sensores ultrassônicos para medição de distâncias, sensores de umidade do solo e sensores de fluxo de água. Sendo assim, um protótipo será criado com base na plataforma do microcontrolador Arduino, que integrará esses sensores para permitir a coleta contínua de dados referentes as variações do nível e fluxo da água. A comunicação desses dados ocorrerá por meio de módulos e antenas LoRa, facilitando a transmissão em áreas remotas e a longas distâncias, através da tecnologia que permite o envio de dados entre dois pontos localizados a 15Km de distância em relação ao outro.

A análise dos dados coletados através dos sensores, permitirá identificar padrões de comportamento e variações no fluxo e nível da água, possibilitando a correlação com parâmetros detectados durante eventos de cabeças d'água ou enchentes. Com base na avaliação desses dados, será implementado um sistema de alerta capaz de emitir sinais sonoros para alertar sobre aumentos abruptos de fluxo e nível da água, enviando informações para um painel de controle central com um alarme sonoro. Com base em fundamentos teóricos e práticos, observa-se que o monitoramento de nascentes para prevenção e redução dos impactos causados por desastres naturais é essencial para a segurança das pessoas em áreas vulneráveis a cabeças d'água. Portanto, o desenvolvimento desse sistema de monitoramento em tempo real oferece uma solução prática e eficaz para lidar com esse problema.

A relevância deste projeto é clara tanto no campo teórico quanto prático, tendo em vista a correção a áreas como hidrologia, estratégias de prevenção de desastres naturais, tecnologias de monitoramento ambiental e políticas de gestão de recursos hídricos. A adoção de tecnologias eficazes, assertivas e acessíveis, como Arduino e sensores especializados, oferecem quando

integradas, uma abordagem eficiente para o monitoramento preciso do fluxo e nível da água nas nascentes.

PALAVRAS-CHAVE: Sensores; Monitoramento; Sistema de alerta; Cabeças d'água; Segurança hídrica.

ABSTRACT

Historically, numerous accidents involving natural phenomena such as waterspouts and floods have been recorded. These natural disasters bring a series of negative impacts to the areas where they occur, ranging from material losses to irreversible damage, such as the loss of human lives, which could be mitigated if there were prior warnings regarding changes in water level and flow from river springs.

Based on this, this study aims to develop and implement a system capable of monitoring the flow and water level of river springs in real-time, with the goal of mitigating the impacts caused and reducing risks associated with natural disasters, such as flash floods, to ensure the safety of those exposed to these risks and the local communities. To ensure the system's effectiveness and accurate functionality, ultrasonic sensors for distance measurement, soil moisture sensors, and water flow sensors will be used. A prototype will be created based on the Arduino microcontroller platform, integrating these sensors to enable the continuous collection of data related to variations in water level and flow. The communication of this data will occur through LoRa modules and antennas, facilitating transmission over remote areas and long distances, using technology that allows data transmission between two points up to 15 km apart.

The analysis of data collected through the sensors will enable the identification of behavioral patterns and variations in water flow and level, allowing for correlation with parameters detected during flash floods or flood events. Based on the assessment of this data, an alert system will be implemented to emit audible signals warning of abrupt increases in water flow and level, sending information to a central control panel with a sound alarm. Based on theoretical and practical foundations, it is observed that monitoring springs to prevent and reduce the impacts caused by natural disasters is essential for the safety of people in areas vulnerable to flash floods. Therefore, the development of this real-time monitoring system offers a practical and effective solution to address this issue.

The relevance of this project is clear in both theoretical and practical fields, considering its application to areas such as hydrology, natural disaster prevention strategies, environmental monitoring technologies, and water resource management policies. The adoption of effective, assertive, and accessible technologies, such as Arduino and specialized sensors, when integrated, offers an efficient approach for accurately monitoring water flow and level in springs.

KEYWORDS: Sensors; Monitoring; Alert System; Flash Floods; Water Security.

LISTA DE ILUSTRAÇÕES

Figura 1: Atlas Brasileiro de Desastres Naturais – Mapa de Desastres Hidrológicos.....	20
Figura 2: Cabeça d’água surpreende banhistas em Itatiaia - RJ.....	21
Figura 3: Estradas e pontes bloqueadas pela maior enchente do Rio Grande do Sul.....	23
Figura 4: Operadores monitorando dados em tempo real na Sala de Situação do CEMADEN	24
Figura 5: Placa Arduino Uno R3 Atmega328p.....	29
Figura 6: Módulo Lora Mesh.....	30
Figura 7: Especificações Técnicas do Arduino Uno.....	32
Figura 8: Pinagem e Configuração do Arduino Uno Rev3.....	32
Figura 9: Arquitetura de Rede LoRaWAN para Comunicação IoT.....	34
Figura 10: Pinagem do módulo WiFi LoRa 32(V2).....	35
Figura 11: Módulo LoRa com Antena e Pinos de Conexão para Monitoramento Remoto.....	38
Figura 12: Sensor de Umidade do Solo por Resistividade.....	39
Figura 13: Sistema de Monitoramento de Umidade do Solo em Tempo Real.....	41
Figura 14: Arquitetura de Rede de Sensores Sem Fio (WSN) para Monitoramento Ambiental em Tempo Real.....	42
Figura 15: Conectividade e Integração de Dispositivos em uma Rede de Internet das Coisas (IoT).....	48
Figura 16: Funcionamento do Sensor Ultrassônico para Medição de Distância.....	51
Figura 17: Sensor Ultrassônico HC-SR04 para Medição de Distância.....	52
Figura 18: Display LCD 16x2 para Monitoramento de Nível de Água.....	53
Figura 19: Esquema de Conexões do Display LCD 16x2 HD44780 com Arduino.....	54
Figura 20: Diagrama Funcional do Display LCD 16x2 com Controlador e Backlight.....	55
Figura 21: Potenciômetro para Ajuste de Contraste do Display LCD.....	56
Figura 22: Sensor de Umidade do Solo GC-58 para Simulação de Intensidade de Chuva.....	57
Figura 23: Fonte de Alimentação Ajustável para Componentes do Circuito.....	58
Figura 24: Protoboard para Prototipagem de Circuitos.....	60
Figura 25: Fluxograma do Sistema de Monitoramento de Nível e Fluxo de Água com Arduino e LoRa.....	63
Figura 26: Configuração de Teste do Sensor de Nível Ultrassônico com Recipiente de Água.....	65
Figura 27: Monitoramento de Dados de Nível de Água no Monitor Serial com Arduino Uno.....	65
Figura 28: Sensor Ultrassônico JSN-SR04M-2 para Medição de Distância.....	66

Figura 29: Sensor de Fluxo de Água YF-S201 para Medição de Vazão	67
Figura 30: Teste do Sensor de Fluxo de Água YF-S201 com Mangueira para Simulação de Taxas de Fluxo.....	68
Figura 31: Módulo LoRa ESP32 TTGO LILYGO T3 V1.6.1 Pax Counter com Display OLED para Monitoramento em Tempo Real	69
Figura 32: Esquema de Pinagem do Módulo LoRa ESP32 TTGO LILYGO T3 com Funções ADC, GPIO, I2C e SPI.....	71
Figura 33: Montagem do Módulo LoRa ESP32 TTGO LILYGO T3 V1.6.1 Pax Counter como LoRa Sender em Protoboard	72
Figura 34: Conexão do Módulo LoRa ESP32 TTGO LILYGO T3 V1.6.1 Pax Counter como LoRa Receiver via USB-C para Monitoramento.....	74
Figura 35 - Vista superior da ponte localizada na Estrada Resende Riachuelo.....	75
Figura 36 - Análise do local sob a ponte para instalação do sistema de monitoramento	76
Figura 37 - Configuração inicial e programação do sistema com altura de referência	77
Figura 38 - Preparação dos jumpers para conexão do sensor de fluxo no rio	77
Figura 39 - Montagem do sensor de fluxo em um balde com vedação	78
Figura 40 - Sistema de tubulação e montagem do sensor de fluxo no rio.....	78
Figura 41 - Montagem interna do circuito no balde com vedação contra água.....	79
Figura 42 - Instalação final do circuito na ponte com sensor de nível a 2,75 metros do rio....	80
Figura 43: Protótipo Virtual no Tinkercad.....	81
Figura 44: Visão esquemática do circuito.....	82
Figura 45: Teste Inicial do LoRa Sender com Display OLED Indicando Pronto para Uso	83
Figura 46: Teste Inicial do LoRa Receiver com Display OLED Indicando Pronto para Uso..	84
Figura 47: Posicionamento do Sensor de Nível a 147 cm para Simulação de Altura Máxima	84
Figura 48 - Configuração de Altura Máxima de 147 cm no Código para Sensor Ultrassônico	85
Figura 49: Configuração de Teste com Bacia Vazia para Verificação de Leitura do Sensor de Nível	85
Figura 50: Teste de Transmissão de Dados entre LoRa Sender em Condição de Ausência de Água	86
Figura 51: Teste de Transmissão de Dados entre LoRa Receiver em Condição de Ausência de Água	86
Figura 52: Teste de Sensibilidade do Sensor de Nível com 2 cm de Água na Bacia	87
Figura 53: Transmissão de Dados do Sensor de Nível para o LoRa Receiver via LoRa Sender	88

Figura 54: Exibição em Tempo Real dos Dados de Nível de Água no Display do LoRa Receiver	88
Figura 55: Teste de Medição de Vazão com Fluxo Constante de Água através do Sensor YF-S201	89
Figura 56: Exibição da Medição de Vazão no Display do LoRa Sender.....	90
Figura 57: Transmissão e Exibição dos Dados de Vazão em Tempo Real no LoRa Receiver .	90
Figura 58: Simulação de Condição de Alto Fluxo com Bacia Vazia para Teste do Sensor de Vazão	91
Figura 59: Registro de Vazão de 23,20 L/min no LoRa Sender com Nível de Água em Zero	92
Figura 60: Exibição em Tempo Real da Vazão Elevada sem Alteração de Nível de Água no LoRa Receiver	92
Figura 61: Gráfico das Variações de Nível e Vazão de Água em Diferentes Condições de Teste	94
Figura 62 - Exibição dos dados de nível e fluxo no display do módulo LoRa receptor em diferentes horários do dia	95
Figura 63 - Trajeto percorrido para o teste de alcance do sinal LoRa, com distância máxima de 742,1 metros	96
Figura 64 - Registros sequenciais capturados pelo módulo LoRa receptor ao longo do trajeto de teste	97

LISTA DE TABELAS

Tabela 1: Resumo Das Medições Para Cada Condição De Teste	93
---	----

LISTA DE ABREVIATURAS E SIGLAS

ADC	Analog-to-Digital Converter (Conversor Analógico-Digital)
ANA	Agência Nacional de Águas e Saneamento Básico
CEMADEN	Centro Nacional de Monitoramento e Alertas de Desastres Naturais
CNRH	Conselho Nacional de Recursos Hídricos
CSS	Chirp Spread Spectrum (Espectro de Espalhamento Chirp)
DC	Direct Current (Corrente Contínua)
EWS	Early Warning Systems (Sistemas de Alerta Precoce)
GPIO	General-Purpose Input/Output (Entrada/Saída de Uso Geral)
Hz	Hertz (Unidade de Frequência)
I2C	Inter-Integrated Circuit (Circuito Inter-Integrado)
IDE	Integrated Development Environment (Ambiente de Desenvolvimento Integrado)
IoT	Internet of Things
JVM	Java Virtual Machine (Máquina Virtual Java)
L/min	Litros por Minuto
LCD	Liquid Crystal Display (Display de Cristal Líquido)
LiDAR	Light Detection and Ranging
LoRa	Long Range (tecnologia de comunicação de longo alcance)
LoRaWAN	Long Range Wide Area Network (Rede de Área Ampla de Longo Alcance)
LPWA	Low-Power Wide-Area Network (Rede de Área Ampla de Baixo Consumo de Energia)
LPWAN	Low-Power Wide-Area Network (Rede de Área Ampla de Baixo Consumo de Energia)
OLED	Organic Light-Emitting Diode (Diodo Emissor de Luz Orgânico)
PHY	Physical Layer (Camada Física)
PNRH	Política Nacional de Recursos Hídricos
RSSF	Redes de Sensores Sem Fio
SIG	Sistema de Informação Geográfica
SINGREH	Sistema Nacional de Gerenciamento de Recursos Hídricos
SMD	Surface-Mount Device (Dispositivo de Montagem em Superfície)
SNIRH	Sistema Nacional de Informações sobre Recursos Hídricos
SPI	Serial Peripheral Interface (Interface Periférica Serial)
SPICE	Simulation Program with Integrated Circuit Emphasis (Programa de Simulação com Ênfase em Circuitos Integrados)
TTL	Transistor-Transistor Logic (Lógica Transistor-Transistor)
WORA	Write Once, Run Anywhere (Escreva Uma Vez, Execute em Qualquer Lugar)
WSN	Wireless Sensor Networks (Redes de Sensores Sem Fio)

SUMÁRIO

1	INTRODUÇÃO	17
2	FUNDAMENTAÇÃO TEÓRICA	19
2.1	DESASTRES HIDROLÓGICOS.....	19
2.2	ESTUDOS E MONITORAMENTO DE DESASTRES HIDROLÓGICOS	21
2.3	SISTEMAS DE ALERTA PRECOCE (<i>EARLY WARNING SYSTEMS - EWS</i>).....	23
2.4	TEORIA HIDROLÓGICA COM APLICAÇÃO À TECNOLOGIA.....	27
2.5	COMPONENTES ELETRÔNICOS PARA MONITORAMENTO.....	28
2.5.1	Arduino UNO: Características e Aplicações na Prototipagem.....	31
2.5.1.1	Arduíno UNO	31
2.5.2	Tecnologia LoRa: Características e Aplicações no Monitoramento	33
2.5.2.1	Características da LoRa	34
2.5.2.2	Pontos Positivos da Tecnologia LoRa	36
2.5.2.3	Utilização da Tecnologia LoRa.....	36
2.5.3	Sensores de Umidade do Solo: Surgimento e Funcionamento.....	39
2.6	REDES DE SENSORES SEM FIO (WSN) PARA MONITORAMENTO	41
2.7	LINGUAGEM C E SUAS DERIVAÇÕES.....	44
2.7.1	Evolução	45
2.7.2	Aplicações na Engenharia Elétrica	46
2.8	FUNDAMENTOS SOBRE INTERNET DAS COISAS (IOT)	47
3	MATERIAIS E MÉTODOS	49
3.1	MATERIAIS E MÉTODOS DO PROTÓTIPO VIRTUAL	49
3.1.1	Sensor ultrassônico de distância HC - SR04	50
3.1.2	Display LCD 16x2	53
3.1.3	Potenciômetro	55
3.1.4	Sensor de umidade do solo GC-58.....	56
3.1.5	Fonte de Alimentação DC 5A 4 Dígitos Duplo HF-3205S	58
3.1.6	Arduino UNO R3	59
3.1.7	Placa de ensaio protoboard.....	60
3.2	MATERIAIS E MÉTODOS DO PROTÓTIPO FÍSICO.....	61
3.2.1	Módulo Sensor Ultrassônico Impermeável JSN-SR04T/AJ-SR04M.....	64
3.2.2	Sensor de Fluxo de Água YF-S201	67
3.2.3	Programação no Arduino Uno R3 SMD e Integração dos Dados	68
3.2.4	Transmissão de Dados e Monitoramento Remoto	69
3.2.5	Configuração e Detalhes Técnicos do Módulo LoRa ESP32 LILYGO T3	
V1.6.1	Pax Counter.....	70

3.2.6	Módulo LoRa Sender (Transmissor)	71
3.2.7	Módulo LoRa Receiver (Receptor)	73
3.3	INSTALAÇÃO DO SISTEMA DE MONITORAMENTO NA PONTE	75
4	RESULTADOS E DISCUSSÃO	80
4.1	PROJETO VIRTUAL FINAL	80
4.2	PROJETO FÍSICO FINAL	82
4.2.1	Teste Inicial de Configuração e Verificação de Leitura Zero	83
4.2.2	Teste de Detecção de Nível de Água (Baixo)	87
4.2.3	Teste de Medição de Fluxo de Água	89
4.2.4	Simulação de Alto Fluxo e Baixo Nível	91
4.2.5	Tabela de Resultados Obtidos	93
4.2.6	Gráfico de Variação de Nível e Fluxo de Água	93
4.2.7	Análise dos Resultados de Teste	94
4.3	RESULTADOS DO MONITORAMENTO NA PONTE	95
4.3.1	Análise dos Dados Coletados no Monitoramento da Ponte	97
4.3.2	Considerações Sobre os Resultados	99
5	CONCLUSÃO	99
6	INDICAÇÕES PARA TRABALHOS FUTUROS	101
	REFERÊNCIAS	104
	BIBLIOGRAFIA CONSULTADA	110
	APÊNDICE A – Programação do Sistema Virtual Tinkercad	112
	APÊNDICE B – Código Arduíno e Sensores do Sistema Físico	131
	APÊNDICE C – Código Módulo LoRa Transmissor	133
	APÊNDICE D – Código Módulo LoRa Receptor	136

1 INTRODUÇÃO

O aumento do nível e fluxo da água é uma questão de grande relevância no contexto global, não apenas pela necessidade de preservar a água como recurso essencial, mas também pela importância de detectar desastres naturais que podem resultar em perda de vidas. As cabeças d'águas, fenômenos repentinos de aumento do fluxo e nível de água, são exemplos de tais desastres que podem causar sérios danos. Este projeto visa abordar a temática do monitoramento do fluxo e nível de água em nascentes com o objetivo de evitar acidentes decorrentes de cabeças d'águas e preservar vidas, utilizando componentes eletrônicos como Arduino, sensores ultrassônicos de distância, sensores de umidade do solo, sensores de fluxo de água, módulos LoRa e antenas LoRa.

O principal objetivo deste estudo é desenvolver e implementar um sistema de monitoramento eficiente e em tempo real para o fluxo e nível de água em nascentes, visando a detecção de desastres naturais como cabeças d'águas, e assim, contribuir para a preservação da vida na região. Para garantir a eficácia do projeto, a utilização dos sensores ultrassônicos de distância, sensores de umidade do solo e sensores de fluxo de água no monitoramento contínuo das nascentes serão necessárias, com isso, será desenvolvido um protótipo utilizando a plataforma Arduino para integrar os diferentes sensores e coletar dados continuamente, realizando a comunicação dos dados coletados através de módulos LoRa e antenas LoRa, permitindo a transmissão dos dados a longas distâncias e em áreas remotas.

A fim de prever o aumento do nível e fluxo de água, é de extrema importância a análise dos dados coletados para identificar padrões e variabilidades no fluxo e nível de água, correlacionando esses dados com eventos de cabeças d'águas. A partir disto, um sistema de alerta precoce baseado nos dados monitorados para mitigar os danos causados por acidentes e desastres naturais relacionados ao aumento súbito do fluxo e nível de água, será utilizado para transmitir as informações para um painel de controle central com alarme sonoro.

O projeto baseia-se em diversas razões teóricas e práticas que legitimam a sua realização. Primeiramente, a prevenção de acidentes e desastres naturais é essencial para garantir a segurança e a vida, especialmente em áreas suscetíveis a cabeças d'águas. A implementação de um sistema de monitoramento contínuo e em tempo real utilizando tecnologias acessíveis e eficazes pode oferecer uma solução prática para esta necessidade urgente.

Sendo assim, a relevância do problema é evidente tanto em contextos teóricos quanto práticos. O referencial teórico deste projeto abrange estudos de hidrologia, prevenção de desastres naturais, tecnologia de monitoramento ambiental e políticas de gestão de recursos

hídricos. O uso de componentes eletrônicos modernos e acessíveis, como Arduino e sensores específicos, oferece uma abordagem viável para monitorar de forma precisa e contínua o fluxo e nível de água em nascentes.

Para a explicação da problemática deste tema, pode-se pensar da seguinte maneira: "Quais são as metodologias mais eficazes para monitorar o fluxo e nível de água em nascentes utilizando componentes eletrônicos, visando prever acidentes e desastres naturais como cabeças d'águas?".

Desta forma, abrange-se a complexidade do monitoramento hidrológico com tecnologia moderna, exigindo uma análise detalhada das capacidades e limitações dos componentes eletrônicos utilizados, bem como a eficácia do sistema proposto em prevenir desastres naturais. O problema deve ser investigado considerando a viabilidade técnica, a relevância dos dados coletados, a novidade das abordagens propostas, a exequibilidade das metodologias e a oportunidade de aplicação prática dos resultados em sistemas de alerta precoce. O contínuo processo de reflexão e investigação permitirá formular hipóteses e desenvolver soluções inovadoras para o monitoramento eficiente das nascentes, contribuindo significativamente para a segurança pública e a preservação deste recurso vital.

Em suma, este projeto busca responder a um problema central na prevenção de desastres naturais, fornecendo dados e metodologias que podem ser aplicadas para garantir a segurança e a conservação das nascentes, essenciais para a manutenção do ciclo hidrológico e da vida em diversas escalas, contribuindo significativamente para a segurança pública através do levantamento de informações essenciais para a tomada de decisões em políticas de prevenção de desastres naturais.

Neste primeiro capítulo, Introdução, apresenta uma contextualização sobre a importância da gestão de recursos hídricos e a relevância do monitoramento constante para a segurança ambiental e a preservação da vida humana. Este capítulo aborda a problemática que motivou o estudo e define os objetivos gerais e específicos do trabalho, destacando a contribuição do projeto para o avanço de práticas de prevenção e mitigação de desastres naturais.

Em seguida, o capítulo de Fundamentação Teórica proporciona um embasamento sólido sobre os principais conceitos relacionados aos desastres hidrológicos, detalhando ocorrências como inundações, chuvas intensas e cabeças d'água, além de explorar estudos e métodos de monitoramento hidrológico existentes. Este capítulo inclui também uma análise das tecnologias aplicadas em sistemas de monitoramento, como sensores ultrassônicos, sensores de fluxo de água, redes de sensores sem fio (WSN) e a integração de módulos de comunicação como LoRa para transmissão de dados em áreas remotas.

No capítulo de Materiais e Métodos, são descritos detalhadamente os componentes eletrônicos utilizados no desenvolvimento do protótipo, incluindo a plataforma Arduino, sensores de umidade e de fluxo, e os módulos LoRa. São apresentados os procedimentos de montagem e programação do sistema, além da metodologia empregada para a coleta de dados em um ambiente simulado, garantindo a replicabilidade e a robustez dos experimentos.

O capítulo de Resultados e Discussão analisa os dados coletados durante os testes práticos e virtuais, comparando-os com os objetivos propostos e discutindo a eficácia do sistema em monitorar em tempo real variações de nível e vazão de água. São apresentadas tabelas, gráficos e uma interpretação crítica dos resultados, destacando a capacidade do sistema em detectar alterações e emitir alertas preventivos em situações de risco.

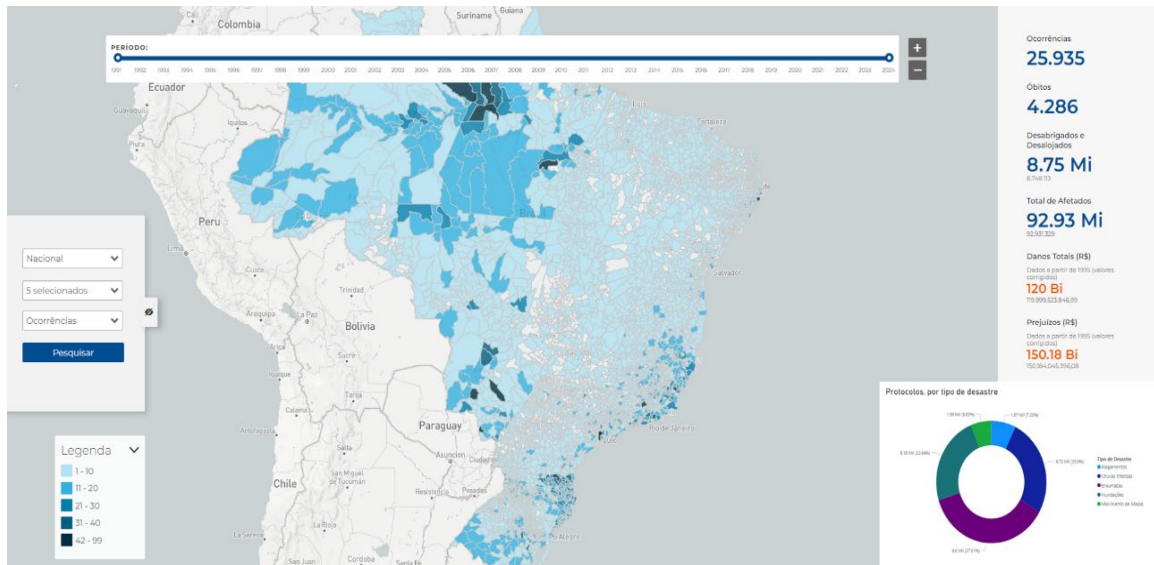
Por fim, a Conclusão sintetiza as principais descobertas, destacando a viabilidade do sistema como uma ferramenta de prevenção e mitigação de desastres naturais. O capítulo discute as limitações encontradas e sugere melhorias para futuras implementações, enfatizando a importância da continuidade de estudos na área de monitoramento hidrológico e a aplicação de novas tecnologias para a gestão de recursos hídricos e a segurança das populações.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 DESASTRES HIDROLÓGICOS

Desastres hidrológicos, como alagamentos, chuvas intensas, enxurradas, inundações e movimentos de massa, têm um impacto significativo em várias regiões do mundo, incluindo o Brasil. Esses eventos naturais ocorrem com frequência e, muitas vezes, com intensidade devastadora. De acordo com dados recentes do Atlas Brasileiro de Desastres Naturais, o Brasil registrou 25.935 ocorrências de desastres hidrológicos, resultando em 4.286 óbitos e deixando 8,75 milhões de pessoas desabrigadas ou desalojadas. No total, 92,93 milhões de pessoas foram afetadas, com danos totais estimados em 120 bilhões de reais e prejuízos atingindo 150,18 bilhões de reais, conforme mostrado na Figura 1. Esses números impressionantes destacam a necessidade urgente de desenvolver e implementar sistemas eficientes de monitoramento e alerta para minimizar os impactos desses desastres.

Figura 1: Atlas Brasileiro de Desastres Naturais – Mapa de Desastres Hidrológicos



Fonte: BRASIL. Ministério da Integração e do Desenvolvimento Regional. Secretaria de Proteção e Defesa Civil. Universidade Federal de Santa Catarina. Centro de Estudos e Pesquisas em Engenharia e Defesa Civil. Atlas Digital de Desastres no Brasil. Brasília: MIDR (2023)

Um exemplo da gravidade desses eventos pode ser visto na Figura 2, onde uma cabeça d'água surpreendeu banhistas e resultou em fatalidades e desaparecimentos em Itatiaia, RJ. Este incidente ressalta a importância de sistemas de alerta precoce para evitar tragédias similares.

Historicamente, a gestão de desastres hidrológicos evoluiu significativamente com o desenvolvimento de novas tecnologias e metodologias. No passado, os métodos de monitoramento eram rudimentares e muitas vezes baseados em observações manuais e registros escritos, o que limitava a precisão e a eficácia das respostas às emergências (Marques, D. M. L. M., & Loureiro, S. R, 2001). Com o avanço da ciência e da tecnologia, surgiram sistemas mais sofisticados, como o uso de radar meteorológico, satélites e modelos de previsão hidrológica (Smith & Ward, 1998). Esses sistemas modernos permitem uma coleta de dados mais precisa e uma análise em tempo real, melhorando a capacidade de resposta a desastres e a elaboração de estratégias de mitigação (Silva, A. M, 2010).

Figura 2: Cabeça d'água surpreende banhistas em Itatiaia - RJ



Fonte: Jornal A Voz da Verdade (2019)

2.2 ESTUDOS E MONITORAMENTO DE DESASTRES HIDROLÓGICOS

O monitoramento de desastres hidrológicos é crucial para a prevenção e mitigação dos danos. Estudos indicam que os sistemas de alerta precoce são fundamentais para salvar vidas e minimizar danos materiais (Tucci, C. E. M, 2007). A fluviometria e a hidrometria são ciências

essenciais para a coleta de dados sobre o fluxo e o volume de água, fundamentais para prever enchentes e inundações. Segundo Ward e Robinson (1990), a análise desses dados permite a formulação de hipóteses e a definição de estratégias eficazes para a gestão dos recursos hídricos. A literatura científica sobre o monitoramento hidrológico é vasta e inclui estudos de caso, análises estatísticas e modelos de previsão que ajudam a entender e prever os comportamentos hidrológicos em diferentes cenários climáticos (Ward & Robinson, 1990).

Sistemas de monitoramento modernos utilizam dados de estações meteorológicas, sensores de nível de água e modelos de previsão hidrológica, o que facilita a tomada de decisão em tempo real e a elaboração de estratégias de resposta a emergências. A integração dessas tecnologias permite uma abordagem mais holística e eficiente na gestão de desastres. Por exemplo, estudos mostram que a combinação de dados de radar meteorológico com modelos hidrológicos pode melhorar significativamente a previsão de enchentes, permitindo uma resposta mais rápida e eficaz (Smith, 2001).

A tragédia das enchentes no Rio Grande do Sul, ocorrida recentemente, como demonstrado na Figura 3, ressalta a necessidade crucial de um monitoramento eficaz do fluxo e do nível de água nas nascentes e rios. Eventos de precipitação intensa, somados à falta de monitoramento contínuo e de sistemas de alerta precoce, podem resultar em inundações devastadoras, como as que ocorreram no estado. O monitoramento constante das nascentes permite a coleta de dados essenciais para prever aumentos nos níveis de água, possibilitando ações preventivas e de mitigação de riscos. Assim, a implementação de tecnologias de monitoramento pode salvar vidas, reduzir perdas materiais e preparar as comunidades para enfrentar eventos extremos com maior resiliência.

Figura 3: Estradas e pontes bloqueadas pela maior enchente do Rio Grande do Sul



Fonte: Poder 360 (2024)

A história do monitoramento hidrológico mostra um progresso constante, desde os métodos manuais até as tecnologias avançadas de sensoriamento remoto e análise de dados em tempo real (Cunha, L. G, 2010). Os primeiros sistemas de monitoramento baseavam-se em observações visuais e registros escritos, mas com o advento da era digital, os sensores eletrônicos e os sistemas de informação geográfica (SIG) tornaram-se ferramentas indispensáveis na coleta e análise de dados hidrológicos (Ward & Robinson, 1990). Atualmente, os pesquisadores e profissionais da área utilizam uma variedade de tecnologias para monitorar e prever desastres hidrológicos, incluindo satélites, drones e redes de sensores conectados via Internet das Coisas (IoT) (Buytaert et al., 2014).

2.3 SISTEMAS DE ALERTA PRECOCE (*EARLY WARNING SYSTEMS - EWS*)

Os Sistemas de Alerta Precoce (EWS) têm se mostrado uma ferramenta essencial na prevenção de desastres hídricos, como enchentes, inundações repentinas e outros eventos que envolvem o aumento do nível das águas. Esses sistemas são formados por um conjunto de sensores, modelos preditivos e sistemas de comunicação que, juntos, monitoram condições ambientais e alertam a população e autoridades com tempo suficiente para a tomada de ações preventivas. A principal vantagem dos EWS é justamente essa capacidade de antecipar eventos

perigosos, o que tem se mostrado eficaz na redução de danos e, principalmente, na preservação de vidas humanas.

Os sistemas de alerta precoce atuam em várias etapas, sendo a primeira delas a detecção de um possível desastre. Isso é feito por meio de sensores instalados em áreas críticas, como rios, bacias hidrográficas e regiões propensas a inundações. Esses sensores podem medir parâmetros como a precipitação, o nível da água e a velocidade do fluxo. De acordo com a Roman (2020), da Agência Nacional de Águas (ANA), "Um Sistema de Alerta te avisa que algo está acontecendo. Uma Sala de Situação monitora o que está acontecendo no presente. Pode até comparar com o passado, mas o foco está no presente e futuro." (ROMAN, 2020, p. 8).

No Brasil, um dos principais centros de monitoramento de desastres naturais é o Centro Nacional de Monitoramento e Alertas de Desastres Naturais (CEMADEN), responsável por monitorar e emitir alertas de riscos de desastres naturais em todo o território nacional. A Sala de Situação do CEMADEN desempenha um papel central no funcionamento do sistema de alerta precoce, permitindo o monitoramento contínuo de fatores como precipitação, nível de rios e deslizamentos de terra.

Figura 4: Operadores monitorando dados em tempo real na Sala de Situação do CEMADEN



Fonte: CEMADEN (2024)

A Figura 4 acima mostra a Sala de Situação do CEMADEN (Centro Nacional de Monitoramento e Alertas de Desastres Naturais), um espaço equipado com múltiplos monitores e estações de trabalho, onde especialistas acompanham, em tempo real, dados hidrológicos e meteorológicos do território brasileiro. No centro da imagem, observa-se operadores

monitorando diversos painéis com gráficos, mapas e informações relacionadas às condições climáticas e aos níveis de rios. O grande painel ao fundo exibe mapas interativos e dados de várias regiões do Brasil, que auxiliam na tomada de decisões e na emissão de alertas para áreas em risco de desastres naturais, como inundações e deslizamentos.

A Sala de Situação é equipada com tecnologias avançadas de sensoriamento remoto, como pluviômetros automáticos, estações meteorológicas, radares e satélites, que fornecem dados em tempo real sobre as condições hidrológicas e meteorológicas do país. Esses dados são continuamente processados por sistemas computacionais e por especialistas em hidrologia e meteorologia, que analisam os riscos iminentes de desastres naturais. O Centro Nacional de Monitoramento e Alertas de Desastres Naturais (2019, p. 9) nos apresenta que,

[...] com a criação do Cemaden, a responsabilidade de implantar um sistema de alertas antecipados da probabilidade de ocorrência de desastres naturais associados aos fenômenos naturais que mais causam vítimas fatais no país, os deslizamentos de encostas e as inundações, por meio da utilização de tecnologias modernas de monitoramento e previsões hidrometeorológicas e geodinâmicas, e do seu aperfeiçoamento contínuo, através da promoção de desenvolvimento científico e tecnológico para avançar na qualidade e confiabilidade dos alertas, e na prevenção e mitigação dos desastres. No escopo do Plano Nacional de Gestão de Riscos e Resposta a Desastres, o Cemaden monitora, atualmente, 958 municípios distribuídos em todas as regiões brasileiras. Os municípios monitorados pelo Cemaden têm mapeamento das áreas de risco identificadas prioritariamente pelo Serviço Geológico do Brasil (CPRM), sendo este um pré-requisito para o município entrar em monitoramento pelo Centro, e têm histórico de registros de desastres naturais decorrentes de movimentos de massa e/ou resultantes de processos hidrológicos (inundações, enxurradas, grandes alagamentos).

A partir das informações coletadas e analisadas, a Sala de Situação emite alertas de risco que são comunicados às Defesas Civis estaduais e municipais. Esse sistema tem sido essencial na prevenção de tragédias em áreas de risco, permitindo uma resposta mais rápida e eficiente para proteger a população.

Além disso, a integração entre a Sala de Situação do CEMADEN e outras entidades governamentais é crucial para garantir a efetividade dos alertas. Esse processo colaborativo permite que as informações sobre os riscos de desastres sejam rapidamente disseminadas para os órgãos responsáveis pela evacuação, planejamento de emergência e assistência às populações afetadas, salvando vidas e reduzindo os danos materiais.

Portanto, a Sala de Situação do CEMADEN é um componente vital dentro do sistema de Sistemas de Alerta Precoce (EWS) no Brasil, demonstrando a importância de um monitoramento hidrológico integrado e coordenado para a prevenção de desastres naturais.

Uma característica fundamental dos EWS é a integração tecnológica com sistemas de comunicação e previsão meteorológica avançada. Esses sistemas, por meio de redes automatizadas, monitoram continuamente as condições atmosféricas e hidrológicas. No Rio de Janeiro, funciona de forma na qual "[...] a ferramenta recebe, em tempo real, informações atualizadas sobre a incidência de chuvas e o nível dos rios monitorados. Esse monitoramento é feito por uma rede de estações telemétricas e de radares meteorológicos." (ENGIE, 2023). Em casos de risco, alertas são emitidos rapidamente, permitindo que a população se prepare para evacuações ou outras medidas de segurança.

Além da detecção, os EWS precisam de uma infraestrutura robusta de comunicação para disseminar os alertas. As tecnologias mais comuns envolvem o uso de redes de rádio, sistemas de mensagens de texto e sirenes instaladas em áreas vulneráveis. Essas redes garantem que os alertas alcancem rapidamente os responsáveis pela gestão de emergências e a população. A eficácia dessas comunicações foi comprovada em diversos cenários de desastres naturais ao redor do mundo. Com o avanço da tecnologia, "O Alerta de Previsão de Inundações do Google distribui avisos via plataforma, com informações detalhadas sobre nível das águas dos rios e previsões de cheias." (ENGIE, 2023).

Exemplos práticos do uso de EWS podem ser observados em diversas regiões no Brasil e no mundo. A Agência Nacional de Águas, por exemplo, utiliza estações pluviométricas automáticas e telemetria para monitorar em tempo real os níveis de rios e reservatórios em várias bacias hidrográficas. Quando um nível crítico é alcançado, os dados são enviados automaticamente para os centros de monitoramento, que emitem alertas às autoridades e à população local. De acordo com o manual de sistemas de alerta da ANA, esses métodos "buscam identificar ocorrências e subsidiar a tomada de decisão para a adoção antecipada de medidas mitigadoras dos efeitos de secas e inundações. Futuramente, essas salas estarão integradas à Sala de Situação da ANA" (ANA, 2024).

Outro exemplo é o sistema desenvolvido em parceria internacional para o monitoramento da brusone no trigo, que ilustra a versatilidade dos EWS. Este sistema, além de prevenir desastres hidrológicos, atua na detecção de condições meteorológicas que favorecem a disseminação de doenças em culturas agrícolas (OPR, 2024). Esse exemplo ressalta como os EWS podem ser adaptados para diferentes áreas de risco, desde desastres naturais até eventos críticos em setores específicos, como a agricultura.

A interoperabilidade dos sistemas de alerta precoce é outro aspecto crucial para seu sucesso. Eles muitas vezes precisam integrar dados de diferentes fontes, como sensores meteorológicos, imagens de satélite e dados históricos, e são capazes de emitir alertas

personalizados para diferentes regiões e níveis de risco. De acordo com o UNDP (2022), "um alerta precoce eficaz depende de uma cooperação e coordenação eficazes entre disciplinas setoriais, bem como entre organismos subnacionais, nacionais, regionais, continentais e internacionais." (UNDP, 2022, p. 16). Esse tipo de cooperação entre entidades regionais e internacionais é essencial para a coordenação e efetividade dos sistemas de alerta.

Por fim, um fator determinante para o sucesso dos Sistemas de Alerta Precoce é a educação e conscientização da população em risco. As comunidades precisam estar cientes dos tipos de alertas e do que fazer ao recebê-los. O apoio de outros países pode influenciar, pois "Construir resiliência em países em dificuldades exige cooperação e apoio internacionais. Um Sistema de Alerta Precoce funcional demonstra o compromisso de um país com a redução de riscos de desastres e a preparação, o que pode atrair apoio internacional [...]", (BERTRAND, 2024). Essa preparação prévia e apoio inclui simulações e programas de treinamento, onde os moradores das áreas vulneráveis são ensinados a identificar os diferentes níveis de alerta e as medidas adequadas a serem tomadas em cada caso.

Em conclusão, os Sistemas de Alerta Precoce (EWS) desempenham um papel central na preservação de vidas humanas em áreas suscetíveis a desastres hidrológicos. A combinação de tecnologias avançadas de monitoramento, previsão e comunicação permite que esses sistemas sejam eficazes na redução dos impactos de desastres naturais. No entanto, sua eficácia depende não apenas da tecnologia empregada, mas também da integração com as autoridades locais e da preparação das comunidades para agir rapidamente em resposta aos alertas.

2.4 TEORIA HIDROLÓGICA COM APLICAÇÃO À TECNOLOGIA

No desenvolvimento de um sistema eficaz de monitoramento do fluxo e nível de água em nascentes, é fundamental organizar e relacionar os dados coletados diretamente com os problemas específicos de gestão hídrica. A literatura existente oferece uma vasta gama de estudos e pesquisas que contribuem para a compreensão e solução desses problemas (Feitosa, F. A. C., & Manoel, M. T, 2001). A definição clara de conceitos e terminologias utilizadas no monitoramento é essencial para garantir a precisão e a eficácia do sistema. Entre os principais autores que contribuíram para esta área estão Smith & Ward (1998), que abordaram os princípios da hidrologia aplicada, e Ward & Robinson (1990), que focaram nos métodos de monitoramento e análise hidrológica.

A síntese dos resultados obtidos a partir dos dados e estudos deve ser feita de maneira que facilite a compreensão e a aplicação prática das informações. Analisar diferentes abordagens e

opiniões na literatura existente permite uma visão mais ampla e crítica sobre o tema. A elaboração de uma bibliografia completa das fontes consultadas garante a integridade e a credibilidade do referencial teórico. Seguindo essas diretrizes, o referencial teórico fornecerá uma base sólida para o desenvolvimento de um sistema de monitoramento do fluxo e nível de água em nascentes, utilizando componentes eletrônicos para a coleta e análise de dados em tempo real, contribuindo significativamente para a prevenção e mitigação dos impactos de desastres hidrológicos (Banzi, 2011).

Além disso, é importante considerar a evolução histórica dos métodos de monitoramento e como as novas tecnologias têm sido integradas nas práticas de gestão hídrica. A transição de métodos manuais para sistemas automatizados e digitais representa um avanço significativo, proporcionando maior precisão e capacidade de resposta. A literatura científica fornece exemplos de casos de sucesso e desafios enfrentados na implementação dessas tecnologias, oferecendo valiosas lições para futuras iniciativas de monitoramento (Buytaert et al., 2014).

Ao abordar o desenvolvimento e a aplicação de tecnologias no monitoramento de desastres hidrológicos, é crucial considerar também os aspectos econômicos, sociais e ambientais. A implementação de sistemas de monitoramento deve ser sustentável e acessível, considerando as limitações de recursos e a necessidade de capacitação.

2.5 COMPONENTES ELETRÔNICOS PARA MONITORAMENTO

A tecnologia desempenha um papel vital no monitoramento dos recursos hidrológicos. Entre os componentes eletrônicos mais utilizados, estão o Arduino, sensores ultrassônicos, sensores de fluxo de água, sensores de umidade do solo e módulos de Lora. O Arduino, conforme Figura 5, uma plataforma de prototipagem eletrônica de código aberto, é amplamente utilizado devido à sua flexibilidade e facilidade de uso. Ele permite a integração de diversos sensores e a programação personalizada para atender às necessidades específicas do monitoramento (Banzi, 2011).

Figura 5: Placa Arduino Uno R3 Atmega328p



Fonte: Embarcados – Fábio Souza (2012)

Os sensores ultrassônicos são empregados para medir a distância até a superfície da água, permitindo a detecção de variações no nível da água. Esses sensores funcionam emitindo pulsos ultrassônicos e medindo o tempo que leva para esses pulsos retornarem após refletirem na superfície da água. A precisão e a confiabilidade desses sensores os tornam ideais para o monitoramento de nascentes e outros corpos d'água. Estudos mostram que esses sensores são eficazes em ambientes variados e podem fornecer dados precisos para a gestão de recursos hídricos. Por exemplo, pesquisas realizadas em regiões propensas a enchentes demonstraram que o uso de sensores ultrassônicos pode melhorar significativamente a capacidade de prever e responder a eventos de alta precipitação (Bovensmann et al., 2009).

Sensores de fluxo de água medem a quantidade de água que passa por um ponto específico, uma informação crucial para a gestão de nascentes e cursos d'água. Esses sensores utilizam diferentes princípios de funcionamento, como a medição de velocidade e volume, para fornecer dados precisos sobre o fluxo de água. A literatura aponta que a utilização desses sensores tem melhorado significativamente a precisão dos dados hidrológicos, permitindo uma gestão mais eficiente dos recursos hídricos (Smith & Ward, 1998). Em estudos de caso, sensores de fluxo de água foram utilizados com sucesso para monitorar o fluxo em rios e canais de irrigação, ajudando a otimizar o uso da água e prevenir desperdícios (Buytaert et al., 2014).

Sensores de umidade do solo são essenciais para avaliar a quantidade de água no solo, ajudando a prever deslizamentos de terra e enxurradas. Esses sensores funcionam medindo a

condutividade elétrica do solo, que varia de acordo com o conteúdo de água. Eles têm sido amplamente utilizados em estudos de agricultura e gestão ambiental, demonstrando sua eficácia na coleta de dados precisos sobre a umidade do solo. Por exemplo, em projetos de manejo sustentável da terra, sensores de umidade do solo têm sido usados para monitorar a saúde do solo e otimizar as práticas de irrigação (Smith, 2001).

A Figura 6 mostra o módulo de Lora, utilizado para comunicação de longa distância entre sensores e unidades de controle, sendo ideal para monitoramento em áreas remotas. Este módulo utiliza a tecnologia de rádio de baixa potência e longa distância, permitindo a transmissão de dados em tempo real, mesmo em áreas com infraestrutura limitada. Ele permite a transmissão de dados em tempo real, melhorando a capacidade de resposta a eventos hidrológicos. Em estudos de monitoramento ambiental, módulos de Lora foram implementados com sucesso para criar redes de sensores que cobrem grandes áreas geográficas, fornecendo dados contínuos e em tempo real para a gestão de recursos naturais (Bovensmann et al., 2009).

Figura 6: Módulo Lora Mesh



Fonte: Mercado Livre (2023)

Historicamente, o uso de tecnologia no monitoramento hidrológico tem mostrado uma evolução constante, desde os métodos manuais até as tecnologias avançadas atuais (Andrade, E. M, 2008). A integração de componentes eletrônicos no monitoramento de nascentes representa um avanço significativo, permitindo a coleta de dados precisos e em tempo real, essencial para a gestão eficaz dos recursos hídricos. A evolução dessas tecnologias continua a

abrir novas possibilidades para o monitoramento ambiental e a gestão sustentável dos recursos naturais (Buytaert et al., 2014).

2.5.1 Arduino UNO: Características e Aplicações na Prototipagem

Criado no ano de 2005 na Itália, o Arduino foi desenvolvido com o intuito de integrar em uma só plataforma o *hardware* e *Software* de fácil utilização, que permitisse a realização de prototipagem rápida aos usuários (ARDUINO, 2021). Desde a sua criação, o Arduino se popularizou e passou a ser utilizado em inúmeros projetos de diferentes aplicações, tornando-se um componente imprescindível para o êxito do funcionamento destes projetos e devido a sua popularização, o Arduino começou a se adaptar a diferentes tipos de utilização e necessidades, então, a partir disso temos alguns tipos deste componente para diferentes tipos de aplicabilidades, sendo os principais:

Arduino Pro Mini – *Hardware* compacto, de baixo custo e apropriado para usuários avançados.

Arduino Fio – *Hardware* mediano, com custo acessível e apropriado para aplicações wireless (sem fio).

Arduino Mega – *Hardware* robusto, com custo elevado, apropriado para projetos grandes devido a sua capacidade de memória.

Arduino UNO – *Hardware* compacto, com boa capacidade de processamento, custo acessível e apropriado para iniciantes, devido a simplicidade da linguagem de programação.

De acordo com Marcos de Lima Carlos (2021), tendo em vista os diferentes tipos de Arduino, sua vasta dimensão de possibilidades de aplicação, a facilidade de manuseio, simplicidade de programação, custo acessível e efetividade, o Arduino começou a ser utilizado por estudantes de universidades, por programadores e profissionais da área de tecnologia para desenvolver diferentes tipos de projetos, inclusive para solução de problemas e automatização de processos.

2.5.1.1 Arduíno UNO

O Arduino UNO é o mais popular e utilizado entre os demais, devido a sua capacidade de acoplamento com periféricos, como sensores, *display* e placas, por se tratar de uma plataforma de prototipagem de código aberto que permite o estudo e modificação de seus códigos, por possuir um microcontrolador potente, hardware robusto, custo acessível e

facilidade de uso em qualquer sistema operacional, sendo Mac, Windows ou Linux, o que permite a utilização em diversos tipos de aplicação e finalidades diferentes. Como ilustrado na Figura 7, apresenta especificações técnicas robustas, incluindo o microcontrolador ATmega328P, clock de 16 MHz e tensão de operação de 5V, tornando-o ideal para aplicações diversas com periféricos como sensores e displays (Carlos, 2021).

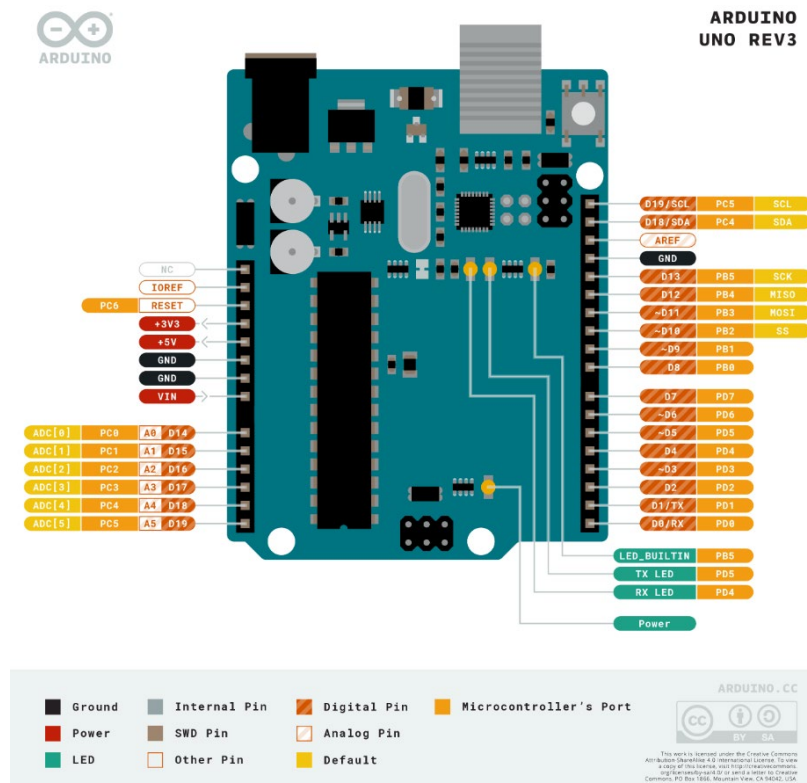
Figura 7: Especificações Técnicas do Arduino Uno

Arduino	Microcontrolador	Memória	Clock	Portas	Tensão de entrada	Tensão de Operação	Corrente
UNO	ATmega328P	Flash: 32KN SRAM: 2.5KB EEPROM: 1KB	16 MHz	Digital: 14 Analogico: 6 PWM: 6	7 - 12V	5V	20mA

Fonte: Smartkits (2024)

As placas de Arduino são muito acessíveis em comparação a outros microcontroladores e ao mesmo tempo muito eficazes, devido a sua alta capacidade de adequação a situações distintas. como exemplificado na Figura 8, são acessíveis e versáteis, permitindo adaptações para aplicações variadas, desde projetos acadêmicos até soluções industriais.

Figura 8: Pinagem e Configuração do Arduino Uno Rev3



Fonte: Smartkits (2024)

Paralelo a isto, o *Software* presente no Arduino é de fácil compreensão e adequado para iniciantes, mas ao mesmo tempo, consegue suprir as exigências de necessidades mais robustas e complexas, desde projetos acadêmicos até soluções de problemas industriais.

A linguagem de programação utilizado pelo Arduino é a C++, oriunda da linguagem de programação C e mundialmente conhecida por ser a base para muitas linguagens. Entretanto, apesar de conhecida, a linguagem C++ necessita de um certo entendimento para realizar a sua manipulação, porém, devido a utilização do modelo de código aberto, o Arduino permite a modificação e compartilhamento dos códigos criados, o que torna o aprendizado e adaptação mais eficaz e rápida, tendo em vista que possibilidades de testes, alterações e adaptações nos códigos, desta maneira, sendo possível adequar para diferentes tipos de necessidades e conseqüentemente agregando diretamente para o aperfeiçoamento através da prática (McRoberts, 2015).

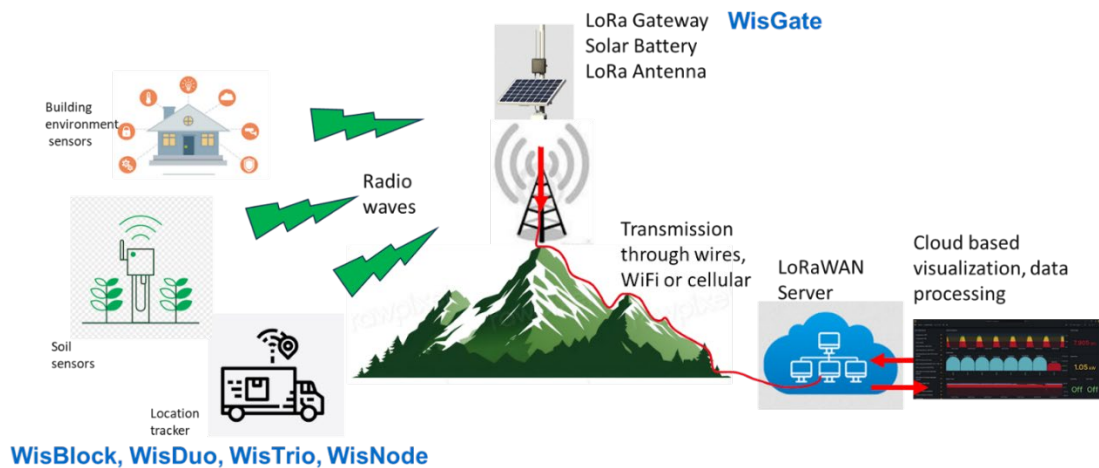
2.5.2 Tecnologia LoRa: Características e Aplicações no Monitoramento

A tecnologia LoRa, que surgiu através da expressão em inglês *Long Range*, tem como suas principais características ser uma rede de baixo consumo de energia e que oferece uma área de conexão ampla, tendo o seu funcionamento através da comunicação sem fio e operação através da rádio frequência. Desta forma, esta tecnologia permite a comunicação a longas distâncias, segurança de rede, baixo custo de implementação, facilidade de instalação, baixo consumo de energia e baixa taxa de utilização de bits (GIL; HENRIQUE; TATEOKI, 2020).

Conhecida mundialmente como *Low-power wide-area network* (LPWA), a tecnologia LoRa utiliza dos conceitos LPWA que realizam conexões sem fio em aparelhos que são operados por bateria e que utilizam internet em redes regionais, nacionais ou globais, como por exemplo, sensores que atendem aos principais requisitos da internet das coisas (IoT), como a como bi-comunicação direcional, segurança ponta a ponta, serviços de mobilidade e localização, permitindo uma comunicação de longo alcance com baixa taxa de bits entre os dispositivos IoT. Sendo assim, a tecnologia LoRaWAN opera através da radiofrequência e possui a capacidade de alcance de até 15 km em áreas rurais, garantindo o baixo consumo de energia e a capacidade de conectar um grande número de dispositivos a uma única rede. A arquitetura inclui dispositivos finais, *gateways*, servidores de rede e servidores de aplicativos, permitindo uma rede escalável e flexível para diversas aplicações do conceito de IoT. A Figura 9 ilustra a arquitetura de uma rede LoRaWAN, que utiliza *gateways* e servidores para

possibilitar a comunicação de longo alcance entre dispositivos IoT em diferentes ambientes (SANTOS, 2022).

Figura 9: Arquitetura de Rede LoRaWAN para Comunicação IoT



Fonte: RAKwireless Technology Limited (2024)

A utilização da tecnologia LoRa, surgiu em meados do ano de 2012 com o intuito de solucionar os problemas encontrados para a transmissão de dados em longa distâncias com baixo consumo de energia. Então, foi desenvolvida pela empresa Cycleo e adquirida pela empresa Semtech, que promoveu o crescimento e aplicação da tecnologia LoRa em grande escala, tornando-a uma das principais tecnologias para criação de redes de comunicação no padrão LoRaWAN (*Long Range Wide Area Network*), que é amplamente utilizada em aplicações de Internet das Coisas (IoT).

2.5.2.1 Características da LoRa

A LoRa opera na camada física (PHY) do modelo de comunicação, realizando uma conexão entre a subcamada MAC e o canal do rádio, oferecendo uma série de serviços e mecanismos de controle de nível físico essenciais para as camadas superiores da arquitetura, sendo a responsável pela modulação do sinal e pela transmissão de dados entre dispositivos. Sua principal função é permitir a comunicação de longa distância de até 20 km em ambientes rurais e até 5 km em áreas urbanas, com baixo consumo de energia, uma característica essencial para dispositivos de IoT que dependem de baterias e precisam de longa autonomia, utilizando a técnica de modulação CSS (*Chirp Spread Spectrum*), que permite maior sensibilidade de

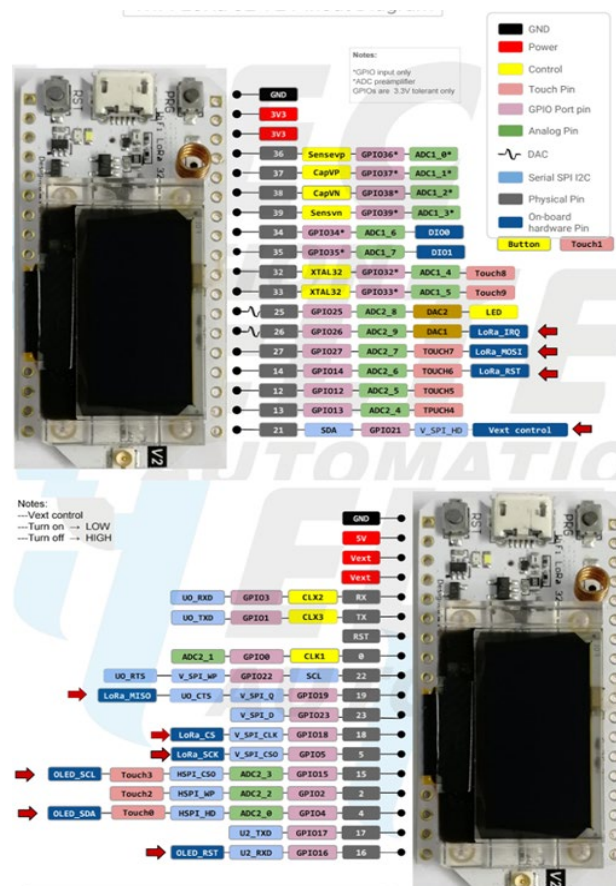
recepção, possibilitando a comunicação em ambientes com altos níveis de interferência, como por exemplo, as nascentes de rios (GERMANO, 2022).

A tecnologia LoRa é utilizada em redes LoRaWAN, que operam em bandas de frequência de 915 MHz nas Américas. Uma rede LoRaWAN é composta por três principais elementos:

- Servidor de rede: Funciona como o sistema central que processa os dados recebidos dos *gateways* e os distribui para aplicações finais.
- *Gateways*: São os dispositivos intermediários que recebem os dados de múltiplos dispositivos finais e os retransmitem para um servidor central via redes de alta capacidade, como o sinal de internet.
- Dispositivos finais: Todos os equipamentos com sensores ou atuadores que transmitem dados via LoRa.

A Figura 10 apresenta um modelo do módulo LoRa com suas portas e pinos detalhados, essenciais para a comunicação de longa distância e o baixo consumo de energia característicos dessa tecnologia.

Figura 10: Pinagem do módulo WiFi LoRa 32(V2)



Fonte: Fernando K Tecnologia (2023)

2.5.2.2 Pontos Positivos da Tecnologia LoRa

A LoRa permite diferentes níveis de segurança, incluindo criptografia de ponta a ponta, integridade de dados e autenticação dos dispositivos, tornando-a uma escolha robusta para muitas aplicações de IoT (SANTOS et al., 2020). Além da segurança fornecida, o conceito da utilização de LoRa vem crescendo cada vez mais, devido aos pontos positivos da sua utilização, sendo eles:

- **Alcance de Longa Distância:** Devido a capacidade de transmitir dados a distâncias que variam entre 5 a 20 km, dependendo do ambiente, a torna ideal para aplicações em áreas rurais ou remotas.
- **Baixo Consumo de Energia:** Os dispositivos LoRa operam por anos com uma única bateria, o que é crucial em dispositivos IoT que precisam de longa vida útil sem manutenção frequente, tendo em vista que os sensores geralmente estão em locais remotos sem infraestrutura elétrica e são alimentados por baterias ou painéis solares, então, podem funcionar por anos sem manutenção.
- **Alta Capacidade de Conexão:** A tecnologia LoRaWAN pode suportar inúmeros dispositivos conectados simultaneamente em uma mesma rede, sendo uma solução escalável para IoT em grande escala.
- **Custo Reduzido:** Devido a sua operação em bandas de frequência não licenciadas, a implementação de redes LoRa pode ser mais barata em comparação com as tecnologias convencionais de celulares.
- **Resistência à Interferência:** A modulação CSS (*Chirp Spread Spectrum*) permite que a LoRa funcione com êxito em ambientes com alta interferência, como em áreas urbanas densamente povoadas.
- **Monitoramento em Tempo Real:** A transmissão contínua de dados permite que os responsáveis possam monitorar a condição da água e do ambiente em tempo real, facilitando uma resposta rápida a qualquer anomalia, como poluição ou redução do fluxo de água.

2.5.2.3 Utilização da Tecnologia LoRa

A utilização da tecnologia LoRa vem crescendo constantemente, devido a sua variedade de aplicações, sendo assim, é amplamente usada em redes de IoT, permitindo a comunicação

entre dispositivos como sensores, atuadores e *gateways*, especialmente onde outras tecnologias como Wi-Fi ou 4G/5G não são viáveis devido ao consumo de energia ou à cobertura e alcance é limitado (SANTOS et al., 2020). Então, através de algumas pesquisas, encontra-se que algumas aplicações comuns para o módulo de LoRa são:

- Monitoramento de Cidades: Através do monitoramento de iluminação pública, sistemas de coleta de resíduos e gestão de tráfego.
- Agricultura de precisão: Utilização de sensores que monitoram umidade do solo, qualidade do ar e condições meteorológicas em fazendas remotas.
- Monitoramento ambiental: Monitoramento de parâmetros ambientais em áreas naturais, como rios, florestas e montanhas.
- Logística e rastreamento: Aplicação em dispositivos de rastreamento que monitoram a localização e condição de bens em trânsito.
- Gestão de energia: Realização do monitoramento de consumo e eficiência energética em sistemas industriais ou residenciais.
- Saúde: Utilização em sensores de saúde que transmitem dados de pacientes remotamente, especialmente em áreas com pouca conectividade.

A tecnologia LoRa tem um papel fundamental na expansão do ecossistema de IoT, possibilitando novas aplicações que antes eram limitadas por restrições de energia e distância. Ela oferece uma solução eficiente para a conectividade de dispositivos em áreas remotas ou em grande escala, sendo uma ferramenta valiosa para cidades inteligentes, agricultura, logística e monitoramento ambiental. Seus pontos fortes, como o baixo consumo de energia, alcance estendido e custo acessível, a tornam uma escolha atrativa para desenvolvedores e empresas que buscam conectar o mundo físico ao digital de maneira eficiente (GERMANO, 2022).

Sendo assim, dentre as inúmeras aplicações da tecnologia LoRa, ela pode ser utilizada no monitoramento de nascentes de rios, pois apresenta uma solução eficiente para o monitoramento do fluxo e nível da água nas nascentes dos rios, especialmente em áreas remotas e de difícil acesso, afim de evitar acidentes fatais envolvendo a água e mitigar os impactos causados por estes desastres naturais. Então, através da utilização de sensores, Arduino, *displays* e *buzzers* conectados a redes LoRa, é possível obter informações críticas sobre o fluxo, nível, qualidade e quantidade de água presente nas nascentes dos rios, contribuindo diretamente para prevenção de cabeças d'água e enchentes, que trazem riscos de vida para a população local e causam grandes danos materiais nas áreas em que ocorrem estes desastres naturais.

Então, para o monitoramento do fluxo e nível de água nas nascentes de rios, sensores interligados com os módulos LoRa são distribuídos ao longo das áreas de interesse, realizando a coleta dos dados que serão cruciais para análise dos fatores que envolvem acidentes naturais e que são necessários para a tomada de decisão, caso algum possível desastre natural seja detectado. Alguns parâmetros são essenciais para o êxito desta aplicação da tecnologia LoRa, sendo imprescindível a realização dos monitoramentos corretos, para obtenção das informações precisas sobre o nível e o fluxo de vazão da água, especialmente em regiões onde há fluxo de pessoas. Então, os dados serão coletados em intervalos programados e transmitidos via LoRa para estações base (*gateways*), que estão conectadas à internet, os *gateways* enviam os dados para uma plataforma de monitoramento centralizada, onde será possível acompanhar em tempo real o estado da nascente e avaliar as variações de comportamento dos dados coletados.

Desta maneira, através da LoRa, neste projeto será possível realizar o monitoramento do nível e fluxo da água nas nascentes dos rios, de maneira em que sejam coletados dados em tempo real do comportamento e variação do volume de água presentes nas nascentes e a velocidade de vazão e fluxo da água, afim de detectar possíveis acidentes naturais causados pelo aumento desses fatores. A Figura 11 exibe um módulo LoRa com antena e pinos de conexão, ideal para aplicações em monitoramento remoto e de longo alcance em ambientes de difícil acesso, como nascentes de rios.

Figura 11: Módulo LoRa com Antena e Pinos de Conexão para Monitoramento Remoto



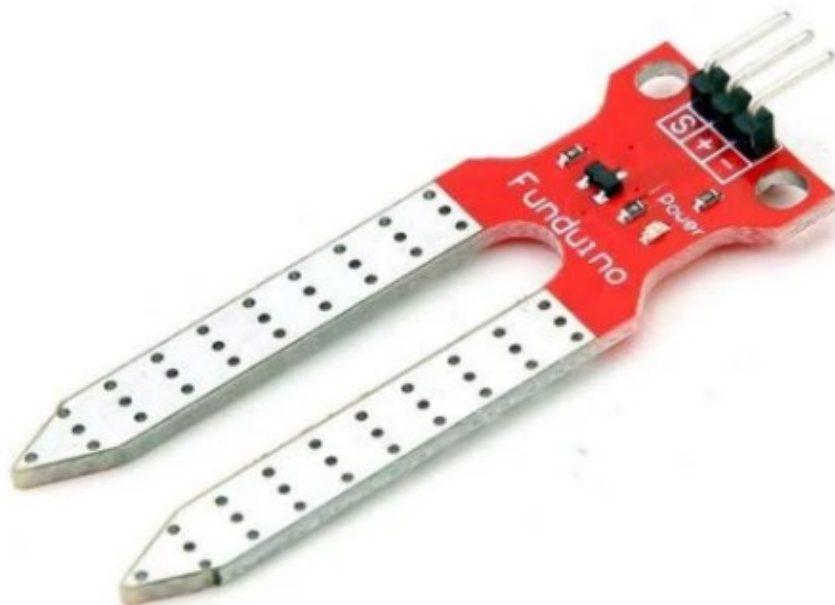
Fonte: DR. Eletronico (2024)

2.5.3 Sensores de Umidade do Solo: Surgimento e Funcionamento

Os sensores de umidade do solo surgiram a partir da necessidade de monitorar e medir o nível de umidade da terra de forma efetiva, trazendo um avanço para os setores agrícolas e pesquisas relacionadas ao meio ambiente. Sendo assim, este componente evoluiu ao longo do tempo, se adaptando as necessidades e, devido a isso, atualmente encontra-se sensores robustos e que possuem a capacidade de transmitir dados em tempo real.

De forma específica, os sensores de umidade do solo, atuam como detectores do nível de água presente nas camadas do solo através da resistividade da terra. A Figura 12 apresenta esse modelo sensor de umidade do solo, utilizado para medir a quantidade de água presente no solo por meio de sua resistividade.

Figura 12: Sensor de Umidade do Solo por Resistividade

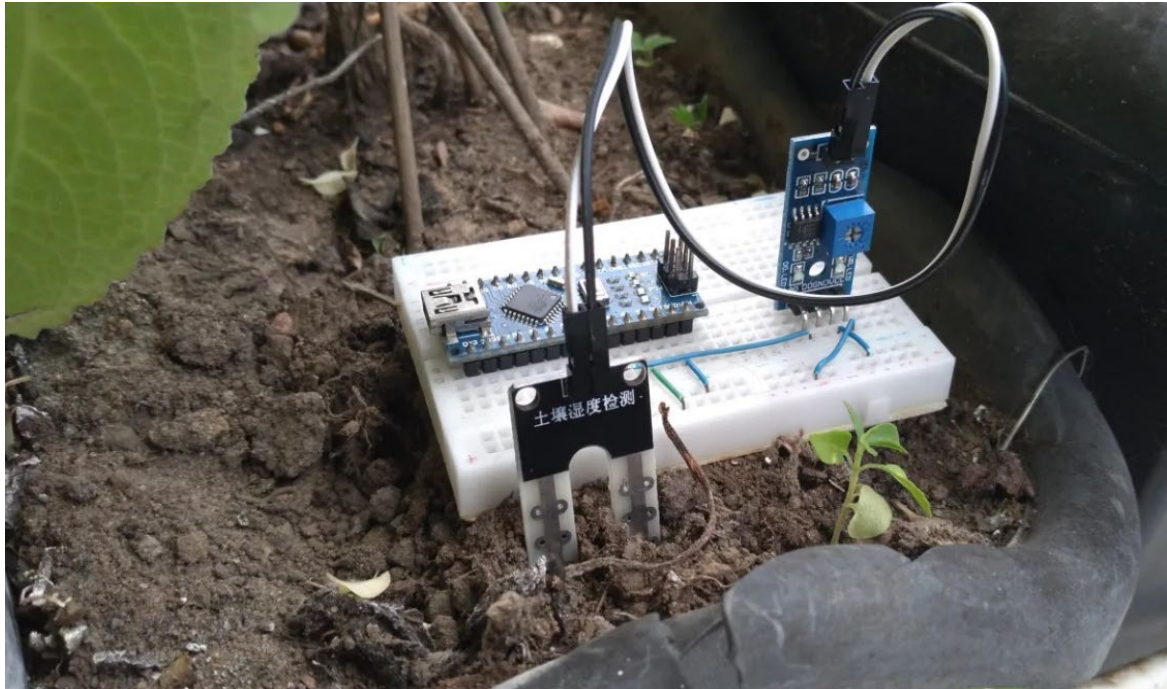


Sendo assim, realizam a medição das variações de umidade do solo, ou seja, quanto menor a resistividade, maior será nível de água presente no solo, levando em consideração os conceitos de condutividade e sabendo que a água é um condutor elétrico. Em contrapartida, quando o solo estiver seco a resistência elétrica será maior, indicando assim o baixo nível de água e por consequência, um valor baixo no nível de umidade da terra (SIGMA SENSORS, 2024). Após a realização da detecção e medição das informações relacionadas ao nível de umidade do solo, os dados coletados são enviados a um sistema de monitoramento, que receberá as informações em tempo real devido a sua precisão, permitindo a avaliação e tomada de decisão imediata.

Os sensores de umidade de solo são componentes duráveis e de fácil utilização, o que torna a sua aplicação cada vez mais recorrente em diversos segmentos. Apesar de possuírem um custo baixo, estes atuadores são capazes de resistir a variações climáticas e longos períodos de exposição ao tempo, além de possuírem tecnologias embarcadas que permitem conexões sem fio com algum dispositivo com microcontrolador.

Sendo assim, entre suas inúmeras aplicações, os sensores de umidade de solo podem ser utilizados para a medição da umidade do solo nas nascentes dos rios, desta maneira, auxiliando na identificação de possíveis variações no nível de umidade da terra. Após as informações serem coletadas, serão enviadas em tempo real para que seja realizada as devidas análises e tomadas de decisão, pois através destas informações, uma análise crítica pode ser feita baseada nesses parâmetros, que quando estiverem sofrendo variação, podem indicar um forte indício de uma tromba d'água ou enchente. A Figura 13 mostra um sensor de umidade do solo conectado a uma placa de prototipagem, permitindo a medição em tempo real do nível de umidade no solo. Sendo assim, os dados que são coletados são de suma importância para detecção de possíveis desastres naturais, que quando previstos com antecedência, reduzem os riscos a vida humana e mitigam as perdas materiais causadas por estes fenômenos naturais (MENDES; 2006).

Figura 13: Sistema de Monitoramento de Umidade do Solo em Tempo Real

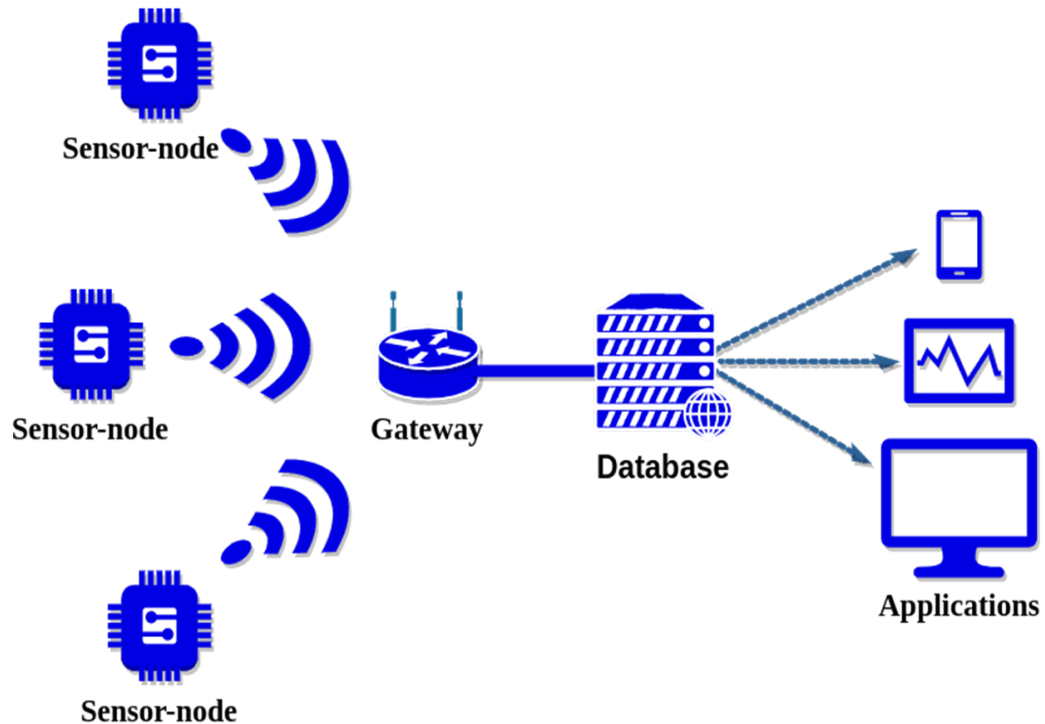


Fonte: Gustavo Cereza (2022)

2.6 REDES DE SENSORES SEM FIO (WSN) PARA MONITORAMENTO

As Redes de Sensores Sem Fio (WSN) têm se tornado uma tecnologia essencial em diversas aplicações, especialmente no monitoramento ambiental e na prevenção de desastres naturais. Estas redes consistem em um conjunto de sensores distribuídos que coletam e transmitem dados sem a necessidade de cabos físicos, permitindo uma comunicação eficiente e em tempo real (SOARES, 2015). No contexto de monitoramento de recursos hídricos, essas redes são particularmente relevantes, pois oferecem uma solução para a detecção precoce de fenômenos como inundações, deslizamentos de terra e contaminação da água, que podem ameaçar vidas humanas e causar danos significativos à infraestrutura. A Figura 14 demonstra a arquitetura de uma Rede de Sensores Sem Fio (WSN), composta por dispositivos sensores conectados a um gateway, que centraliza e transmite dados para um servidor. Essa configuração é essencial para monitoramento ambiental em tempo real, especialmente em contextos de prevenção de desastres naturais, onde a coleta e análise de dados são críticas para a tomada de decisões rápidas e eficientes.

Figura 14: Arquitetura de Rede de Sensores Sem Fio (WSN) para Monitoramento Ambiental em Tempo Real



Fonte: Embarcados (2020)

De acordo com Lucas Dias Anchesqui (2023), a utilização de WSN no desenvolvimento de sistemas de irrigação inteligente exemplifica a aplicabilidade dessa tecnologia em ambientes críticos. Esses sistemas não apenas otimizam o uso da água, mas também podem ser integrados a plataformas de alerta que informam os usuários sobre condições adversas, como aumento dos níveis de água em rios e reservatórios, contribuindo para a prevenção de desastres relacionados a inundações.

As WSNs são capazes de monitorar parâmetros ambientais, como temperatura, umidade, pressão e nível de água, através de sensores implantados em locais estratégicos. Essas informações são transmitidas para um servidor central que, por meio de algoritmos de processamento, analisa os dados e fornece previsões e alertas. Como um exemplo disso, Carvalho et al. (2012, p. 16) destacam que:

“A implantação de redes de sensores para controlar o consumo de água e energia elétrica, bem como para fins de monitoramento e segurança em estufas, é detalhado em [33], [34]. Trabalho semelhante foi desenvolvido em estufas para plantação de melão e repolho [35]. Um sistema para monitoramento de plantações de arroz foi projetado a partir de uma rede baseada no padrão ZigBee em [36] e [37]. Com o objetivo de monitorar o nível de água em uma plantação para evitar os efeitos da seca, detalha-se uma rede de sensores que foi testada em laboratório e em seguida foi implantada em uma plantação de chá [38].”

Silva e Fruett (2013) destacam que o monitoramento de variáveis ambientais tornou-se essencial frente às mudanças climáticas, permitindo ações preditivas e corretivas. Imagens de satélite, comumente usadas para análise de temperatura e movimento de massas de ar, não são suficientes para o monitoramento preciso de variáveis locais como a temperatura e a pressão atmosférica. Para tal finalidade, o uso de estações meteorológicas é necessário, fornecendo informações adicionais, como a presença de partículas e dados atmosféricos. Além disso, redes de sensores sem fio têm sido desenvolvidas como alternativa para superar os desafios logísticos, permitindo a coleta de dados sem a necessidade de cabos.

As redes de sensores sem fio (WSNs) podem ser classificadas em duas categorias principais: redes proativas e redes reativas, cada uma com diferentes modos de operação e requisitos energéticos. Nas redes proativas, os nós trocam periodicamente dados, o que as torna adequadas para monitoramento contínuo, enquanto nas redes reativas, os nós apenas reagem a mudanças no ambiente, sendo mais indicadas para aplicações que demandam resposta rápida a eventos. Além disso, as arquiteturas de roteamento podem ser planas ou hierárquicas, dependendo da necessidade de agregação de dados e da otimização do consumo de energia. Em ambas as estratégias, o uso de algoritmos localizados coordenados é fundamental para garantir escalabilidade, robustez e eficiência energética, embora esses algoritmos enfrentem desafios devido à sensibilidade a parâmetros e à necessidade de equilíbrio entre recursos e fidelidade de sensoriamento (BARROS, 2002).

Oliveira (2019, p. 18) ressalta a relevância das redes de sensores sem fio (WSNs) no contexto científico e social, afirmando:

"Em termos sociais a abordagem do assunto é relevante pois o uso de tecnologias de código aberto, facilmente reaplicáveis, enquadram-se plenamente como tecnologias sociais, oportunizando ao cidadão comum, a construção de dispositivos que podem favorecer a obtenção de conhecimentos relacionados à saúde e à qualidade de vida. De mesmo modo, tecnologias de baixo custo também podem ser bastante atrativas ao pequeno empreendedor, podendo apresentar soluções viáveis para implementações de melhorias em processos produtivos artesanais e/ou familiares, como o exemplo do agronegócio familiar. No contexto científico a discussão da temática é relevante pois o uso de projetos com tecnologias de código aberto pode viabilizar pesquisas a custos menores, através do favorecimento de procedimentos como coleta, armazenagem, tratamento e recuperação de dados através da utilização de sistemas que podem ser customizados às necessidades dos processos investigativos."

O monitoramento de rios desempenha um papel fundamental na sustentabilidade e no bem-estar das comunidades ribeirinhas, além de ser essencial para diversas aplicações, como a gestão de recursos hídricos e a prevenção de desastres naturais. Com o avanço das tecnologias, esse processo evoluiu significativamente, passando de métodos manuais para sistemas

automatizados que utilizam Redes de Sensores Sem Fio (RSSFs). Esses sistemas inovadores não apenas aumentam a precisão das medições, mas também ampliam a cobertura das áreas monitoradas, permitindo a coleta de dados em tempo real e a comunicação eficiente entre os sensores. Como ressalta Martins Júnior (2013), é importante compreender a terminologia utilizada nesse contexto, onde "mote" se refere ao hardware específico e "nodo" designa um ponto de interconexão na rede.

Um bom exemplo, com a proposta do tema, é o trabalho realizado por Souza et al. (2017) no qual os autores focaram no monitoramento das condições ambientais na bacia hidrográfica do rio Jacaraípe. A pesquisa destacou a importância das Redes de Sensores Sem Fio (RSSF) como ferramentas essenciais para coletar dados relevantes que podem ajudar a mitigar os impactos das enchentes na região. A rápida ocupação desordenada do território, que resultou em sérios problemas como o assoreamento do rio, reforça a necessidade de soluções de monitoramento contínuo. Embora as soluções definitivas exijam tempo e investimentos em infraestrutura, a utilização de RSSF pode fornecer uma alternativa viável e de baixo custo, possibilitando a coleta e análise de dados essenciais para a gestão dos recursos hídricos. A implementação de WSNs não é apenas uma questão técnica, mas também envolve uma abordagem holística que considera a colaboração entre diferentes setores, como governamentais, acadêmicos e a sociedade civil.

Em resumo, as Redes de Sensores Sem Fio oferecem uma solução viável e eficaz para o monitoramento de recursos hídricos e a prevenção de desastres naturais. A capacidade de coletar e analisar dados em tempo real, juntamente com a implementação de sistemas de alerta, pode salvar vidas e minimizar danos. À medida que a tecnologia avança, a integração de WSNs com outras plataformas digitais promete revolucionar a forma como monitoramos e gerenciamos os riscos associados a desastres naturais, garantindo uma resposta mais rápida e eficaz.

2.7 LINGUAGEM C E SUAS DERIVAÇÕES

A linguagem de programação C foi citada por Deitel em 1994, porém, foi desenvolvida em 1972 por Dennis Ritchie. Desde o seu desenvolvimento, a linguagem C foi criada para fazer parte da nova versão do sistema operacional Unix e por isso tornou-se popular, tendo como objetivo principal operar com o *hardware* de computadores, de maneira eficiente, flexível e de baixo nível, tendo a capacidade de desenvolver sistemas operacionais e outros *Softwares* mais complexos. Sendo considerada a base de outras linguagens de programação e de nível intermediário, a linguagem C permite ter o controle exato da funcionalidade de um programa,

sendo assim, é muito utilizada por programadores iniciantes, pois possui uma sintaxe simples e por ser a base de outros tipos de linguagem de programação (JUNIOR; MARTINS; VIRTUOSO, 2012).

2.7.1 Evolução

A partir da linguagem de programação C, surgiram inúmeras outras linguagens com o objetivo de expandir as capacidades de aplicação e adaptar-se a diferentes necessidades de utilização. Sendo assim, algumas linguagens de programação derivadas do C, são:

- C++: Criada em 1979 por Bjarne Stroustrup, a linguagem de programação C++ trata-se de uma extensão da linguagem C, que quando criada teve como objetivo principal adicionar recursos de programação orientada a objetos ao C, mantendo sua eficiência e flexibilidade. Ao passar dos anos, a linguagem de programação C++ evoluiu de forma considerável, com a adição bibliotecas padrão, suporte a templates, e outras funcionalidades avançadas. Portanto, com a norma ISO/IEC 27001 a evolução de suas capacidades e a garantia de relevância em aplicações modernas para sistemas operacionais e *Softwares* de altas performance permaneceu em constante expansão.

Seguindo estritamente a base C, a linguagem de programação C++ manteve a eficiência e a capacidade de baixo nível, porém, acrescentou suporte a classes, herança, encapsulamento e polimorfismo, permitindo o desenvolvimento de Software mais modular e reutilizável (JUNIOR; MARTINS; VIRTUOSO, 2012).

- C#: Desenvolvida pela Microsoft, lançada em 2000 como parte da plataforma .NET. Criada por Anders Hejlsberg, C# foi projetada para ser simples, moderna e orientada a objetos, com sintaxe inspirada em C++ e Java. A linguagem foi concebida para facilitar o desenvolvimento de aplicações robustas no ambiente Windows e se destacou pela sua integração com o .NET Framework. Ao longo dos anos, C# evoluiu com várias versões que introduziram recursos como LINQ, *async/await*, e expressões lambda, tornando-se uma das principais linguagens para desenvolvimento de aplicações empresariais, jogos com Unity, e sistemas em nuvem. Sua constante atualização, agora com o .NET Core e .NET 5/6/7, garante sua relevância no cenário tecnológico atual. Ela é usada principalmente no desenvolvimento de aplicações no ecossistema .NET, trazendo facilidades para o

desenvolvimento de aplicativos empresariais, jogos e outras aplicações complexas.

- Java (1995): De acordo com JAVA, embora não seja uma derivada direta de C, Java foi fortemente influenciada por C e C++. Java é uma linguagem de programação criada por James Gosling e sua equipe na Sun Microsystems, lançada em 1995. Inicialmente chamada de Oak, Java foi projetada para ser simples, portátil e segura, com a filosofia "escreva uma vez, execute em qualquer lugar" (WORA), graças à Máquina Virtual Java (JVM). Java rapidamente se tornou popular devido à sua aplicabilidade em diversos ambientes, desde aplicações *desktop* até servidores e dispositivos móveis, e corporativas (SILVA; SOBRAL, 2017).

2.7.2 Aplicações na Engenharia Elétrica

De acordo com António Manuel G.P. de Almeida (2023), entende-se que na engenharia elétrica, C e suas derivações desempenham um papel crucial em várias aplicações devido à sua eficiência e capacidade de operar próximo ao *hardware*. Alguns exemplos incluem:

- Desenvolvimento de Sistemas Embarcados: C é amplamente utilizada em sistemas embarcados, onde a eficiência e o controle de *hardware* são críticos. Microcontroladores e processadores em sistemas de controle, automação e dispositivos IoT frequentemente são programados em C para maximizar o desempenho e minimizar o uso de recursos.
- Simulação e Modelagem: Linguagens como C e C++ são usadas em simulações de circuitos elétricos e modelagem de sistemas, devido à sua capacidade de realizar cálculos complexos de maneira rápida e eficiente. Ferramentas como o SPICE, usada para simulação de circuitos eletrônicos, são escritas em C.
- Automação e Controle: Em sistemas de controle industrial e automação, onde a confiabilidade e a resposta em tempo real são cruciais, C e suas variantes são frequentemente utilizadas para desenvolver Software que interage diretamente com *hardware*, sensores e atuadores.
- Desenvolvimento de *Software* para Eletrônica de Potência: Em aplicações de eletrônica de potência, como inversores e conversores, C é usada para programar

o *firmware* que controla os dispositivos de comutação, garantindo a eficiência e estabilidade dos sistemas.

A linguagem de programação C tem sido fundamental no desenvolvimento de Software para diversas áreas, inclusive para engenharia elétrica, graças à sua eficiência e flexibilidade. Suas derivações, como C++ e Objective-C, continuaram a expandir suas capacidades, permitindo que os engenheiros desenvolvessem sistemas mais complexos e robustos. A evolução da linguagem C reflete sua importância contínua na engenharia e em muitas outras áreas tecnológicas (ALMEIDA, 2023).

2.8 FUNDAMENTOS SOBRE INTERNET DAS COISAS (IOT)

A Internet das Coisas (IoT, do inglês Internet of Things) refere-se à interconexão de dispositivos físicos à internet, permitindo que se comuniquem e compartilhem dados entre si de maneira autônoma. A IoT tem se destacado como uma das principais inovações tecnológicas, com o potencial de transformar diferentes setores ao integrar objetos do cotidiano em uma rede digital inteligente (SANTOS et al.). O Conceito de IoT pode ser descrito como uma evolução natural da internet, que não mais conecta apenas computadores e smartphones, mas uma ampla variedade de dispositivos, desde sensores em instalações industriais até sistemas de monitoramento em cidades inteligentes. A Figura 15 ilustra a conectividade entre dispositivos em uma rede de Internet das Coisas (IoT), demonstrando como sensores, atuadores e dispositivos inteligentes podem se comunicar e compartilhar dados de forma autônoma, integrando-se em uma rede digital inteligente.

Figura 15: Conectividade e Integração de Dispositivos em uma Rede de Internet das Coisas (IoT)



Fonte: Engenharia Híbrida (2022)

Segundo Neves (2021, p.19), “IoT pode, por exemplo, viabilizar o monitoramento em tempo real dos movimentos na cidade, o que permite fundamentar de maneira mais concreta o desenvolvimento de políticas públicas, com base em maior quantidade de dados.”. A IOT contínua facilita o processamento de grandes volumes de dados em tempo real, o que é fundamental para aplicações que necessitam de respostas imediatas, como monitoramento de desastres naturais e automação industrial.

A IoT também promove uma transformação significativa na forma como os negócios e os serviços públicos são operados. A integração de sensores e atuadores conectados proporciona um monitoramento constante e permite ações preventivas em diversas áreas, como segurança, saúde e gestão de recursos hídricos (GALEGALE et al., 2016) . Por exemplo, a de IoT em serviços de prevenção de enchentes permite o monitoramento do nível de rios e da intensidade das chuvas, reduzindo o risco de inundações em áreas urbanas (LEAL, 2019) .

A infraestrutura da IoT envolve sensores, redes de comunicação e plataformas de processamento de dados. Os sensores capturam informações do ambiente, como temperatura, umidade e pressão, enquanto as redes de comunicação (como LoRa, Wi-Fi ou 4G) transmitem esses dados para plataformas de análise. A partir daí, sistemas automatizados processam as

informações, gerando insights para a tomada de decisão (GOMES e CAMEIRA, 2018) . Esse ciclo contínuo de coleta, tra processamento de dados em tempo real possibilita que empresas e governos implementem estratégias de resposta mais eficazes e imediatas.

Em termos de impacto econômico e social, a IoT é vista como uma tecnologia com potencial para revolucionar a infraestrutura de cidades inteligentes e promover avanços na sustentabilidade. Um estudo de Oliveira e Conti (2019) enfatiza que a IoT pode contribuir para a mitigação de desastres ambientais, atuando como uma ferramenta preventiva em áreas de risco e auxiliando na gestão de emergências, principalmente em contextos urbanos .

A aplicação de IoT em sistemas de monitoramental, especialmente para o gerenciamento de recursos hídricos e prevenção de desastres, tem se mostrado promissora. A integração de sensores e módulos de comunicação como LoRa permite que áreas remotas sejam monitoradas com eficiência, mesmo em condições de infraestrutura limitada. Júnior et al. (2018) abordam como a IoT pode ser utilizada no monitoramento de enchentes, onde a coleta de dados em tempo real é crucial para prever e minimizar impactos, salvando vidas e reduzindo perdas econômicas

Portanto, a Internet das Coisas não apenas representação tecnológica, mas também oferece uma base para inovações em setores estratégicos, contribuindo para a criação de ambientes mais seguros, sustentáveis e conectados.

3 MATERIAIS E MÉTODOS

3.1 MATERIAIS E MÉTODOS DO PROTÓTIPO VIRTUAL

Para a realização do projeto, utilizamos uma série de materiais específicos e métodos detalhados para garantir a eficácia e precisão dos dados coletados. Um Arduino Uno R3, que é uma placa microcontroladora baseada no microcontrolador ATmega328P, é o núcleo do sistema, controlando e gerenciando todos os sensores e atuadores conectados. Um display LCD 16x2, que é um display de cristal líquido com duas linhas de 16 caracteres cada, exibe informações sobre os dados coletados pelos sensores, como o nível da água e a umidade do solo.

Para ajustar o contraste do display LCD, utilizamos um potenciômetro e para a proteção do LCD contra correntes excessivas, empregamos um resistor. Um piezo, que é um dispositivo que emite som quando acionado, funciona como uma sirene de alerta para indicar níveis críticos de água ou falhas no sistema.

O sensor ultrassônico (Ultrasonic Distance Sensor - 4 pinos) utiliza ondas ultrassônicas para medir distâncias, fornecendo dados para a medição do nível da água na nascente, o que é

crucial para a prevenção de enchentes ou secas. O sensor de umidade do solo mede o nível de umidade no solo, monitorando as condições ambientais nas nascentes.

Um motor CC com codificador (313 Motor CC com codificador), que possui um sensor de efeito Hall integrado, é utilizado para simular as pás de uma turbina, gerando energia limpa para o sistema. Para a proteção do motor contra correntes excessivas, utilizamos um resistor.

Uma fonte de energia fornece energia para todos os componentes do circuito. Para monitorar a vazão do rio, utilizamos um sensor de fluxo de água, que fornece dados sobre o fluxo de água na nascente. O módulo de comunicação LoRa, que é um transceptor para comunicação de longa distância utilizando a tecnologia LoRa, permite a transmissão dos dados coletados para um servidor remoto. Para facilitar a comunicação de longa distância, utilizamos uma antena específica para o módulo LoRa.

Os métodos utilizados no projeto envolvem a montagem do circuito no Tinkercad, uma plataforma que permite a simulação eletrônica do circuito antes da montagem física, validando o funcionamento dos componentes e a lógica do programa desenvolvido para o Arduino. A programação do Arduino é desenvolvida em C/C++ utilizando a IDE do Arduino, controlando a leitura dos sensores, a exibição dos dados no LCD, o acionamento do piezo e a comunicação com o módulo LoRa.

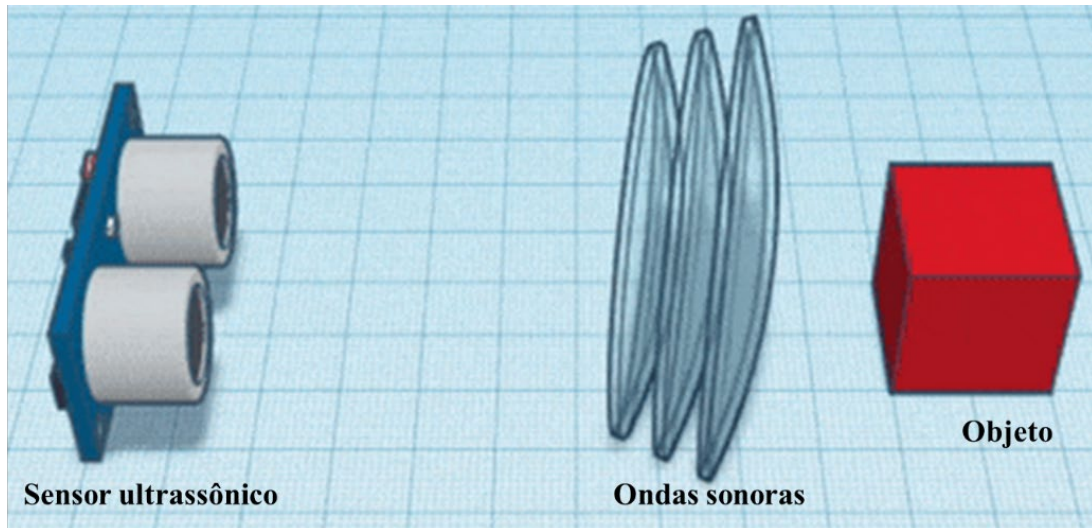
Realizamos testes virtuais no Tinkercad para garantir a funcionalidade do sistema, ajustando o código e a configuração dos componentes conforme necessário. O código utilizado para esses testes encontra-se documentado no Apêndice A. Após a validação virtual, o sistema pode ser montado fisicamente e instalado nas nascentes próximas à região, onde será coletado dados reais sobre o nível da água, a umidade do solo e o fluxo do rio para análise.

3.1.1 Sensor ultrassônico de distância HC - SR04

A utilização do sensor ultrassônico de distância neste projeto, deriva-se da necessidade de medição do nível de água na nascente de um rio. O funcionamento deste componente nos permite realizar medições de 2cm a 400cm com precisão de medição de até 3mm e inicia-se com emissão de um sinal sonoro agudo de 40kHz que é imperceptível a audição humana. Então, com a emissão da onda sonora, o som desloca-se no ar até atingir um determinado corpo, e, após isso, o sensor ultrassônico de distância detecta o primeiro sinal do eco, assim, calculando o tempo necessário para que o som reflita no alvo e retorne, o que resultará na distância em que o objeto se encontra. A Figura 16 demonstra o funcionamento de um sensor ultrassônico,

ilustrando a emissão de ondas sonoras que são refletidas por um objeto e retornam ao sensor, permitindo a medição da distância com precisão.

Figura 16: Funcionamento do Sensor Ultrassônico para Medição de Distância



Fonte: TinkerCad (2024)

A medição da distância é definida através da seguinte equação:

$$\text{Distância de teste} = \frac{(\text{Tempo de nível alto} \times \text{Velocidade do som})}{2}$$

Para o funcionamento do sensor ultrassônico garantir a precisão durante a medição das distâncias, é necessário realizar as ligações de forma correta dos 4 pinos base. Então, foram conectados o pino de alimentação (VCC) ao terminal de 5V do arduino, o aterramento (GND) foi conectado ao GND do arduino. Os pinos acionadores ECHO E TRIG foram conectados aos pinos digitais 9 e 10 do arduino que controlam o envio e recebimento de pulsos, ou seja, realizam a medição da distância do nível da água até o sensor em tempo real. A Figura 17 apresenta o sensor ultrassônico HC-SR04, utilizado para medir distâncias por meio da emissão e recepção de ondas sonoras.

Figura 17: Sensor Ultrassônico HC-SR04 para Medição de Distância



Fonte: ElecFreaks (2022)

A função `NivelRio()`, utilizada na configuração do sensor ultrassônico, calcula a distância existente entre o nível do rio e o sensor, de maneira em que o pino TRIG realize a leitura do tempo que o pulso leva para retornar até o pino ECHO, convertendo o tempo gasto em uma distância e atribuindo uma classificação baseada no valor que foi encontrado, ou seja, o sensor ultrassônico irá classificar as informações de acordo com sua medição. Através da função de risco, é realizada uma soma ponderada para definir as classificações de risco, desta forma, será indicando o nível estável de água quando o valor encontrado estiver dentro dos parâmetros estipulados e quando os valores estiverem acima, será interpretado como inundação.

Após as medições e classificação do nível da água, as informações serão coletadas pelo Arduino e transmitidas ao visor LCD que irá exibir em tempo real a mensagem referente ao nível de risco.

Os parâmetros técnicos deste componente são:

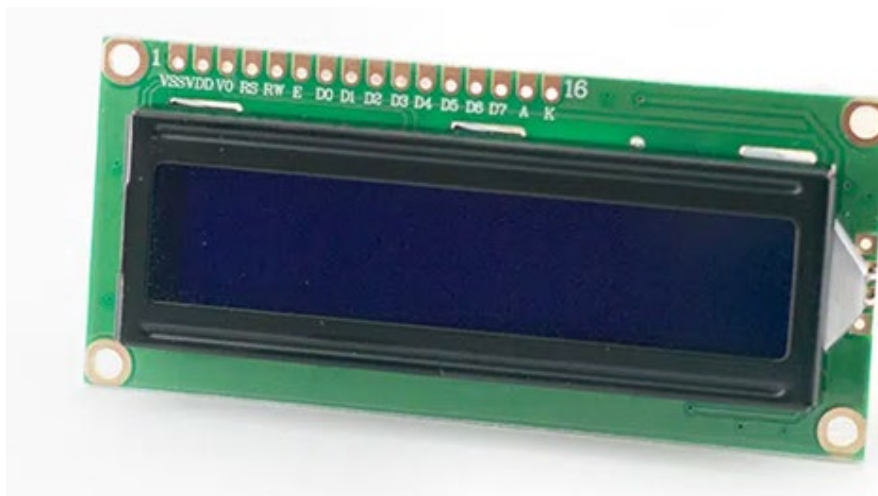
- Tensão de Trabalho: DC 5 V
- Corrente de Trabalho: 15 mA
- Frequência de Trabalho: 40 Hz
- Alcance Máximo: 4 m
- Alcance Mínimo: 2 cm
- Ângulo de Medição: 15 graus
- Sinal de Entrada de Disparo: Pulso TTL de 10 μ s
- Sinal de Saída do Eco: Sinal de nível TTL de entrada proporcional ao alcance
- Dimensões: 45 \times 20 \times 15 mm

3.1.2 Display LCD 16x2

O display LCD 16X2 é um dispositivo visualizador de cristal líquido, que tem a capacidade de exibir duas linhas de texto com 16 caracteres cada. Este dispositivo possui uma luz de fundo LED e um "aglomerado" de vidro polarizado em torno de cristais líquidos, que fazem girar eletronicamente a luz polarizada, assim, permitindo que a luz de fundo passe e seja vista ou bloqueada pelo polarizador na parte superior.

Neste projeto, o display LCD possui a finalidade de transmitir uma mensagem em forma de texto baseada nas informações detectadas pela função Risco (), permitindo a visualização do status atual do nível da água presente na nascente e em caso de risco, uma mensagem de alerta que indicará uma enchente. Figura 18 mostra um display LCD de 16x2, onde no projeto, ele é utilizado para apresentar mensagens sobre o nível da água detectado, alertando para situações de risco de enchente quando necessário.

Figura 18: Display LCD 16x2 para Monitoramento de Nível de Água



Fonte: Maker Hero (2022)

O display LCD deste projeto está interligado ao Arduino e utiliza alguns pinos digitais para transmitir as informações. Os pinos de controle e de dados do display LCD estão conectados aos pinos digitais 12, 11, 5, 4, 3 e 6 do Arduino, que são definidos no código de programação como os pinos responsáveis pela comunicação com o display. Entretanto, a alimentação do display LCD é feita pelos pinos de alimentação 5V (VCC) e o aterramento (GND) do Arduino, enquanto um potenciômetro está ligado ao pino de contraste (VO) do LCD, permitindo o ajuste da clareza da imagem transmitida no display. A Figura 19 apresenta o esquema de conexões do display LCD 16x2 HD44780 com o Arduino, detalhando os pinos de

controle, dados e alimentação. O funcionamento do LCD neste circuito é baseado nos comandos enviados pelo microcontrolador Arduino, que irão definir o que deverá ser exibido de acordo com os valores capturados pelos sensores presentes no circuito. Sendo assim, o Arduino mede continuamente o nível de água e a vazão do rio por meio de sensores ultrassônico de distância e sensor de umidade, que com base nesses dados, calcula o nível de risco de enchente.

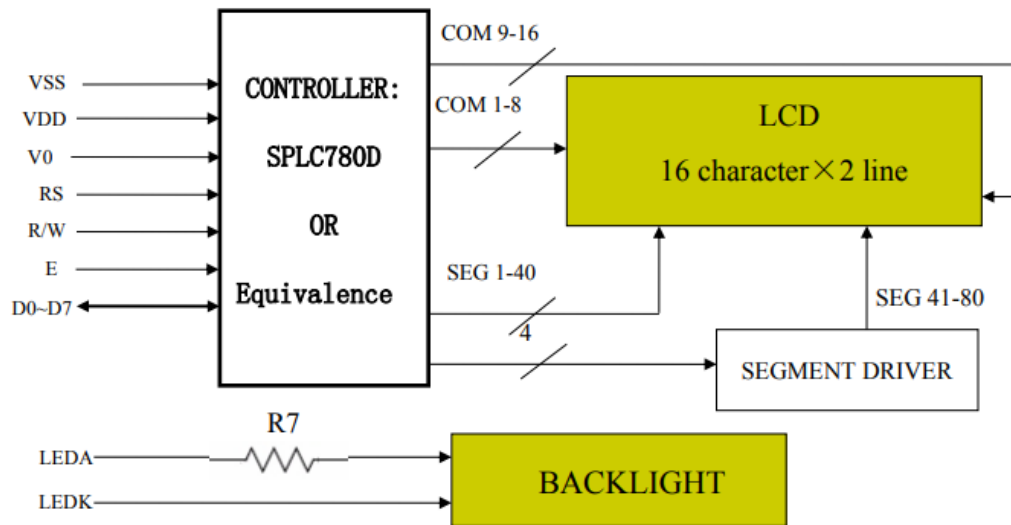
Figura 19: Esquema de Conexões do Display LCD 16x2 HD44780 com Arduino

Conexões LCD 16x2 - HD44780		
Pino LCD	Função	Ligação
1	Vss	GND
2	Vdd	Vcc 5V
3	V0	Pino central potenciômetro
4	RS	Pino 12 Arduino
5	RW	GND
6	E	Pino 11 Arduino
7	D0	Não conectado
8	D1	Não conectado
9	D2	Não conectado
10	D3	Não conectado
11	D4	Pino 5 Arduino
12	D5	Pino 4 Arduino
13	D6	Pino 3 Arduino
14	D7	Pino 2 Arduino
15	A	Vcc 5V
16	K	GND

Fonte: Maker Hero (2022)

Após cada medição e a recepção das informações, o Arduino envia as informações de status e os resultados das análises ao display LCD, que exibe o texto correspondente, ajudando o usuário a monitorar a situação em tempo real. O LCD é acionado pelo Arduino através da biblioteca LiquidCrystal que auxilia no controle de displays de cristal líquido, permitindo a comunicação em modos de 4 ou 8 bits e utilizando os pinos específicos no Arduino para controlar a tela LCD, assim, facilitando o envio de comandos e dados para o display. A Figura 20 ilustra o diagrama funcional do display LCD 16x2, destacando a interface entre o controlador, o segmento de exibição e a luz de fundo.

Figura 20: Diagrama Funcional do Display LCD 16x2 com Controlador e Backlight



Fonte: JHD LCD (2024)

No código de programação, algumas funções específicas controlam as posições onde cada dado será exibido, ajustando a linha e a coluna para que cada medição seja apresentada de forma organizada e clara.

Entretanto, a cada nova leitura dos sensores, o display LCD atualiza os dados em tempo real, permitindo que o usuário acompanhe o comportamento das variáveis do nível e fluxo da água.

3.1.3 Potenciômetro

A utilização do potenciômetro neste circuito, é explicada por sua capacidade de dividir a tensão, possibilitando realizar o ajuste do nível de sensibilidade do sistema de exibição do display LCD em relação à leitura das informações recebidas dos sensores ultrassônico e de umidade conectados ao Arduino.

Neste projeto, o potenciômetro está interligado a alimentação +5V (VCC), ao aterramento (GND) e o pino central ao pino 3 (V0) do display LCD, desta forma, nos permite ajustar o contraste da tela. O ajuste do potenciômetro possibilita regular a quantidade de corrente que passa para o display, influenciando diretamente a visibilidade do texto e quando é realizado algum movimento no eixo do potenciômetro, a resistência do circuito sofre variação, o que modifica a tensão que vai para o pino de contraste do LCD. Desta forma, interferindo diretamente na visibilidade dos caracteres que são exibidos na tela do display LCD, fazendo

com que as mensagens com informações relacionadas ao nível e fluxo da água na nascente do rio fiquem mais claras ou mais escuras, melhorando a nitidez e facilitando a leitura em diferentes condições de iluminação e visualização. A Figura 21 apresenta um potenciômetro utilizado para ajustar o contraste do display LCD no projeto, regulando a visibilidade das informações exibidas sobre o nível e fluxo da água.

Figura 21: Potenciômetro para Ajuste de Contraste do Display LCD

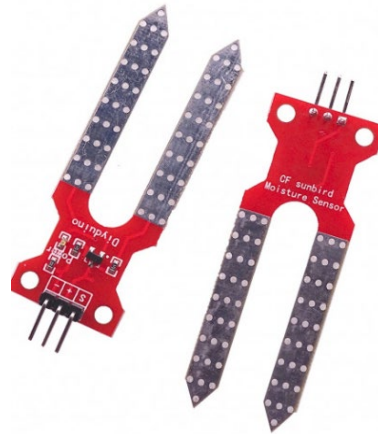


Fonte: Mercado Livre (2024)

3.1.4 Sensor de umidade do solo GC-58

Neste circuito, o sensor de umidade do solo atua como um indicador da intensidade da chuva e é utilizado para simular o funcionamento deste componente eletrônico quando exposto à chuva, apenas para fins didáticos, tendo em vista que se trata de uma simulação virtual. A Figura 22 mostra o sensor de umidade do solo GC-58, utilizado neste projeto para simular a intensidade da chuva em um ambiente virtual.

Figura 22: Sensor de Umidade do Solo GC-58 para Simulação de Intensidade de Chuva



Fonte: Mult Comercial (2024)

Na simulação virtual deste projeto, o sensor de umidade está interligado ao microcontrolador Arduino através dos pinos A0 e A1, onde A0 fornece energia ao sensor, e A1 lê os dados coletados referente a umidade. Fisicamente, esse sensor tem como função medir a umidade do solo, entretanto, neste projeto virtual ele simula a medição dos níveis chuva, para detectar variações de precipitação e possíveis enchentes. Para seu funcionamento, o sensor é alimentado momentaneamente no pino A0, realiza uma leitura de intensidade da chuva no pino A1, e, em seguida, a energia é desligada para evitar corrosão dos terminais. Essa leitura realizada é processada no código, onde valores analógicos correspondem a diferentes descrições de intensidade, como "sem chuva", "chuvisco" ou "tempestade". O código de programação, então, traduz essas medições para exibi-las em forma de mensagem de texto no display LCD e também como variáveis usadas no cálculo da função Risco() de enchente.

O código de programação utilizado, define as funções e as variáveis necessárias para realizar essa leitura e interpretar os dados. Além de ler o valor analógico do sensor, o código converte o valor em uma escala de intensidade de chuva, que contribui para determinar o nível de risco de alagamento, juntamente com as medições de nível e vazão do rio. Para ativação do sensor, o comando `digitalWrite(A0, HIGH)` é utilizado para ativar o pino A0, que fornece energia ao sensor de umidade do solo, em seguida, o código aplica o comando `delay(10)` de 10 ms, garantindo que o sensor tenha tempo para estabilizar antes da leitura.

Logo após, o comando `intensidadeChuva = analogRead(A1)` é utilizado para ler o valor analógico do sensor de umidade conectado ao pino A1, que indica a quantidade de umidade presente, ou seja, a intensidade da chuva. Para desativarmos o sensor, utilizamos o comando `digitalWrite(A0, LOW)` que é responsável por desligar o pino de alimentação A0.

3.1.5 Fonte de Alimentação DC 5A 4 Dígitos Duplo HF-3205S

A fonte de energia deste circuito, fornece a tensão e a corrente elétrica necessárias para o funcionamento de todos os componentes conectados. Com a configuração da fonte para uma tensão de 12V e uma corrente de 3,20A, esta fonte de alimentação está interligada a uma série de dispositivos, incluindo o módulo microcontrolador Arduino Uno, os sensores ultrassônico e de umidade e a um display LCD, garantindo o funcionamento sem sobrecarregar nenhum componente.

Com o ajuste correto dos valores de tensão e corrente, a fonte de energia funciona adequadamente para cada um dos elementos do circuito, garantido que todos operem dentro dos limites de segurança e eficiência. Neste circuito, a fonte não fornece energia somente ao Arduino Uno, mas também distribui a energia necessária para o display, os sensores e outros componentes, garantindo que o circuito funcione em conjunto para monitorar o ambiente e detectar possíveis enchentes, acionando alarmes e exibindo alertas quando alguma variação nos níveis e fluxo de água forem detectados.

A Figura 23 apresenta uma fonte de alimentação ajustável, configurada para fornecer 12V e 3,20A, garantindo energia estável para todos os componentes do circuito, incluindo o Arduino, sensores e display LCD.

Figura 23: Fonte de Alimentação Ajustável para Componentes do Circuito



Fonte: Loja do Mecânico (2024)

Especificações técnicas do componente:

- Saída Variável: Simples
- Tensão / Corrente Variável: 0 ~ 32V / 0 ~ 5A
- Proteção de Sobrecarga: SIM
- Proteção de Inversão de Polaridade: SIM
- Display: 4 Dígitos Duplo
- Precisão Básica do Display: 1,0%
- Regulação de Carga em Tensão: $\leq(0,01\% + 2\text{mV})$
- Regulação de Linha e Tensão: $\leq(0,01\% + 3\text{mV})$
- Regulação de Carga em Corrente: $\leq(0,1\% + 10\text{mA})$
- Regulação de Linha em Corrente: $\leq(0,1\% + 3\text{mA})$
- Ripple & Ruído (Tensão / Corrente): 2mV / 3mA RMS
- Dimensões: 156 x 110 x 260mm
- Peso: 4,8kg

3.1.6 Arduino UNO R3

O principal componente deste projeto é o microcontrolador Arduino Uno R3, que cumpre o papel de controlador central do circuito, que coordena a coleta de dados e a execução das ações dos componentes eletrônicos conectados a ele. Sua presença no circuito é crucial para o funcionamento do projeto, pois é responsável por processar as informações recebidas dos sensores e controlar as saídas de acordo com o que foi programado para detectar e responder a sinais de possível enchente.

O funcionamento deste microcontrolador é essencial para chegarmos ao objetivo final do projeto, primeiramente ele recebe dados do sensor de umidade do solo, que mede a presença de água, e do sensor ultrassônico, que monitora a distância de possíveis superfícies de água, ajudando a avaliar o nível de enchente. Após a coleta de todas as informações, os sensores mencionados enviam os sinais ao Arduino, que terá a função de interpretar os valores de entrada e decidir qual a melhor ação a ser tomada com base nos parâmetros configurados no código de programação.

Além dos sensores, o Arduino também se conecta através de seus terminais, a um display LCD, que recebe as informações do microcontrolador sobre os níveis de água detectados e as exibe em tempo real. Esse display LCD, interligado ao Arduino, permite que os usuários monitorem as leituras de forma visual e rápida. O microcontrolador Arduino também está

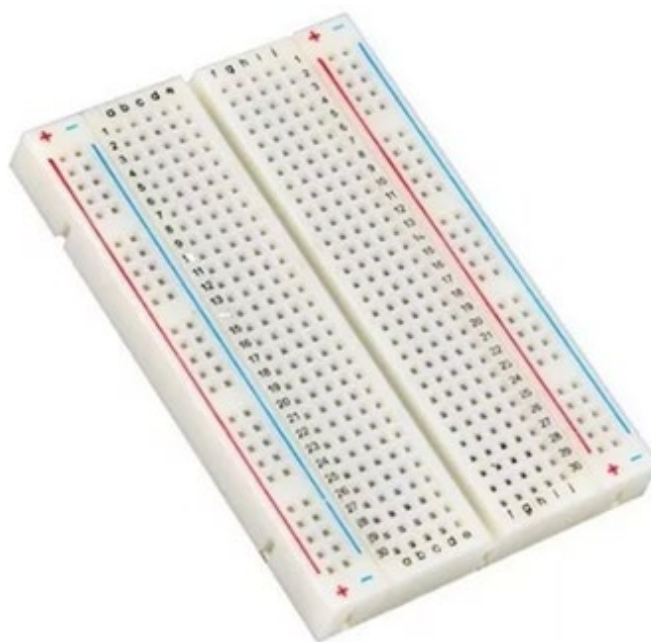
conectado a um piezo ou buzzer, que emite um alerta sonoro em caso de risco de enchente, e a outros dispositivos de controle, como LEDs, que podem indicar visualmente diferentes níveis de alerta.

Entretanto, além da parte física dos componentes, para que o Arduino funcione é necessária uma fonte de energia, e o microcontrolador deve conter um programa previamente gravado em sua memória. O código de programação é feito no software Tinkercad utilizando a linguagem de programação C++, e através disso definimos as instruções para cada situação possível, permitindo ao Arduino atuar de forma inteligente, decidindo quando deverá acionar o alarme, atualizar o display LCD, ou manter o sistema em estado de alerta conforme os valores coletados pelos sensores presentes no sistema.

3.1.7 Placa de ensaio protoboard

Conhecidas por sua utilização em prototipagem de circuitos, as placas de ensaio permitem conectar componentes com várias portas, através das partes condutivas da placa de ensaio que permitem o fluxo de elétrons entre os componentes eletrônicos conectados a ela. Composta por duas linhas em cada lado, a protoboard possui conectores longos que servem para conexões de energia e aterramento, usadas com muita frequência. A Figura 24 exibe uma placa de ensaio (protoboard) usada para prototipagem de circuitos.

Figura 24: Protoboard para Prototipagem de Circuitos



Neste projeto, a placa de ensaio tem como função interligar os diversos componentes eletrônicos sem a necessidade de solda. Sendo assim, o Arduino Uno, sensor ultrassônico, display LCD, buzzer e o sensor de umidade do solo estão conectados a ela, e através dela, os cabos de energia e sinais dos diferentes dispositivos serão conectados de forma organizada, garantindo que a energia distribuída pela fonte e pelo Arduino seja compartilhada entre todos os componentes. Desta maneira, facilitando o fluxo de dados entre todos os componentes que estão interligados através do protoboard, permitindo que o sistema opere de forma integrada, com cada sensor e atuador respondendo de acordo com as leituras e as instruções fornecidas pelo Arduino, criando um circuito funcional para monitoramento de enchentes.

3.2 MATERIAIS E MÉTODOS DO PROTÓTIPO FÍSICO

Para o contexto do projeto, vale destacar que o desenvolvimento deste protótipo prático foi realizado com uma configuração e componentes específicos, destinados a implementar o módulo LoRa e atingir os objetivos de monitoramento remoto. No entanto, alguns elementos presentes no protótipo real podem diferir em relação ao ambiente simulado ou protótipo virtual. Essas diferenças incluem tanto adaptações para simular as condições reais de um ambiente fluvial quanto ajustes técnicos para otimizar a integração dos componentes físicos.

Neste projeto físico, o Arduino Uno R3 SMD foi utilizado como microcontrolador principal, responsável pelo processamento das leituras de sensores e pela integração do sistema de monitoramento de nível e fluxo de água em um ambiente simulado. O uso do Arduino como microcontrolador foi fundamental para este projeto devido à sua versatilidade e compatibilidade com uma ampla gama de sensores, possibilitando a coleta de dados de forma precisa e em tempo real. A programação do Arduino permitiu a captura e o envio de dados de forma eficiente, integrando-se perfeitamente aos módulos de comunicação LoRa para transmissão dos dados coletados.

Para simular o fluxo constante de um rio, foi utilizada uma mangueira que alimentava o sensor de fluxo com um fluxo controlado de água, replicando a dinâmica de um ambiente natural. Essa configuração permitiu ajustes finos e validação da precisão dos sensores em condições controladas, garantindo que as leituras de nível e fluxo fossem confiáveis e representativas de um sistema real.

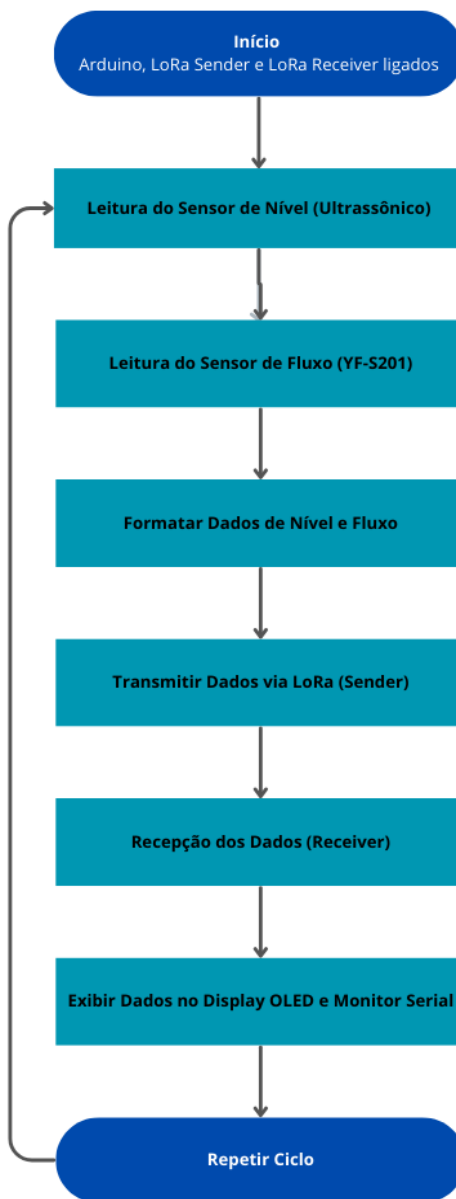
Cada componente foi previamente testado com um multímetro digital, para garantir a tensão e corrente corretas de operação, evitando possíveis falhas de leitura e danos ao circuito. A plataforma Arduino IDE foi empregada para a programação, utilizando bibliotecas

específicas para cada sensor e para a comunicação com os módulos LoRa, integrando toda a operação de forma eficiente.

A escolha do Arduino Uno R3 SMD e dos sensores se justifica pela versatilidade, precisão e compatibilidade com os requisitos do estudo, pois essas características permitem uma adaptação prática e precisa para um sistema de monitoramento de nível e fluxo de água em condições simuladas. Esta seleção de materiais é fundamental para garantir a precisão dos dados e a robustez necessária para possíveis aplicações futuras em ambientes externos. Os códigos estão disponíveis no Apêndice B, Apêndice C e Apêndice D.

A seguir, a Figura 25 ilustra um fluxograma de funcionamento do sistema de monitoramento de nível e fluxo de água é representado de forma sequencial e cíclica. O processo inicia com o Arduino e os módulos LoRa Sender e Receiver ativados. Em seguida, o Arduino realiza a Leitura do Sensor de Nível Ultrassônico JSN-SR04T/AJ-SR04M, calculando a altura da água e a porcentagem de enchimento. A leitura do Sensor de Fluxo YF-S201 contabiliza os pulsos para calcular a vazão em litros por minuto.

Figura 25: Fluxograma do Sistema de Monitoramento de Nível e Fluxo de Água com Arduino e LoRa



Fonte: Autoria própria (2024)

Os dados de altura, enchimento e fluxo são então formatados e enviados pelo Módulo LoRa Sender ao Receiver, que os recebe, interpreta e exibe no Display OLED e no Monitor Serial para acompanhamento em tempo real. Após um breve intervalo, o ciclo retorna à leitura inicial dos sensores, atualizando os dados periodicamente. Esse processo contínuo permite um monitoramento remoto e em tempo real do nível e fluxo de água.

3.2.1 Módulo Sensor Ultrassônico Impermeável JSN-SR04T/AJ-SR04M

O sensor JSN-SR04T/AJ-SR04M foi escolhido devido à sua capacidade de operar em ambientes úmidos, essencial para simulações de medições de nível de água em um ambiente externo. Esse sensor emite ondas ultrassônicas de 40 kHz, que se propagam no ar até encontrar a superfície da água, onde são refletidas e retornam ao sensor. A fórmula utilizada para calcular a distância é:

$$Distância = \frac{(Tempo \times Velocidade\ do\ Som)}{2}$$

onde a velocidade do som no ar é aproximadamente 340 m/s, ajustada para compensar pequenos desvios. A altura da coluna de água (h) foi calculada subtraindo-se a distância medida (d) da altura total do recipiente (H), definida em 63 cm:

$$h = H - d$$

A porcentagem de enchimento foi derivada como:

$$Porcentagem\ de\ Enchimento = \left(\frac{h}{H}\right) \times 100$$

Para o primeiro teste, a altura de 63 cm foi considerada devido a altura do solo até a mesa em que foi posicionado. Conforme Figura 26, utilizamos um recipiente para inserirmos a água e testarmos se estava operando conforme esperado.

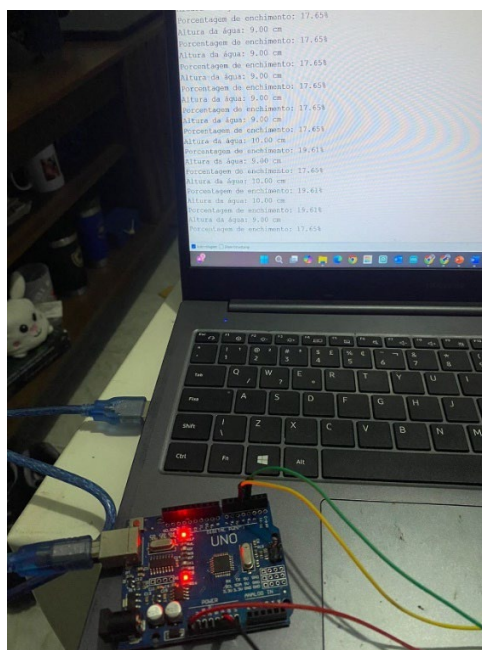
Figura 26: Configuração de Teste do Sensor de Nível Ultrassônico com Recipiente de Água



Fonte: Autoria própria (2024)

Na Figura 27 mostrou que os dados foram totalmente fidedignos com o que estávamos esperando, conforme mostrado no monitor serial.

Figura 27: Monitoramento de Dados de Nível de Água no Monitor Serial com Arduino Uno



Fonte: Autoria própria (2024)

O sensor foi conectado ao Arduino Uno R3 SMD da seguinte forma:

VCC do sensor ao pino de 5V do Arduino para a alimentação.

GND do sensor ao GND do Arduino para estabelecer o circuito de aterramento.

TRIG do sensor ao pino digital 3 do Arduino, configurado como saída para controlar o envio do pulso ultrassônico.

ECHO do sensor ao pino digital 4 do Arduino, configurado como entrada para capturar o retorno da onda.

No código, a biblioteca Ultrasonic.h simplificou a integração com o sensor. O intervalo de leitura foi ajustado para 2 segundos, evitando sobrecarga de dados e mantendo a precisão na leitura.

A escolha deste sensor ultrassônico impermeável justifica-se pela capacidade de fornecer medições confiáveis em condições de umidade, requisito fundamental para simular monitoramento em áreas próximas a cursos d'água e para a replicabilidade em ambientes reais. Abaixo, pela Figura 28, pode-se observar com mais detalhes o tipo de sensor utilizado.

Figura 28: Sensor Ultrassônico JSN-SR04M-2 para Medição de Distância



Fonte: MSS Eletrônica (2024)

3.2.2 Sensor de Fluxo de Água YF-S201

Para a medição do fluxo de água, o sensor YF-S201 foi escolhido por sua precisão em medir vazões entre 1 e 30 L/min, através de uma turbina interna conectada a um sensor de efeito Hall. Este módulo é apresentado pela Figura 29 abaixo.

Figura 29: Sensor de Fluxo de Água YF-S201 para Medição de Vazão



Fonte: Mercado Livre (2024)

A cada passagem de água, a turbina gira, gerando pulsos elétricos que representam o volume de água. Os cálculos foram baseados na fórmula:

$$Q = \frac{(P)}{7,5}$$

onde P representa a contagem de pulsos por segundo, e o fator de conversão de 7.5 foi fornecido pelo fabricante.

As conexões físicas foram realizadas da seguinte maneira:

Fio vermelho do sensor ao pino de 5V do Arduino para alimentação.

Fio preto ao GND do Arduino para o circuito de aterramento.

Fio amarelo ao pino digital 7 do Arduino, configurado como uma entrada com interrupção para registrar com precisão os pulsos, mesmo durante outras operações do Arduino.

Esse sensor foi testado com uma mangueira, conforme Figura 30, para simular diferentes taxas de fluxo, verificando-se a variação de pulsos conforme o ajuste da pressão da água. A

interrupção configurada no pino 7 garantiu que cada pulso fosse registrado independentemente do estado do loop principal.

Figura 30: Teste do Sensor de Fluxo de Água YF-S201 com Mangueira para Simulação de Taxas de Fluxo



Fonte: Autoria própria (2024)

A justificativa para o uso do sensor YF-S201 baseia-se em sua precisão e capacidade de operação em faixas de vazão adequadas para o objetivo do estudo. Esse sensor permite monitorar variações de fluxo com alto grau de precisão, característica essencial para a replicação de condições de monitoramento hídrico em campo.

3.2.3 Programação no Arduino Uno R3 SMD e Integração dos Dados

A programação foi realizada na plataforma Arduino IDE, utilizando-se as bibliotecas Ultrasonic.h para o sensor de nível e LoRa.h para a comunicação com os módulos LoRa. No código, a função `setup()` inicializa a comunicação serial para 9600 bps, configura os pinos de cada sensor e assegura que o sistema esteja pronto para receber dados de forma contínua.

A função `loop()` é composta por duas seções principais: leitura do sensor ultrassônico e contagem de pulsos do sensor de fluxo. A cada 2 segundos, o Arduino envia um pulso pelo pino TRIG, recebe o retorno pelo pino ECHO, calcula a altura da coluna de água e converte o valor para porcentagem de enchimento. Em paralelo, os pulsos do sensor de fluxo são registrados, permitindo o cálculo da vazão em tempo real. Esses dados são integrados em uma string e transmitidos via LoRa para o módulo receptor.

Os cálculos de altura e vazão são atualizados no display OLED para visualização contínua, permitindo o monitoramento em tempo real dos dados coletados.

3.2.4 Transmissão de Dados e Monitoramento Remoto

Para monitorar os dados de nível e fluxo de água em tempo real, foi utilizado o módulo LoRa ESP32 TTGO LILYGO T3 V1.6.1 Pax Counter, conforme demonstrado o seu modelo na Figura 31 abaixo, configurado na frequência de 915 MHz, ideal para comunicação em longo alcance. O módulo possui um display OLED de 128x64 pixels, facilitando a visualização direta dos dados.

Figura 31: Módulo LoRa ESP32 TTGO LILYGO T3 V1.6.1 Pax Counter com Display OLED para Monitoramento em Tempo Real



Fonte: Mercado Livre (2024)

O display OLED foi configurado com a biblioteca Adafruit_SSD1306, enquanto o módulo LoRa foi inicializado utilizando LoRa.h. A conexão USB com o notebook forneceu alimentação e permitiu o monitoramento em tempo real durante os testes.

O uso do módulo LoRa justifica-se pela sua capacidade de transmissão de dados a longas distâncias, característica essencial para um sistema de monitoramento remoto. O alcance e a estabilidade oferecidos pelo módulo são ideais para ambientes que exijam comunicação confiável, como o monitoramento de cursos de água em áreas extensas e de difícil acesso.

3.2.5 Configuração e Detalhes Técnicos do Módulo LoRa ESP32 LILYGO T3 V1.6.1 Pax Counter

O ESP32 LILYGO T3 V1.6.1 Pax Counter integra três tipos de comunicação: WiFi, Bluetooth e LoRa, sendo que este último foi utilizado para a comunicação do projeto. Equipado com o chip SX1276, o módulo permite comunicação a longa distância com baixa potência, essencial para monitoramento em áreas remotas.

No código, a biblioteca SPI.h gerencia a comunicação entre os pinos de controle do módulo (SCK, MISO, MOSI e SS), enquanto LoRa.h configura a frequência de 915 MHz para o protocolo LoRa. O display OLED é atualizado a cada segundo com os dados de altura, percentual de enchimento e vazão, garantindo precisão e estabilidade. Na Figura 32, temos a demonstração de seus pinos com legendas, indicando suas funções específicas, como ADC, GPIO, I2C, SPI, entre outros.

Figura 32: Esquema de Pinagem do Módulo LoRa ESP32 TTGO LILYGO T3 com Funções ADC, GPIO, I2C e SPI



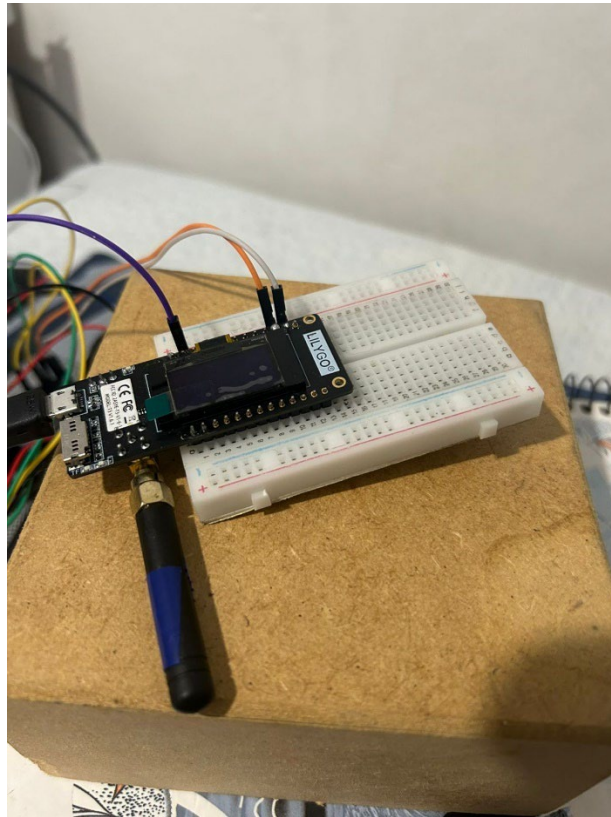
Fonte: Mercado Livre (2024)

3.2.6 Módulo LoRa Sender (Transmissor)

O módulo LoRa Sender desempenha a função essencial de capturar os dados de nível e fluxo de água enviados pelo Arduino Uno, formatá-los adequadamente e transmiti-los ao módulo LoRa Receiver via comunicação sem fio de longo alcance. Esta configuração é fundamental para simular um sistema de monitoramento remoto, onde dados capturados em campo são enviados para um ponto de recepção distante, refletindo uma aplicação real de monitoramento ambiental.

A montagem do LoRa Sender foi realizada em uma protoboard para permitir fácil conexão e ajustes durante o processo de prototipagem. O módulo ESP32 TTGO LILYGO T3 V1.6.1 Pax Counter foi conectado a um notebook via USB-C, que serve tanto para a alimentação quanto para monitoramento e ajustes de programação em tempo real. A antena SMA, acoplada ao módulo, foi orientada de forma a otimizar o alcance do sinal, maximizando a estabilidade na transmissão dos dados para o LoRa Receiver. Na Figura 33 pode-se observar sua montagem:

Figura 33: Montagem do Módulo LoRa ESP32 TTGO LILYGO T3 V1.6.1 Pax Counter como LoRa Sender em Protoboard



Fonte: Autoria própria (2024)

Na programação, o Sender foi configurado para receber os dados do Arduino Uno por meio de comunicação serial. A cada nova leitura de dados, o Sender processa uma string de dados no formato "altura,porcentagem,fluxo", que representa, respectivamente, a altura da coluna de água em centímetros, o percentual de enchimento do recipiente e a vazão em litros por minuto. Essa string é segmentada e convertida em variáveis do tipo float para que possam ser manipuladas pelo código. Os pinos de controle do LoRa foram configurados utilizando as bibliotecas SPI.h e LoRa.h, essenciais para estabelecer a comunicação entre os módulos.

O programa configura os pinos do módulo LoRa, como SCK, MISO, MOSI, SS, RST e DI0, e realiza a inicialização da biblioteca LoRa.h com a frequência de 915 MHz, ideal para comunicação de longo alcance sem interferências. A biblioteca Adafruit_SSD1306, por sua vez, permite o controle do display OLED integrado ao módulo, o qual apresenta os dados transmitidos, como altura da água, percentual de enchimento e vazão, facilitando a verificação direta dos dados em tempo real.

O fluxo do código no Sender é estruturado de forma a receber os dados do Arduino, formatá-los e enviá-los para o Receiver com um intervalo de 1 segundo entre cada envio. Esse

tempo de atualização foi escolhido para garantir a atualização constante dos dados sem sobrecarregar o sistema ou causar atrasos. O display OLED também é atualizado a cada segundo, exibindo os valores recebidos para uma rápida verificação visual, proporcionando uma interface amigável e útil para o usuário durante o monitoramento.

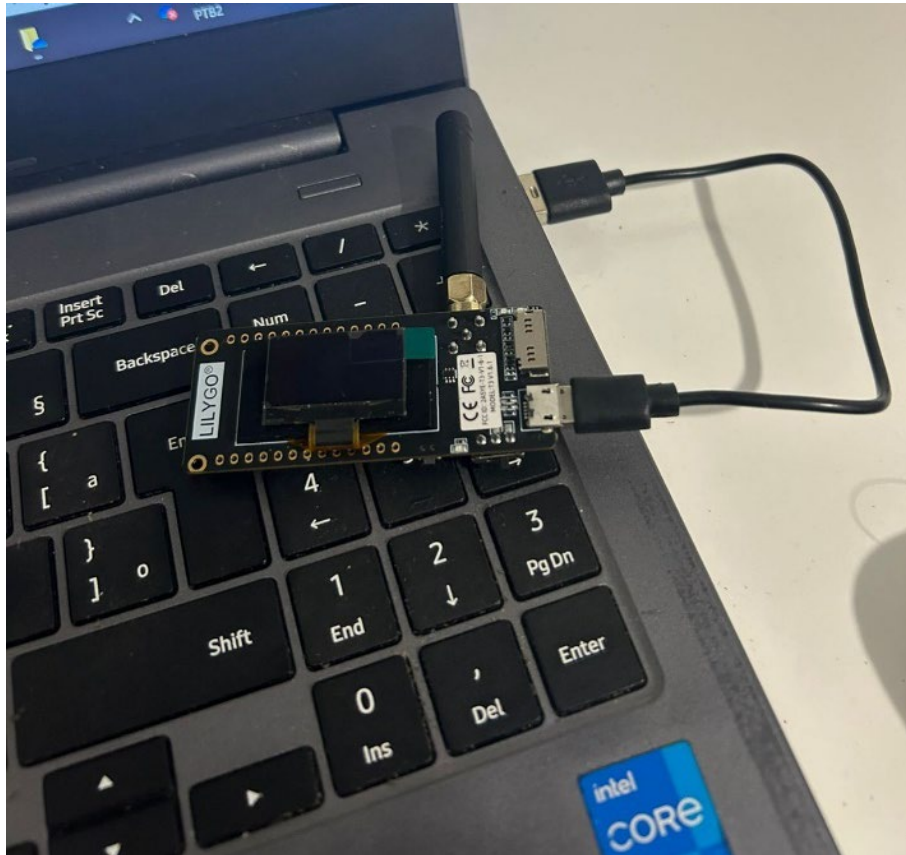
A implementação do Sender é projetada para funcionar continuamente, recebendo e enviando os dados de forma precisa. Essa configuração permite simular um sistema de monitoramento remoto em ambientes reais, como um curso de água, onde os dados capturados remotamente precisam ser transmitidos para uma base de controle para análise e tomada de decisão.

3.2.7 Módulo LoRa Receiver (Receptor)

O módulo LoRa Receiver é a unidade de recepção de dados no sistema e foi projetado para coletar os pacotes transmitidos pelo Sender, interpretar os dados e exibi-los em seu display OLED. Esse processo replica o funcionamento de um sistema de monitoramento remoto em campo, permitindo que informações capturadas em um ponto remoto sejam visualizadas em tempo real no ponto de recepção.

Fisicamente, o módulo LoRa Receiver não precisou ser montado em uma protoboard, porém, assim como o Sender, foi conectado a um notebook via USB-C. Essa conexão USB serve tanto para alimentação quanto para monitoramento dos dados e ajustes de código em tempo real. A antena SMA foi posicionada para garantir uma boa recepção do sinal transmitido pelo Sender, otimizando a qualidade do link de comunicação de longo alcance. Na Figura 34, é demonstrado sua conexão via USB-C.

Figura 34: Conexão do Módulo LoRa ESP32 TTGO LILYGO T3 V1.6.1 Pax Counter como LoRa Receiver via USB-C para Monitoramento



Fonte: Autoria própria (2024)

Na programação, a recepção de dados via LoRa é gerenciada pela biblioteca LoRa.h, que configura os pinos de controle e inicializa a recepção dos dados. Quando um pacote é recebido, o código usa a função `LoRa.parsePacket()` para iniciar a leitura dos dados. Cada valor - altura da água, porcentagem de enchimento e vazão - é extraído do pacote de dados utilizando manipulações de string e convertido para o formato float, garantindo a precisão dos valores recebidos e permitindo sua exibição adequada.

A biblioteca `Adafruit_SSD1306`, utilizada para controlar o display OLED, permite que as informações sejam atualizadas a cada segundo, sincronizando o display com os dados recebidos do Sender. A tela OLED exibe a altura da água em centímetros, a porcentagem de enchimento e a vazão em litros por minuto, facilitando a análise dos dados diretamente no Receiver. O intervalo de atualização de 1 segundo foi mantido para garantir uma transmissão contínua e evitar atrasos na recepção dos dados.

Além do display OLED, os dados recebidos são enviados para o monitor serial, permitindo que sejam visualizados diretamente no notebook conectado ao Receiver. Essa

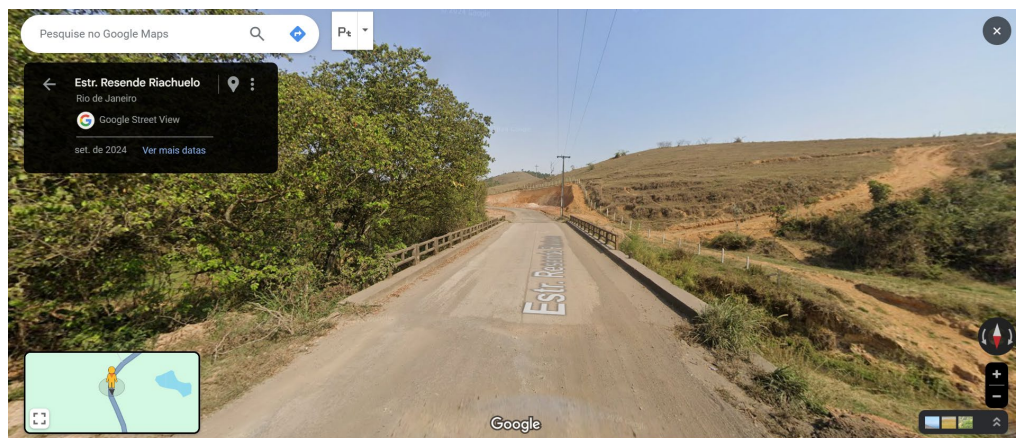
funcionalidade é especialmente útil para verificar a integridade dos dados transmitidos e realizar testes durante a fase de prototipagem e ajuste do sistema.

A estrutura de programação do LoRa Receiver e sua configuração para comunicação de longo alcance, com a frequência de 915 MHz, garantem a estabilidade e a robustez do sistema em um ambiente simulado de monitoramento remoto. Dessa forma, o Receiver completa o sistema, recebendo e apresentando os dados de maneira confiável, o que possibilita o monitoramento preciso e contínuo de variáveis ambientais, como nível e fluxo de água em um cenário simulado de monitoramento de recursos hídricos.

3.3 INSTALAÇÃO DO SISTEMA DE MONITORAMENTO NA PONTE

Para validar o sistema de monitoramento em um ambiente real, escolhemos como local de instalação uma ponte localizada atrás da antiga Votorantim, atualmente ArcelorMittal, situada na Estrada Resende Riachuelo. A escolha deste local foi estratégica devido à sua proximidade com um curso d'água que apresenta características ideais para simular diferentes condições de monitoramento hídrico. A Figura 35 ilustra a vista superior da ponte, mostrando a estrutura utilizada como suporte para a instalação do sistema.

Figura 35 - Vista superior da ponte localizada na Estrada Resende Riachuelo



Fonte: Google Street View (2024)

Após a definição do local, realizamos uma descida até a margem do rio para avaliar as condições físicas e estruturais da área destinada à instalação. Esta etapa foi fundamental para identificar os fatores que poderiam comprometer o desempenho do sistema, como a presença de vegetação densa, correnteza elevada ou possíveis danos à infraestrutura em períodos de

chuva intensa. A Figura 36 mostra o ponto avaliado sob a ponte, destacando a posição onde o sistema seria fixado e os desafios enfrentados durante o planejamento da instalação.

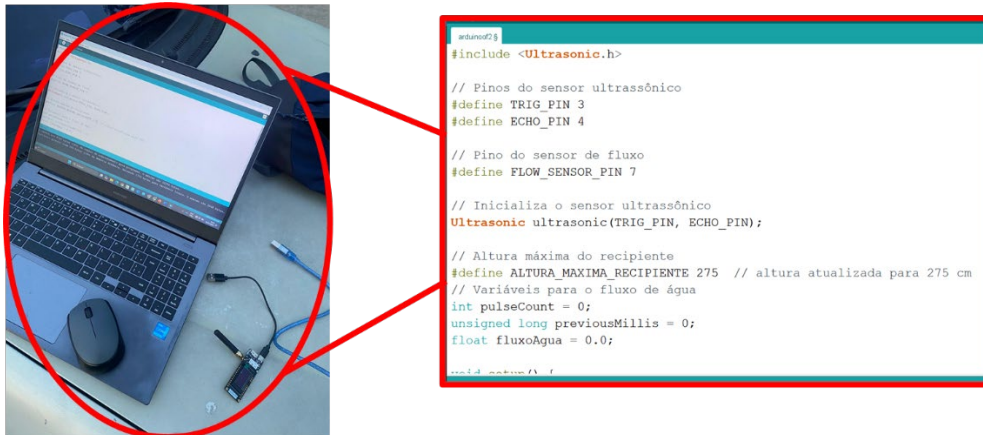
Figura 36 - Análise do local sob a ponte para instalação do sistema de monitoramento



Fonte: Autoria própria (2024)

Inicialmente, a instalação foi planejada para coincidir com um período de chuvas, com o objetivo de testar o comportamento do sistema em condições adversas. No entanto, as chuvas previstas não ocorreram no dia da instalação, obrigando a equipe a ajustar os parâmetros e realizar o monitoramento em condições de fluxo normal do rio. Para manter a consistência com os testes realizados anteriormente em ambiente controlado, definimos uma altura de referência, chamada de “altura zero,” que serviria como base para a detecção de mudanças no nível do rio. Após medições com uma trena, determinamos que a altura total da ponte era de 8 metros em relação ao leito do rio. Para o sensor de nível, configuramos uma altura inicial de 2,75 metros, que seria utilizada como referência para calcular as variações no nível da água. A Figura 37 ilustra a configuração inicial da programação realizada no local.

Figura 37 - Configuração inicial e programação do sistema com altura de referência



Fonte: Autoria própria (2024)

A montagem do sistema exigiu a preparação de cabos extensores, feitos com jumpers, para conectar o sensor de fluxo instalado diretamente na água ao restante do circuito, que seria suspenso acima do nível do rio para proteção contra danos. Na Figura 38, é possível observar o processo de extensão dos jumpers, garantindo que o sensor de fluxo ficasse submerso enquanto o restante do circuito permanecia protegido.

Figura 38 - Preparação dos jumpers para conexão do sensor de fluxo no rio



Fonte: Autoria própria (2024)

Para proteger o sensor de fluxo contra infiltrações e danos causados pela exposição direta à água, desenvolvemos um sistema de tubulação utilizando canos e nípeis rosqueáveis. Este

sistema foi projetado para alojar o sensor dentro de um balde vedado, equipado com flanges para impedir a entrada de água. O balde foi posicionado estrategicamente dentro do rio, garantindo que o sensor ficasse submerso e protegido, como mostrado na Figura 39 e Figura 40.

Figura 39 - Montagem do sensor de fluxo em um balde com vedação



Fonte: Autorial própria (2024)

Figura 40 - Sistema de tubulação e montagem do sensor de fluxo no rio



Fonte: Autorial própria (2024)

Para estabilizar o balde e evitar deslocamentos provocados pela correnteza, utilizamos pedras como contrapeso em seu interior. Adicionalmente, bambus foram fixados no leito do rio, criando uma estrutura de suporte que assegurou a estabilidade do sistema. Essa abordagem foi essencial para replicar as condições reais de operação em um ambiente natural.

No caso do sensor de nível, juntamente com o módulo LoRa e o Arduino, a alimentação foi realizada por meio de uma bateria de 9V conectada a uma protoboard. A protoboard distribuiu a tensão necessária para o sensor ultrassônico, o sensor de fluxo e o módulo LoRa. Para proteger o circuito contra intempéries, como chuva, todo o conjunto foi alojado em um segundo balde, que recebeu adaptações para passagem dos fios. Estas passagens foram vedadas com canos acoplados, garantindo que o sistema permanecesse seco e funcional. A Figura 41 apresenta o circuito completo montado dentro do balde.

Figura 41 - Montagem interna do circuito no balde com vedação contra água



Fonte: Autoria própria (2024)

A instalação do circuito na ponte foi realizada com o auxílio de fios resistentes, que suportaram o peso do balde contendo o circuito de monitoramento. O balde foi suspenso a partir da ponte, sendo ajustado até atingir a altura programada de 2,75 metros acima do nível do rio. A instalação foi concluída após a conexão dos jumpers ao sensor de fluxo posicionado no rio. A Figura 42 ilustra a configuração final do sistema, mostrando o balde suspenso na ponte e o sensor de fluxo conectado no leito do rio.

Figura 42 - Instalação final do circuito na ponte com sensor de nível a 2,75 metros do rio



Fonte: Autoria própria (2024)

Após a conclusão da instalação, o módulo LoRa receptor foi configurado seguindo o mesmo procedimento utilizado nos testes anteriores, realizado em casa. A alimentação do receptor foi feita por meio de conexão USB ao notebook, permitindo o monitoramento em tempo real dos dados coletados pelo sistema de sensores.

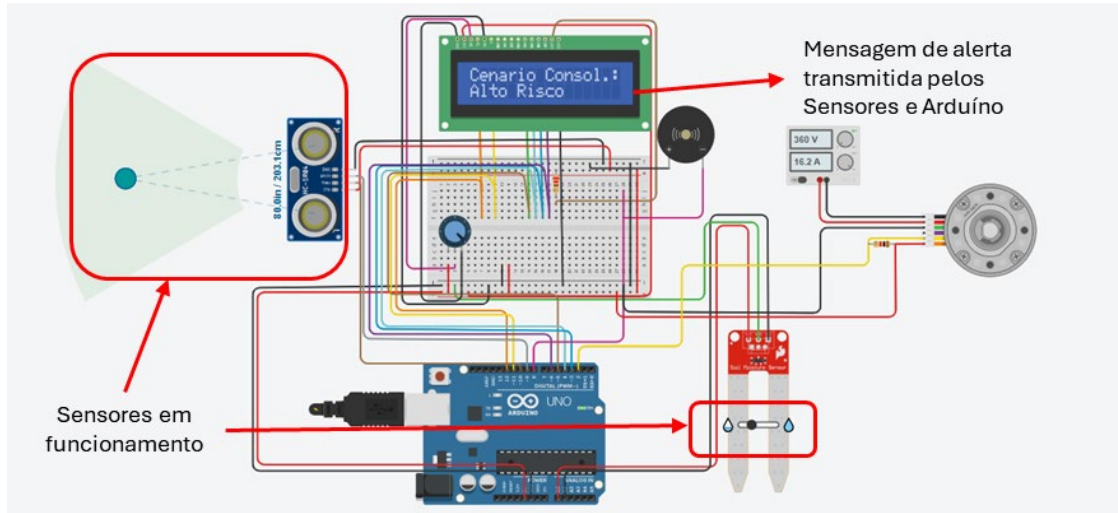
4 RESULTADOS E DISCUSSÃO

4.1 PROJETO VIRTUAL FINAL

O resultado virtual do projeto, consistiu na simulação do projeto em funcionamento utilizando o Software Tinkercad, conforme Figura 43. Com a utilização do Software que possibilita o uso dos materiais mencionados no capítulo anterior, foi possível realizar a montagem de um circuito para realizar a simulação do monitoramento do fluxo e nível de água. A ideia inicial do projeto, consiste na construção de um sistema que possua confiabilidade e eficácia na análise das variáveis do fluxo e nível da água através de sensores, de módulos LoRa e que ao mesmo tempo seja capaz de emitir um sinal sonoro de alerta através da utilização de um buzzer, que indicará uma possível enchente ou tromba d'água, entretanto, não foi possível integrar o módulo de LoRa neste projeto devido ao software não disponibilizar este componente eletrônico para uso, sendo substituído virtualmente por um display LCD. Em contrapartida, através do Tinkercad, foi possível realizar a utilização do buzzer (piezo) para a emissão de sinais

sonoros de alerta, o que não foi possível no projeto físico, assim, garantindo as premissas da objetivo inicial do projeto.

Figura 43: Protótipo Virtual no Tinkercad



Fonte: A autoria própria (2024)

Sendo assim, a simulação no *software* Tinkercad permitiu observar o comportamento dos componentes em um ambiente virtual, facilitando ajustes no código e na estrutura de ligações do circuito antes de sua implementação física. A utilização dos sensores de nível de água e o sensor ultrassônico demonstrou uma alta sensibilidade na detecção de alterações de nível e distância, fundamentais para o monitoramento eficiente em cenários com variação do fluxo e nível de água.

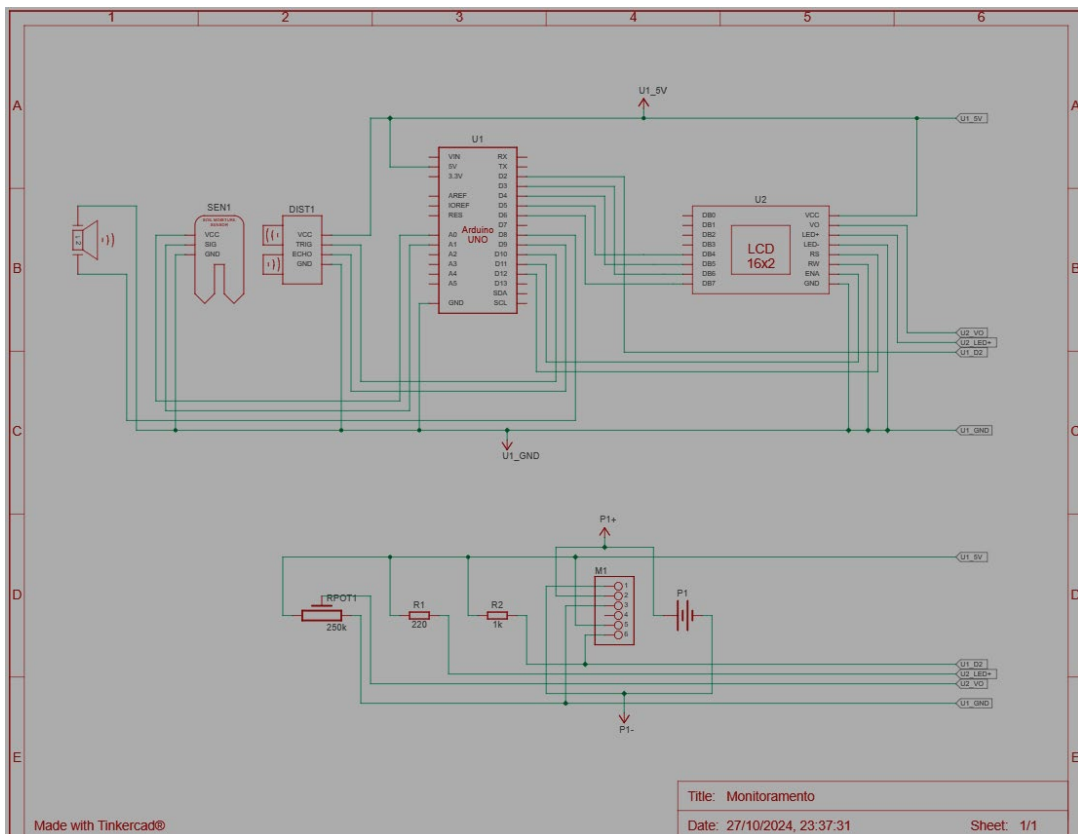
Os dados captados pelos sensores foram processados pelo Arduino Uno R3, que, em seguida, acionou o buzzer (piezo) para emitir alertas sonoros. Esse sistema de alerta, de rápida resposta, cumpriu seu papel ao avisar imediatamente sobre uma possível enchente ou tromba d'água, comprovando a viabilidade de uso do circuito para detecção e alerta em situações de risco eminente a vida. Além disso, o display LCD fornece uma interface visual que mostra em tempo real o status dos sensores e o nível detectado, contribuindo para uma análise mais detalhada e monitoramento constante da variação dos comportamentos da água.

Durante o teste, foram identificadas algumas necessidades de ajustes, como a calibração dos sensores para evitar falsos alarmes e melhorar a precisão. A simulação permitiu realizar essas correções, validando a eficácia do circuito para o objetivo proposto. Com esses ajustes, o circuito alcançou um nível de precisão satisfatório para o monitoramento de enchentes, confirmando que o sistema proposto pode ser aplicado em cenários reais, onde o monitoramento

e o alerta preventivo são fundamentais para a segurança das pessoas próximas aos locais suscetíveis a alagamentos ou cabeça d'água.

Portanto, através disto, foi possível realizar as correções necessárias para o funcionamento correto, comprovar a eficácia do projeto, compreender as ligações do circuito, conforme Figura 44, e observar na prática o funcionamento do circuito, cumprindo o objetivo principal que foi proposto pelo tema.

Figura 44: Visão esquemática do circuito



Fonte: Autoria própria (2024)

4.2 PROJETO FÍSICO FINAL

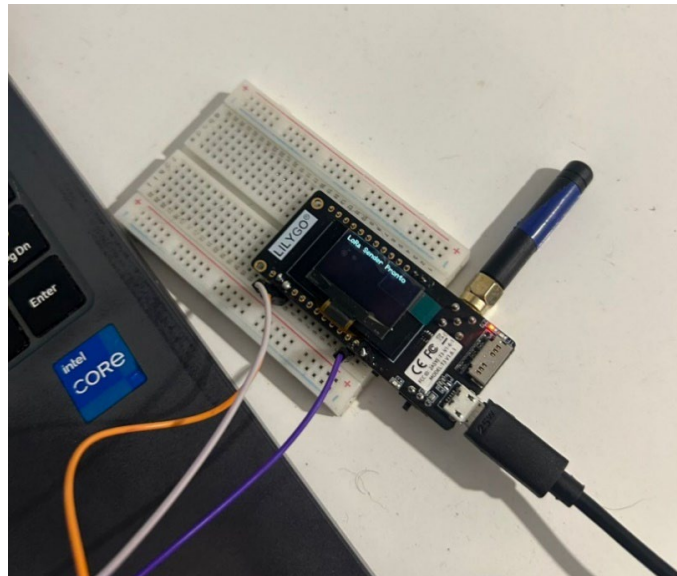
O estudo foi desenvolvido com o objetivo de monitorar o nível e o fluxo de água em um ambiente simulado, utilizando sensores conectados a um Arduino Uno e módulos de comunicação LoRa para a transmissão de dados. Esse sistema foi projetado para simular variações naturais de um corpo d'água, como um rio, permitindo medir a altura da coluna de água e o fluxo em tempo real. Neste capítulo, os resultados são apresentados de forma detalhada, com uma análise para cada condição de teste. A estrutura é complementada por uma tabela e

um gráfico, que facilitam a compreensão dos dados coletados e das respostas do sistema em diferentes cenários.

4.2.1 Teste Inicial de Configuração e Verificação de Leitura Zero

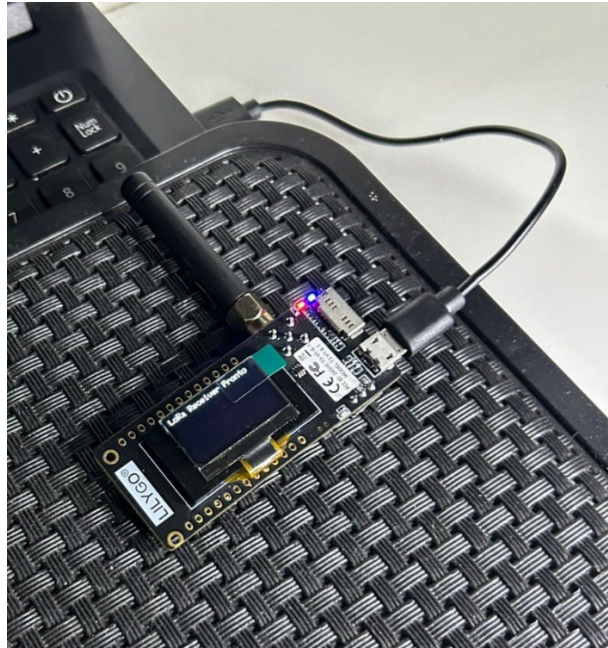
No primeiro teste, foi realizada uma verificação inicial do funcionamento dos sensores e módulos de comunicação. Assim que o sistema foi ligado, os displays dos módulos LoRa exibiram mensagens indicando que estavam prontos para uso, conforme mostram as Figura 45 e Figura 46.

Figura 45: Teste Inicial do LoRa Sender com Display OLED Indicando Pronto para Uso



Fonte: A autoria própria (2024)

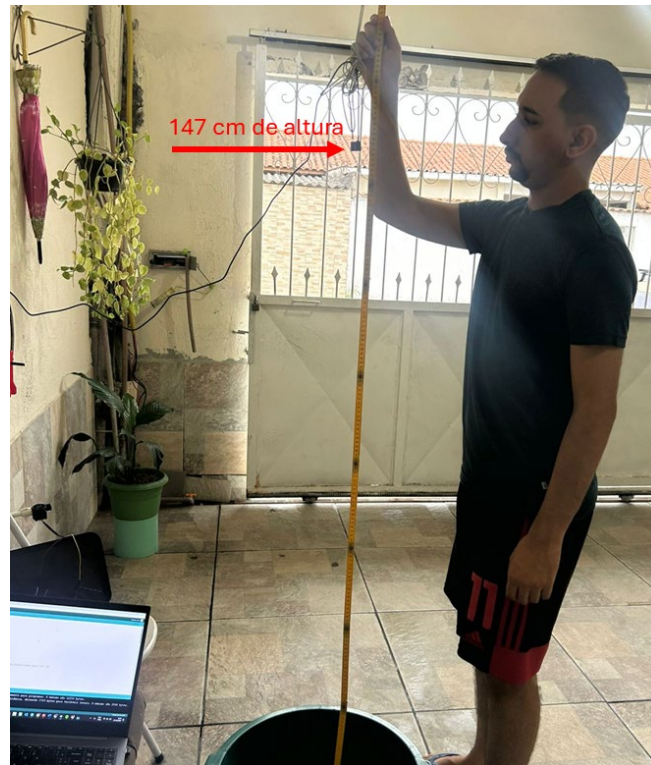
Figura 46: Teste Inicial do LoRa Receiver com Display OLED Indicando Pronto para Uso



Fonte: Autorial própria (2024)

O sensor de nível foi posicionado a uma altura de 147 cm, ajustada no código para simular a altura máxima de um rio em uma situação prática, como mostra a Figura 47 e Figura 48.

Figura 47: Posicionamento do Sensor de Nível a 147 cm para Simulação de Altura Máxima



Fonte: Autorial própria (2024)

Figura 48 - Configuração de Altura Máxima de 147 cm no Código para Sensor Ultrassônico

```
// Inicializa o sensor ultrassônico
Ultrasonic ultrasonic(TRIG_PIN, ECHO_PIN);

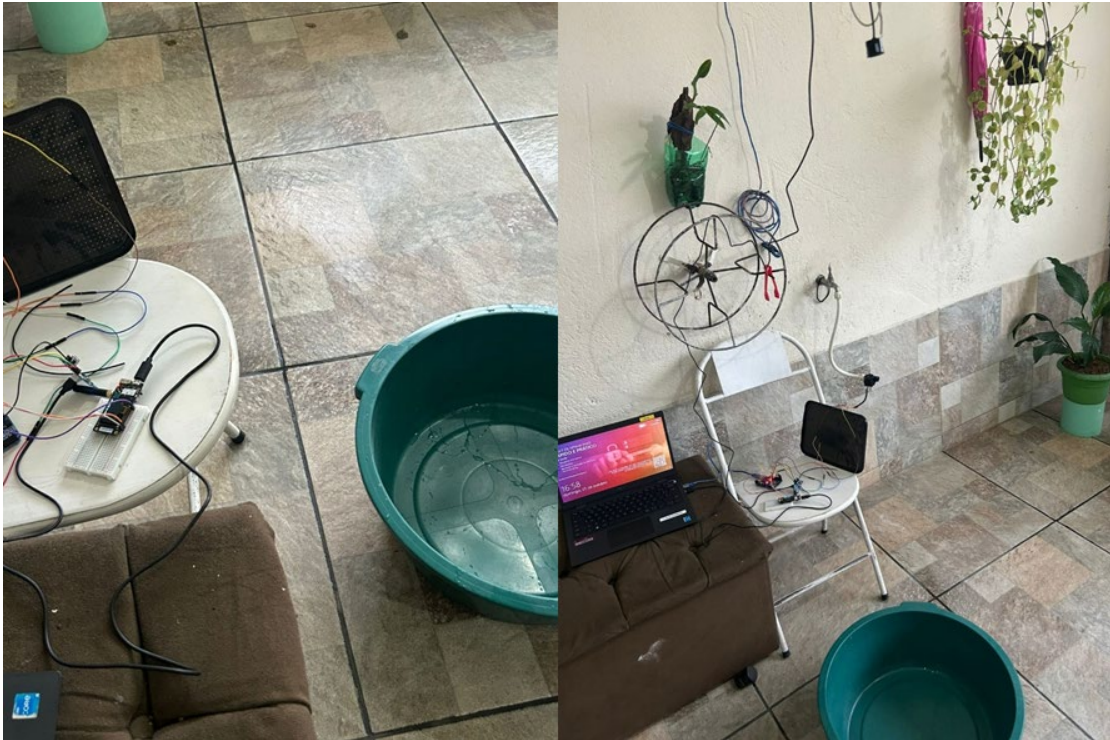
// Altura máxima do recipiente
#define ALTURA_MAXIMA_RECIPIENTE 147 // altura atualizada para 147 cm

// Variáveis para o fluxo de água
```

Fonte: A autoria própria (2024)

Com a bacia vazia, como mostra a Figura 49, o sistema não detectou nenhum nível de água, e os valores de altura e porcentagem de enchimento foram registrados como zero.

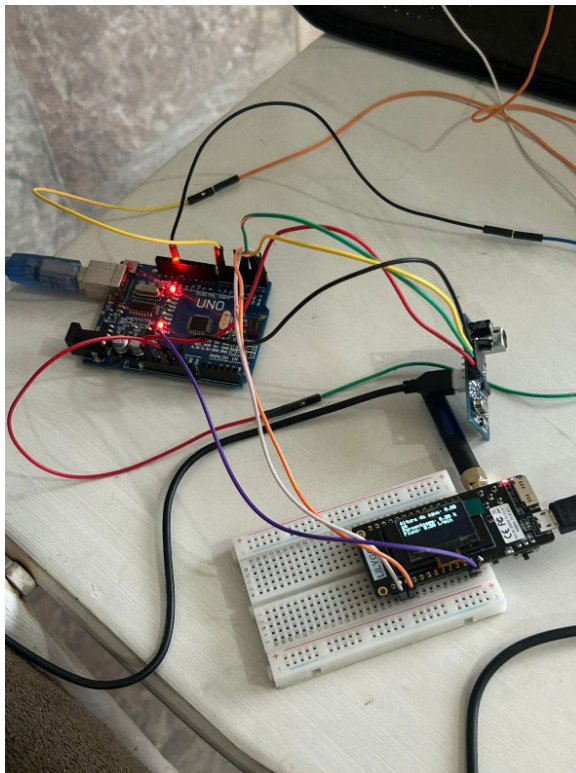
Figura 49: Configuração de Teste com Bacia Vazia para Verificação de Leitura do Sensor de Nível



Fonte: A autoria própria (2024)

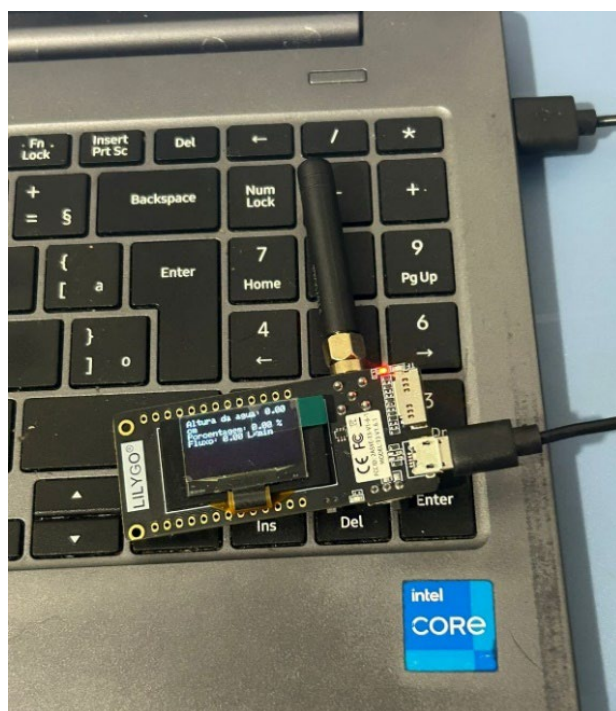
Esse dado foi transmitido com sucesso do LoRa Sender para o LoRa Receiver, como a Figura 50 e Figura 51, confirmando a precisão do sistema em uma condição de ausência de água. Esse teste inicial é fundamental, pois define a referência para os demais testes, garantindo a calibragem dos sensores e a comunicação adequada entre os módulos.

Figura 50: Teste de Transmissão de Dados entre LoRa Sender em Condição de Ausência de Água



Fonte: Aatoria própria (2024)

Figura 51: Teste de Transmissão de Dados entre LoRa Receiver em Condição de Ausência de Água



Fonte: Aatoria própria (2024)

4.2.2 Teste de Detecção de Nível de Água (Baixo)

Para avaliar a sensibilidade do sistema, adicionamos uma pequena quantidade de água na bacia, aproximadamente 2 cm, como mostra a Figura 52. O sensor de nível detectou essa variação, registrando uma altura correspondente e uma porcentagem de enchimento de 1,36%.

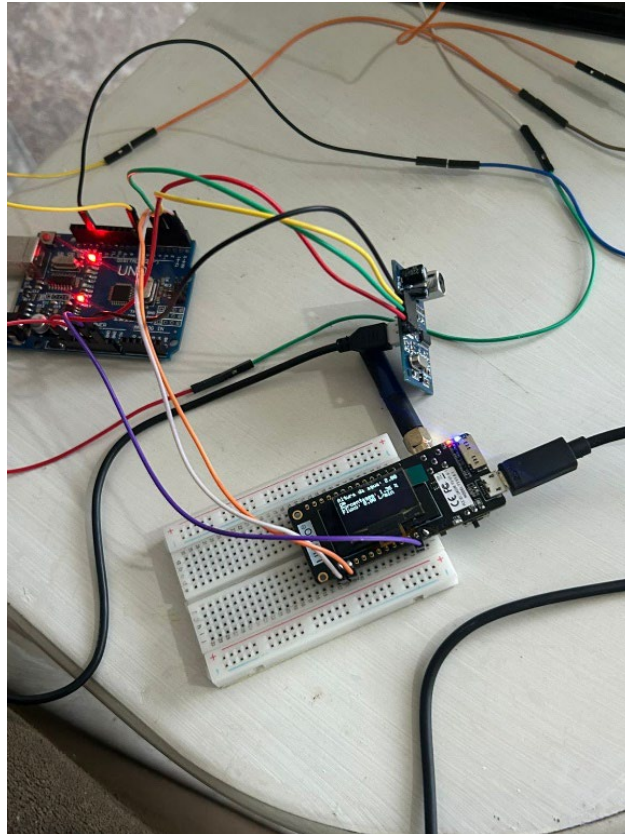
Figura 52: Teste de Sensibilidade do Sensor de Nível com 2 cm de Água na Bacia



Fonte: Autoria própria (2024)

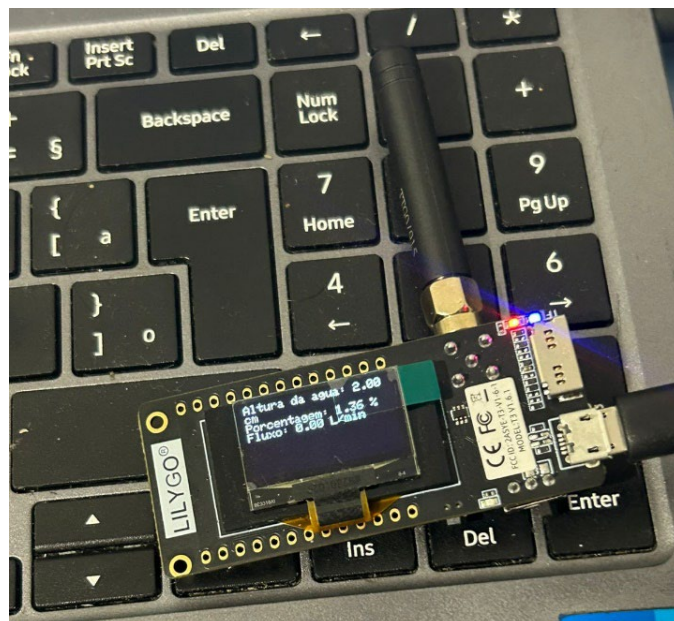
Esses dados foram transmitidos ao LoRa Receiver e exibidos em tempo real, como pode-se observar na Figura 53 e Figura 54. Esse teste demonstra a capacidade do sistema de capturar e transmitir pequenas variações no nível de água com precisão. Essa sensibilidade é crucial para o monitoramento de corpos d'água, onde pequenas mudanças de nível podem representar alterações significativas, especialmente em ambientes naturais sensíveis a flutuações sazonais ou climáticas.

Figura 53: Transmissão de Dados do Sensor de Nível para o LoRa Receiver via LoRa Sender



Fonte: Autoria própria (2024)

Figura 54: Exibição em Tempo Real dos Dados de Nível de Água no Display do LoRa Receiver

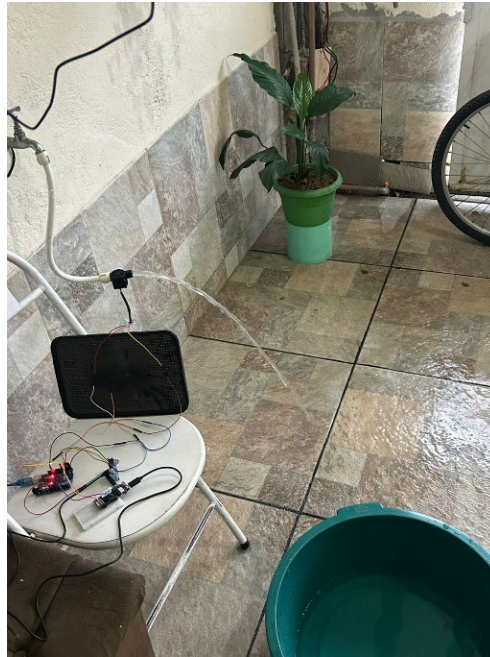


Fonte: Autoria própria (2024)

4.2.3 Teste de Medição de Fluxo de Água

Para avaliar a medição de vazão, conectamos uma mangueira à torneira para gerar um fluxo constante de água através do sensor de fluxo YF-S201, mostrado na Figura 55.

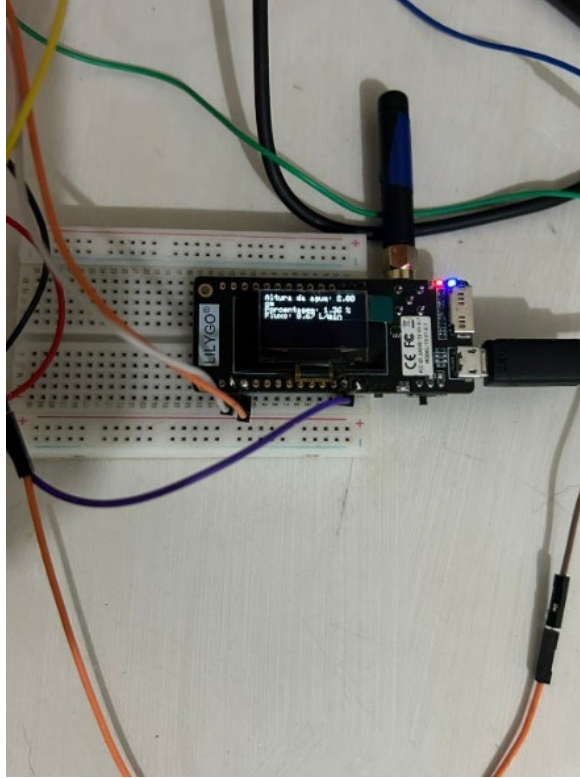
Figura 55: Teste de Medição de Vazão com Fluxo Constante de Água através do Sensor YF-S201



Fonte: A autoria própria (2024)

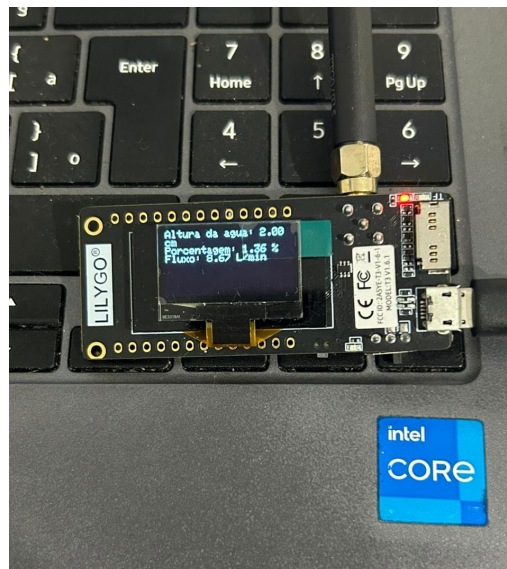
A vazão foi medida em litros por minuto e exibida no display do LoRa Sender, que enviou os dados ao Receiver, como a Figura 56 e Figura 57. A vazão foi registrada em 8,67 L/min, refletindo a resposta do sistema a um fluxo moderado. Esse teste é essencial para validar o uso do sistema em situações em que a vazão é uma variável-chave, permitindo monitorar mudanças na taxa de fluxo em tempo real, característica importante para a análise da dinâmica de corpos d'água ou sistemas de irrigação.

Figura 56: Exibição da Medição de Vazão no Display do LoRa Sender



Fonte: Autoria própria (2024)

Figura 57: Transmissão e Exibição dos Dados de Vazão em Tempo Real no LoRa Receiver

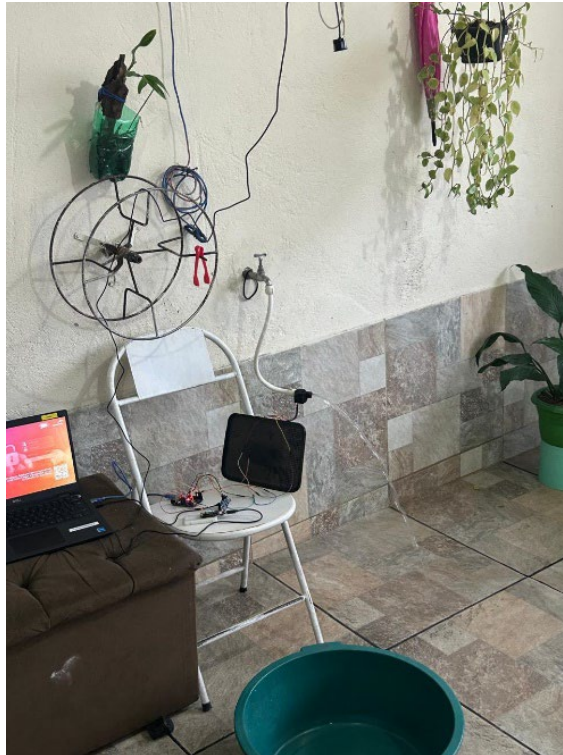


Fonte: Autoria própria (2024)

4.2.4 Simulação de Alto Fluxo e Baixo Nível

Por fim, simulamos uma condição de alto fluxo sem elevação de nível de água, esvaziando a bacia e aumentando a vazão na torneira, como na Figura 58.

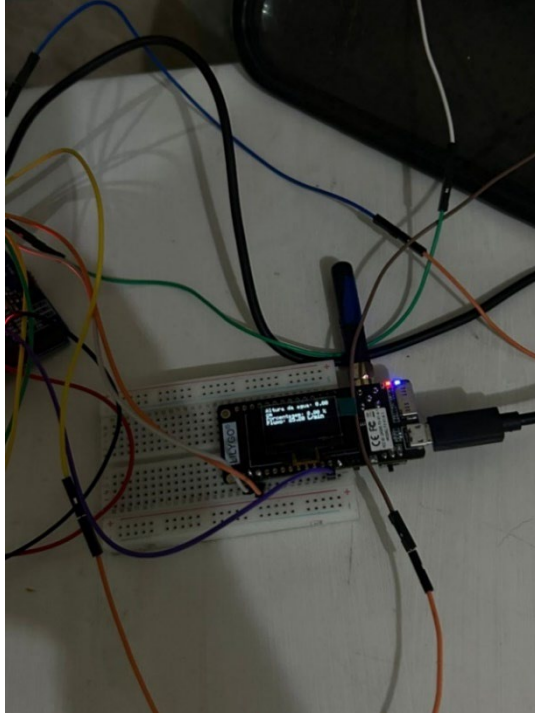
Figura 58: Simulação de Condição de Alto Fluxo com Bacia Vazia para Teste do Sensor de Vazão



Fonte: A autoria própria (2024)

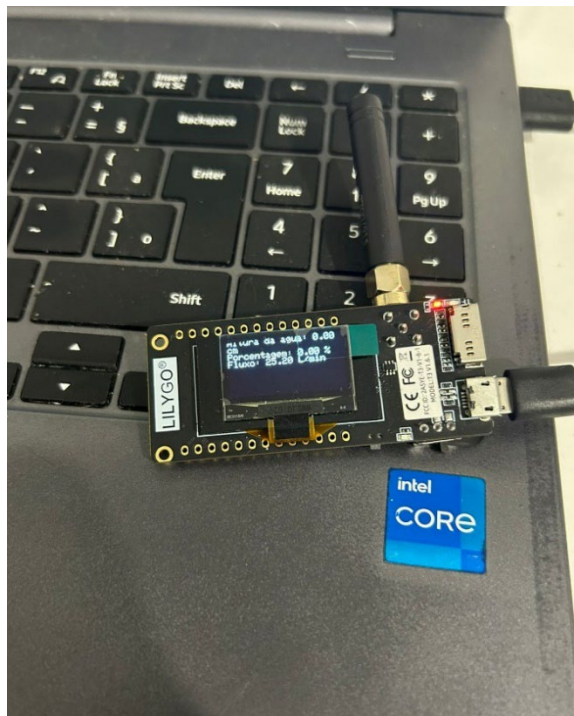
Esse cenário foi criado para avaliar a capacidade do sistema de monitorar vazões elevadas sem alteração no nível de água. O sistema registrou a vazão em aproximadamente 23,20 L/min enquanto a altura de água permaneceu em zero, como mostra na Figura 59 e Figura 60.

Figura 59: Registro de Vazão de 23,20 L/min no LoRa Sender com Nível de Água em Zero



Fonte: Autoria própria (2024)

Figura 60: Exibição em Tempo Real da Vazão Elevada sem Alteração de Nível de Água no LoRa Receiver



Fonte: Autoria própria (2024)

Esse resultado demonstra a flexibilidade do sistema em cenários onde o fluxo e o nível podem variar de forma independente, como em áreas de drenagem ou monitoramento de bacias hidrográficas onde a vazão é alta, mas a altura da coluna d'água permanece constante.

4.2.5 Tabela de Resultados Obtidos

Para facilitar a compreensão dos dados coletados, a Tabela 1 abaixo apresenta um resumo das medições para cada condição de teste, incluindo a altura, a porcentagem de enchimento e o fluxo de água.

Tabela 1: Resumo Das Medições Para Cada Condição De Teste

Condição de Teste	Altura da Água (cm)	Porcentagem (%)	Fluxo (L/min)
Leitura Inicial (Bacia Vazia)	0.00	0.00	0.00
Nível Baixo (2 cm de Água)	2.00	1.36	0.00
Teste de Fluxo (Fluxo Moderado)	2.00	1.36	8.67
Alto Fluxo, Nível Zero	0.00	0.00	23.20

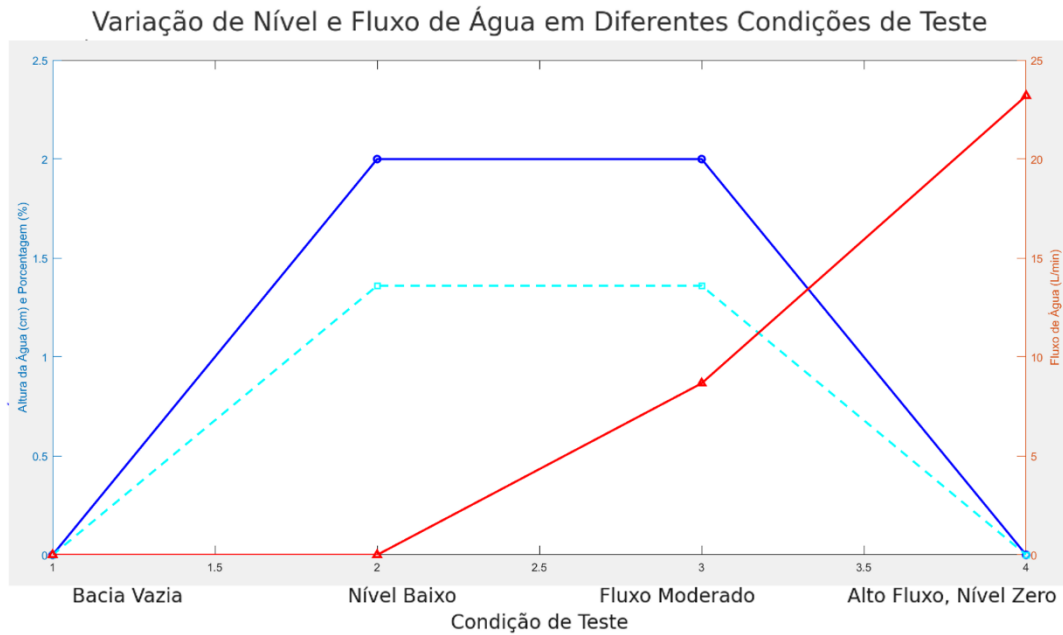
Fonte: Autoria própria (2024)

Esses valores indicam que o sistema foi capaz de capturar e transmitir corretamente as condições de nível e fluxo, mantendo a precisão em cada cenário.

4.2.6 Gráfico de Variação de Nível e Fluxo de Água

O gráfico da Figura 61 a seguir fornece uma representação visual das variações de nível e fluxo de água nas diferentes condições testadas. No eixo vertical esquerdo, estão indicados os valores de altura da água (cm) e a porcentagem de enchimento, enquanto o eixo vertical direito representa a vazão em litros por minuto.

Figura 61: Gráfico das Variações de Nível e Vazão de Água em Diferentes Condições de Teste



Fonte: Autoria própria (2024)

Esse gráfico permite visualizar com clareza a resposta do sistema às diferentes condições. Observa-se que, nas condições de “bacia vazia” e “alto fluxo, nível zero”, o sistema registrou ausência de nível, enquanto o fluxo variou conforme a intensidade simulada. Na condição de “nível baixo”, mesmo com apenas 2 cm de água, o sistema foi capaz de detectar a pequena quantidade, confirmando a sensibilidade do sensor de nível. Em situações de fluxo moderado, a leitura de vazão aumentou de forma proporcional, demonstrando a precisão do sensor YF-S201 em quantificar o movimento da água.

4.2.7 Análise dos Resultados de Teste

A partir da tabela e do gráfico, fica evidente que o sistema é eficaz em monitorar e reportar condições de nível e fluxo de água de maneira independente e precisa. No teste inicial com a bacia vazia, o sistema detectou corretamente a ausência de água, estabelecendo um ponto de referência. No segundo teste, a adição de 2 cm de água comprovou que o sensor de nível possui precisão para detectar pequenas variações. O teste de fluxo demonstrou a capacidade do sistema de registrar variações na taxa de fluxo, independentemente da altura da coluna de água.

No último teste, simulando um fluxo elevado sem alteração no nível, o sistema registrou a vazão de forma precisa, mesmo com a leitura de nível em zero. Esse comportamento reforça

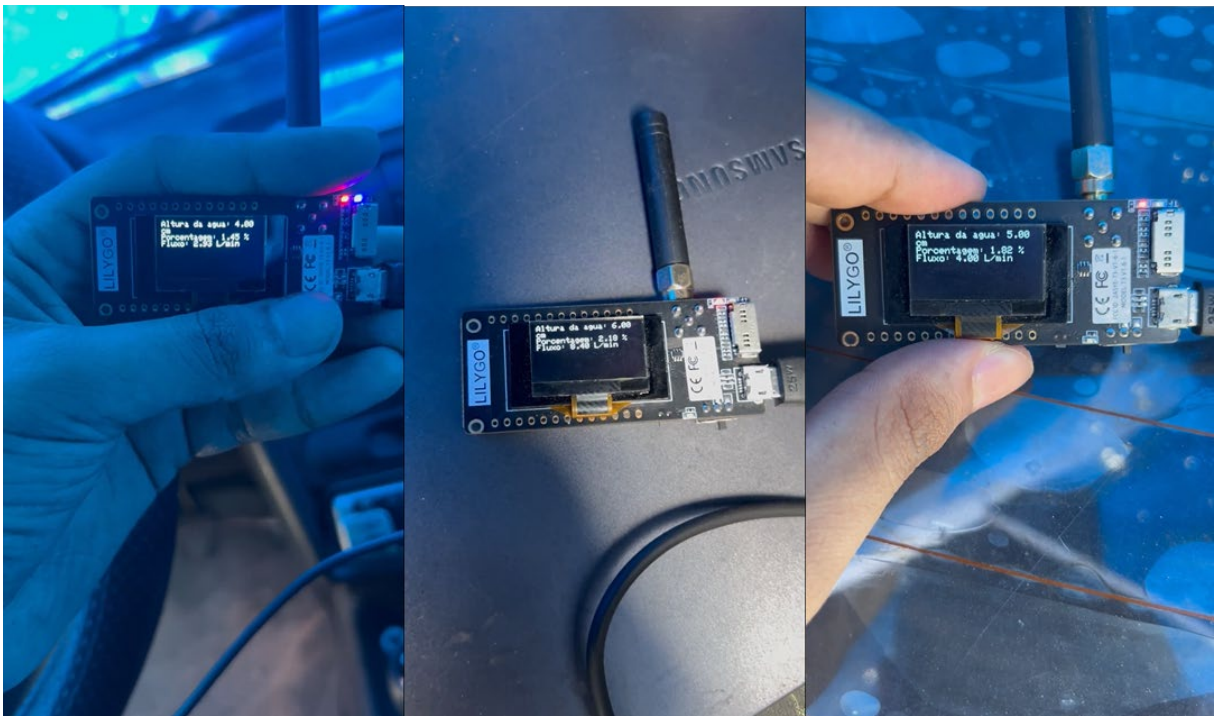
a aplicabilidade do sistema em cenários onde apenas a vazão precisa ser monitorada, como em regiões de captação de água ou controle de drenagem.

Os resultados apresentados indicam que o sistema desenvolvido é capaz de monitorar tanto o nível quanto o fluxo de água de forma contínua e confiável, com potencial para aplicação prática em ambientes de monitoramento hídrico, controle de irrigação e análise de recursos naturais em tempo real.

4.3 RESULTADOS DO MONITORAMENTO NA PONTE

Após a instalação do sistema de monitoramento na ponte, iniciamos o acompanhamento em tempo real do nível e fluxo do rio utilizando o módulo LoRa receptor. O display OLED do módulo mostrou as leituras captadas pelos sensores ao longo do dia, permitindo verificar as alterações de nível e fluxo do rio em diferentes horários. A Figura 62 ilustra três momentos distintos: o início da manhã (esquerda), o começo da tarde (centro) e o meio da tarde (direita). Esses registros evidenciam o funcionamento contínuo e confiável do sistema ao longo do dia.

Figura 62 - Exibição dos dados de nível e fluxo no display do módulo LoRa receptor em diferentes horários do dia

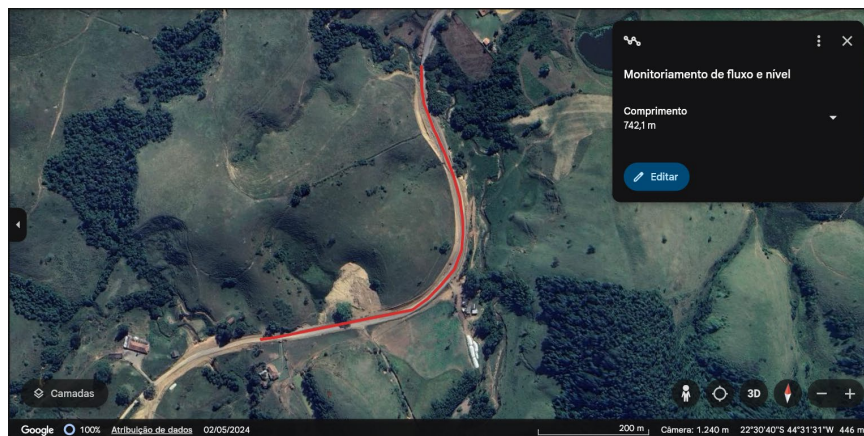


Fonte: A autoria própria (2024)

Durante o monitoramento, identificamos a necessidade de substituir a bateria do sistema em algumas ocasiões, devido à demanda energética constante para alimentar os sensores, o módulo LoRa e o Arduino. Essa troca foi essencial para manter a operação ininterrupta durante todo o período de teste e destaca a importância de considerar melhorias na alimentação do sistema, como o uso de fontes mais robustas ou energias renováveis em futuras implementações.

Além de monitorar localmente, realizamos um teste de alcance do sinal transmitido pelo módulo LoRa. O objetivo foi verificar a distância máxima em que o receptor ainda conseguia captar os dados enviados pelo transmissor instalado na ponte. Para isso, deslocamos o receptor progressivamente para pontos mais distantes, enquanto monitorávamos os dados exibidos no display OLED. O trajeto percorrido foi registrado utilizando o Google Maps, como mostrado na Figura 63. A distância máxima alcançada com sucesso foi de 742,1 metros, confirmando a capacidade do sistema de operar em longas distâncias, característica fundamental para aplicações em áreas remotas.

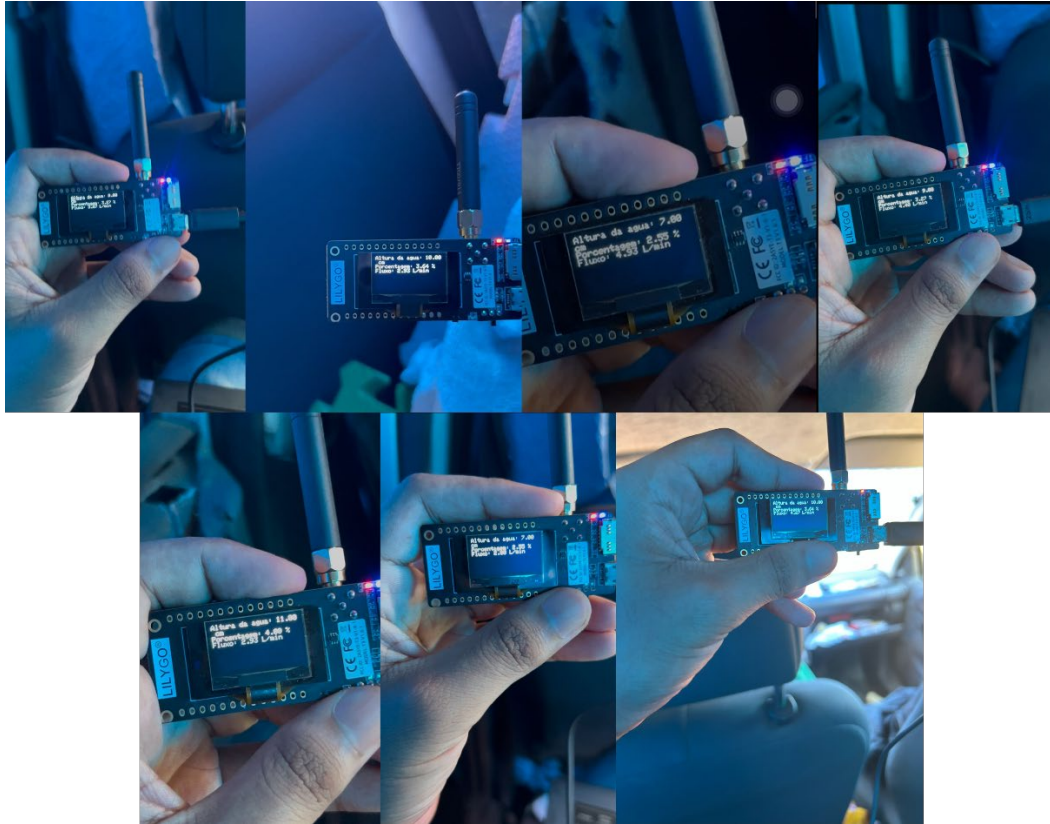
Figura 63 - Trajeto percorrido para o teste de alcance do sinal LoRa, com distância máxima de 742,1 metros



Fonte: Autoria própria (2024)

A Figura 64 apresenta os registros capturados pelo módulo receptor ao longo do trajeto, em ordem sequencial, desde o ponto de instalação inicial na ponte até o local mais distante onde o sinal foi recebido com sucesso. Esses registros demonstram que o sistema manteve sua precisão na transmissão de dados, mesmo em condições de distância progressivamente maiores. A estabilidade da comunicação foi confirmada pela exibição consistente dos valores no display OLED.

Figura 64 - Registros sequenciais capturados pelo módulo LoRa receptor ao longo do trajeto de teste



Fonte: Autoria própria (2024)

4.3.1 Análise dos Dados Coletados no Monitoramento da Ponte

Durante o monitoramento do sistema instalado na ponte, foram registrados dados referentes ao nível de água (em centímetros) e ao fluxo de água (em litros por minuto) em três momentos distintos: início da manhã, início da tarde e meio da tarde. O gráfico apresentado utiliza dois eixos verticais para evidenciar as variações observadas em ambos os parâmetros, com os dados corrigidos refletindo as condições monitoradas.

No primeiro registro, o nível da água foi de 4,00 cm, enquanto o fluxo foi de 2,93 L/min. Esses valores indicam um comportamento inicial estável do rio, típico de um período matutino, quando as condições externas, como atividades humanas ou alterações climáticas, ainda não impactaram significativamente o sistema.

O nível da água aumentou para 6,00 cm, acompanhado por um fluxo de 8,40 L/min. Esse aumento sugere maior volume de água no sistema, possivelmente devido à maior incidência solar ou ao escoamento de águas acumuladas nas margens do rio, levando ao incremento do fluxo detectado.

O nível da água estabilizou-se em 5,00 cm, enquanto o fluxo apresentou uma redução para 4,00 L/min. Essa estabilização indica que o sistema fluvial ajustou-se a um regime mais equilibrado ao longo do período monitorado, com menor variação nos parâmetros detectados.

Os dados destacam capacidade do sistema de monitoramento em capturar essas mudanças em tempo real, fornecendo dados confiáveis para análise da dinâmica do corpo d'água. A leitura simultânea dos dois parâmetros é essencial para identificar tendências e comportamentos que possam influenciar diretamente a gestão dos recursos hídricos.

Além do monitoramento local na ponte, realizamos um teste para verificar o alcance do sistema de transmissão de dados utilizando o módulo LoRa. Durante o deslocamento do receptor, os dados de nível e fluxo de água foram registrados em diferentes distâncias a partir do ponto de instalação inicial. Os dados abaixo apresentam as variações de nível e fluxo em função da distância percorrida, desde a ponte até o ponto mais distante com sinal efetivo, 742,1 metros.

- Ponte (0 m): No ponto de partida, o nível da água foi registrado em 9,00 cm, e o fluxo em 3,87 L/min. Esses valores representaram o comportamento inicial no local de instalação.
- 200 metros: Observou-se uma alteração no nível da água para 10,00 cm, com o fluxo em 2,93 L/min. Essa leitura reflete a continuidade do monitoramento em uma distância ainda próxima ao ponto inicial.
- 400 metros: O nível da água reduziu para 7,00 cm, enquanto o fluxo subiu para 4,93 L/min, indicando a sensibilidade do sistema em captar alterações significativas mesmo em distâncias intermediárias.
- 600 metros: Os dados mostraram um nível de 9,00 cm, com o fluxo reduzido para 4,40 L/min, destacando a precisão na transmissão e registro de dados.
- 742,1 metros: No ponto mais distante registrado, o sistema alcançou um nível de 11,00 cm, e o fluxo retornou para 2,93 L/min, demonstrando o limite funcional da transmissão sem perda de sinal.

Esses resultados validam a capacidade do sistema de operar em longas distâncias, mantendo a integridade dos dados transmitidos e a confiabilidade das medições. A variação dos parâmetros monitorados ao longo do trajeto reforça a aplicabilidade do sistema para monitoramento remoto em áreas extensas e de difícil acesso.

4.3.2 Considerações Sobre os Resultados

Os dados coletados ao longo do percurso validam a capacidade do sistema de operar em longas distâncias, mantendo a integridade das leituras e a estabilidade da transmissão de informações. O módulo LoRa demonstrou eficiência no envio de dados até a distância máxima testada de 742,1 metros, com variações no nível e no fluxo que acompanham o comportamento dinâmico do sistema.

Além disso, os dados registrados evidenciam a sensibilidade do sistema ao captar pequenas alterações nos parâmetros monitorados, mesmo em diferentes distâncias. Essa característica é essencial para aplicações em áreas remotas, onde o acesso físico ao ponto de monitoramento pode ser limitado.

Esses resultados reforçam a viabilidade do sistema desenvolvido para monitoramento hídrico contínuo, com potencial para integração em projetos maiores de gestão de recursos naturais e análise ambiental em tempo real. Ajustes no sistema de alimentação e em parâmetros de transmissão podem ampliar ainda mais seu alcance e eficiência.

5 CONCLUSÃO

Este trabalho, teve como propósito desenvolver um sistema virtual e físico de monitoramento para o nível e fluxo de água em nascentes, com foco na mitigação dos impactos causados por desastres naturais, como as cabeças d'água. Mediante a isto, utilizou-se tecnologias acessíveis e amplamente eficazes, como a plataforma microcontroladora Arduino, sensor ultrassônico, sensor de umidade do solo, sensor de fluxo de água, módulos de LoRa e um *buzzer* possibilitando a criação de protótipos integrados que coletam e analisam continuamente dados sobre o nível e o fluxo de água.

A viabilidade do trabalho foi comprovada através da montagem física do projeto e de simulações virtuais no *software* Tinkercad, que permitiu validar a utilização dos sensores para coleta de dados, os módulos de LoRa para transmitir as informações, o Arduino para processar os dados e a utilização do *buzzer* integrado ao circuito, e através disso, foi realizado a simulação do alerta sonoro proposto nas premissas do projeto, assim, permitindo validar os materiais e métodos propostos. Em contrapartida, na implementação física o *buzzer* não pôde ser integrado devido à ausência de aquisição do material, o que impediu a replicação completa do sistema no protótipo físico, entretanto, essa limitação foi considerada e não comprometeu os objetivos principais do projeto, dado que a funcionalidade sonora foi comprovada na simulação.

Os testes realizados demonstraram a eficiência do sistema em identificar alterações no nível e fluxo de água, acionando alarmes sonoros em situações de risco, o que confirma sua capacidade para futuras aplicações em nascentes reais. A instalação do sistema em nascentes reais trará um avanço significativo para a segurança das pessoas expostas aos riscos e as comunidades locais, fornecendo uma ferramenta eficiente de monitoramento e alerta antecipado para possíveis desastres naturais. Além disso, o uso de tecnologias de comunicação de longa distância, como o módulo LoRa, garantirá a transmissão de dados em áreas remotas, ampliando a aplicabilidade e efetividade do sistema.

Em resumo, este projeto trata de uma questão essencial para a mitigação dos impactos causados por desastres naturais envolvendo água, fornecendo dados e metodologias com potencial para garantir a segurança e a preservação das vidas humanas. Esses esforços são fundamentais para a continuidade do ciclo hidrológico e da vida em várias escalas, contribuindo para a segurança pública e para o desenvolvimento de políticas eficazes de prevenção de desastres naturais.

6 INDICAÇÕES PARA TRABALHOS FUTUROS

Este trabalho, teve como propósito desenvolver um sistema virtual e físico para monitoramento das variações do fluxo e nível de água nas nascentes dos rios. Além das aplicações realizadas durante este trabalho, diante dos resultados obtidos e das limitações identificadas neste trabalho, é possível propor algumas direções para estudos futuros que possam ampliar e aprofundar o conhecimento sobre o tema abordado.

- **Melhoria na robustez dos sensores e materiais utilizados:**

Tendo em vista a exposição do circuito a condições naturais de clima e tempo, a melhoria da robustez dos materiais utilizados possibilitaria a utilização com mais segurança e proteção em relação a fenômenos extremos como chuvas intensas e correnteza forte, garantindo maior durabilidade e aumentando a eficiência do circuito de uma maneira geral.

- **Integração de tecnologias avançadas de monitoramento:**

A integração de drones com LiDAR (Light Detection and Ranging) engrandece os estudos relacionados ao tema, contribuindo para o levantamento de dados precisos e tomadas de decisão. A tecnologia LiDAR aplicada em drones contribui para a leitura e monitoramento de níveis de água em nascentes, oferecendo precisão e eficiência na análise e coleta de informações devido a utilização de pulsos de luz laser para medir distâncias até superfícies ou objetos, assim, criando modelos tridimensionais detalhados que permitem o mapeamento de terrenos e níveis da água com alta resolução, mesmo em áreas de difícil acesso.

- **Utilização de IA para pressagiar causas naturais:**

As redes neurais artificiais permitem pressagiar possíveis desastres hidrológicos, como enchentes e cabeças d'água. Para que a previsão de desastres hidrológicos aconteça, as redes neurais utilizam diversos tipos de dados, como informações meteorológicas referentes a chuvas, temperatura e vento, além dos níveis de água em rios.

O funcionamento começa com o treinamento do modelo, onde as redes neurais aprendem a relacionar entradas relacionadas a dados ambientais e hidrológicos, com saídas esperadas como inundações, utilizando dados históricos. Após esse treinamento, a rede é capaz de prever eventos futuros ao processar novos dados em tempo real.

Por exemplo, para prever uma enchente, a rede pode analisar variáveis como

a quantidade de chuva em um curto período, a capacidade do solo de absorver água e o nível atual dos rios. Se os dados indicarem uma alta probabilidade, o sistema pode emitir alertas precoces, permitindo que autoridades e comunidades tomem medidas preventivas.

- **Implementação de painéis solares em áreas remotas:**

Ao permitir a geração de energia diretamente no local, a implementação de painéis solares em sistemas localizados em áreas remotas apresenta uma solução eficiente e sustentável para reduzir a dependência da troca periódica das baterias. Com isso, os painéis solares garantem maior autonomia energética ao sistema, eliminando a necessidade de transportar baterias carregadas para áreas de difícil acesso, além disso, agrega na redução de custos operacionais.

A energia solar é uma escolha sustentável, por se tratar de fonte limpa e renovável, ela contribui para a redução de emissões de carbono e diminui o impacto ambiental gerado pelo descarte frequente de baterias, tornando essa utilização adequada para operações em regiões ambientalmente sensíveis, onde a preservação do ecossistema é prioritária.

Além disso, os sistemas solares oferecem flexibilidade para expansão quando necessário. Então, caso haja aumento na demanda energética, o sistema pode ser ampliado de maneira modular, adaptando-se às novas necessidades sem grandes alterações na infraestrutura existente.

- **Incorporar rede 5G para melhor eficiência na transmissão de dados:**

Se tratando da coleta de dados referentes ao monitoramento de variações do nível e do fluxo de água, a incorporação da tecnologia de rede 5G pode melhorar significativamente a eficiência na transmissão de dados de áreas remotas para uma central. A alta velocidade de transmissão do 5G permite o envio de grandes volumes de dados, como informações referentes as variações de nível e fluxo de água em tempo real. Além disso, a baixa latência, característica dessa tecnologia, garante que informações críticas, como alertas sobre mudanças rápidas no nível da água, cheguem instantaneamente à central destinada, mesmo com a presença de vales ou áreas com obstáculos naturais.

A compatibilidade do 5G com a Internet das Coisas (IoT) permite a utilização em muitas aplicações. Com a aplicação desta tecnologia, os sensores e módulos de LoRa instalados em áreas remotas, podem enviar dados continuamente para a central de forma mais rápida e eficiente, permitindo o monitoramento

simultâneo de diversos pontos e através do processamento em tempo real, permite integrar a sistemas para alertas automatizados, como notificações de risco de enchentes.

- **Expansão para outras aplicações ambientais:**

O sistema desenvolvido neste trabalho também pode ser adaptado para uma ampla gama de aplicações ambientais, como monitoramento da qualidade da água em rios, lagos e represas. Pesquisas futuras poderiam explorar sensores especializados para detectar poluentes químicos e biológicos, permitindo ações preventivas contra a contaminação. Além disso, a integração com sistemas de gestão ambiental poderia fornecer insights valiosos sobre o impacto humano e industrial em ecossistemas aquáticos, promovendo práticas mais sustentáveis e responsáveis.

- **Desenvolvimento de redes colaborativas:**

Uma direção promissora seria a criação de redes colaborativas de sensores conectados, capazes de monitorar grandes bacias hidrográficas de maneira integrada. Essa abordagem permitiria o compartilhamento de dados em tempo real entre diferentes localidades, facilitando a coordenação entre comunidades, pesquisadores e autoridades responsáveis. Trabalhos futuros podem investigar arquiteturas de redes baseadas em IoT (Internet das Coisas) e tecnologias de comunicação avançadas, como 5G e LoRaWAN, para garantir conectividade mesmo em regiões remotas. A criação de um banco de dados centralizado com informações contínuas sobre variáveis ambientais poderia fortalecer a gestão sustentável dos recursos hídricos.

- **Avaliação de impacto ambiental e sustentabilidade econômica:**

Estudos futuros poderiam abordar a avaliação do impacto ambiental associado à implementação e operação do sistema, considerando tanto a produção quanto o descarte de componentes tecnológicos. Paralelamente, seria importante avaliar a sustentabilidade econômica em diferentes contextos, como áreas remotas e comunidades com recursos limitados. Essa análise poderia incluir o desenvolvimento de estratégias de financiamento, como parcerias público-privadas e subsídios governamentais, para tornar a tecnologia mais acessível. Além disso, a implementação de soluções modulares e escaláveis poderia atender a diferentes necessidades e capacidades financeiras, promovendo uma adoção mais ampla.

REFERÊNCIAS

AGÊNCIA NACIONAL DE ÁGUAS (ANA). **Dados Abertos**. Disponível em: <https://dados-abertos.ana.gov.br/>. Acesso em: 29 ago. 2024.

AGÊNCIA NACIONAL DE ÁGUAS E SANEAMENTO BÁSICO (ANA). **Plano Nacional de Recursos Hídricos (PNRH)**. Disponível em: <https://www.gov.br/ana/pt-br/assuntos/gestao-das-aguas/planos-de-recursos-hidricos/plano-nacional-de-recursos-hidricos-pnrh>. Acesso em: 30 set. 2024.

ANA. **Sistemas de alerta para inundações**. 2024. Disponível em: <https://progestao.ana.gov.br/destaque-superior/eventos/webinarios/cotas-de-alerta/2-sistemas-de-alerta-para-inundacoes.pdf>. Acesso em: 20 set. 2024.

ANDRADE, E. M. **Hidrologia Básica**. Fortaleza: EdUECE, 2008. 456 p.

ARDUINO. **Arduino Documentation**. Fall 2015. Disponível em: <https://schoolpress.sch.gr/stream/files/2024/01/Arduino.pdf>. Acesso em: 22 out. 2024.

ARDUINO. **Página oficial**. Disponível em: <https://www.arduino.cc/>. Acesso em: 29 set. 2024.

BANZI, M. **Getting Started with Arduino**. Sebastopol: O'Reilly Media, 2011. 130 p.

BANZI, Massimo. **Primeiros passos com o Arduino**. 2011. Disponível em: <https://s3.novatec.com.br/capitulos/capitulo-9788575222904.pdf>. Acesso em: 8 set. 2024.

BARON WEATHER. **The Importance of Early Warning Systems**. 2024. Disponível em: <https://baronweather.com/weather-insights/early-warning-systems-importance>. Acesso em: 8 set. 2024.

BARROS, Flávio Alencar do Rêgo. **Rede de Sensores Sem Fio**. GTA/UFRJ, 2002. Disponível em: https://www.gta.ufrj.br/seminarios/semin2002_1/flavio/. Acesso em: 8 out. 2024.

BOVENSMANN, H.; BURROWS, J. P.; BUCHWITZ, M.; FRERICK, J.; NOËL, S.; ROZANOV, V. V.; CHANCE, K. V.; GOEDE, A. H. **SCIAMACHY: Mission Objectives and Measurement Modes**. Atmospheric Chemistry and Physics, v. 9, n. 2, p. 18-24, 2009.

BRASIL. Constituição (1988). **Constituição da República Federativa do Brasil**. Brasília, DF: Senado Federal, 1988.

BRASIL. Lei nº 11.445, de 5 de janeiro de 2007. **Estabelece diretrizes nacionais para o saneamento básico; altera as Leis nos 6.766, de 19 de dezembro de 1979, 8.036, de 11 de maio de 1990, 8.666, de 21 de junho de 1993, 8.987, de 13 de fevereiro de 1995; revoga a Lei no 6.528, de 11 de maio de 1978; e dá outras providências**. Diário Oficial da União: seção 1, Brasília, DF, 8 jan. 2007.

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, regula o Art. 21 da Constituição Federal e dá outras providências**. Diário Oficial da União: seção 1, Brasília, DF, p. 470, 09 jan. 1997.

BRASIL. Resolução CNRH nº 1, de 17 de fevereiro de 2006. **Dispõe sobre o processo de implementação da Política Nacional de Recursos Hídricos**. Diário Oficial da União, Brasília, DF, 20 fev. 2006.

BRASIL. Resolução CNRH nº 2, de 20 de julho de 2007. **Dispõe sobre o Plano Nacional de Recursos Hídricos**. Diário Oficial da União, Brasília, DF, 24 jul. 2007.

BUYTAERT, W.; ZULKAFI, Z.; GRAINGER, S.; ACOSTA, L.; BIDWELL, V.; DEWULF, A. **Citizen Science in Hydrology and Water Resources: Opportunities for Knowledge Generation, Ecosystem Service Management, and Sustainable Development**. Environmental Science & Policy, New York, v. 54, p. 6-20, 2014.

CARLOS, Marcos de Lima. **História do Arduino**. Disponível em: <https://www.newtonbraga.com.br/index.php/microcontroladores/138-atmel/18422-historia-do-arduino-mic571.html>. Acesso em: 3 out. 2024.

CARVALHO, Fabrício Braga S. de; LEAL, Brauliro G.; FILHO, Jose Valentim dos S.; BAIOCCHI, Orlando R.; LOPES, Waslon T. A.; ALENCAR, Marcelo S. de. **Aplicações Ambientais de Redes de Sensores Sem Fio**. Disponível em: <http://www.univasf.edu.br/~brauliro.leal/pesquisa/Carvalho-1.pdf>. Acesso em: 12 set. 2024.

CARVALHO, Matheus Souza de. **Especificações para sistemas de monitoramento ambiental**. 2016. Disponível em: <https://www.quixada.ufc.br/wp-content/uploads/2017/03/2016-1-ES-MATHEUS-SOUZA-DE-CARVALHO.pdf>. Acesso em: 22 ago. 2024.

CEMADEN. **Anuário da Sala de Situação 2017**. 2017. Disponível em: http://www2.cemaden.gov.br/wp-content/uploads/2020/06/Anuario_Sala_Situa%C3%A7%C3%A3o_2017.pdf. Acesso em: 3 out. 2024.

CONSULTIMER. **O que é LoRa? Conheça a tecnologia de radiofrequência de longo alcance**. Disponível em: [https://consultimer.com/o-que-e-lora-conheca-a-tecnologia-de-radiofrequencia-de-longo-alcance/#:~:text=LoRa%2C%20abrevia%C3%A7%C3%A3o%20de%20%E2%80%9Clongo%20orange,Internet%20das%20Coisas%20\(IoT\)](https://consultimer.com/o-que-e-lora-conheca-a-tecnologia-de-radiofrequencia-de-longo-alcance/#:~:text=LoRa%2C%20abrevia%C3%A7%C3%A3o%20de%20%E2%80%9Clongo%20orange,Internet%20das%20Coisas%20(IoT)). Acesso em: 12 set. 2024.

CUNHA, L. G. **Sensoriamento Remoto e Hidrologia**. São Paulo: Oficina de Textos, 2010. 178 p.

EBAC. **O que é linguagem C?**. Disponível em: <https://ebaonline.com.br/blog/o-que-e-linguagem-c>. Acesso em: 22 ago. 2024.

ENGIE. **Early Warning Systems: Como funciona?**. 2023. Disponível em: <https://www.alemnaenergia.engie.com.br/early-warning-systems-como-funciona/>. Acesso em: 5 out. 2024.

FEITOSA, F. A. C.; MANOEL, M. T. **Hidrogeologia: Conceitos e Aplicações**. São Paulo: CPRM, 2001. 835 p.

GALEGALE, G. P.; SIQUEIRA, É.; HILÁRIO E SILVA, C. B.; SOUZA, C. A. de. **Internet**

das coisas aplicada a negócios – um estudo bibliométrico. *JISTEM - Journal of Information Systems and Technology Management*, 2016. Disponível em: <https://www.scielo.br/j/jistm/a/xVZfWsmzsVY5Tj55YDBDRGG/?format=pdf>. Acesso em: 27 out. 2024.

GERMANO, Stefane Bressani. **Protótipo de solução IoT para uma estação meteorológica aplicando tecnologia LoRa no ambiente agro**. 2022. Disponível em: https://repositorio.ifg.edu.br/bitstream/prefix/1616/1/tcc_Stefane%20Bressani%20Germano.pdf. Acesso em: 3 out. 2024.

GIL, Maria Eduarda Aparecida; HENRIQUE, Thiago Timoteo; TATEOKI, Getúlio Teruo. **Monitoração remota de reservatórios líquidos utilizando o módulo ESP32-LoRa**. 2020. Disponível em: <https://ocs.ifsp.edu.br/conict/xiconict/paper/view/6677/1750>. Acesso em: 18 set. 2024.

GOMES, Flávio Henrique Ferreira; CUNHA, Fernando Nobre; LOPES FILHO, Luiz César; VIDAL, Vitor. **Calibração de um sensor de umidade do solo de baixo custo**. Revista Brasileira de Agricultura Irrigada, v. 11, n. 4, p. 1509-1516, jul. 2017. Disponível em: https://www.researchgate.net/profile/Flavio-Henrique-Ferreira-Gomes/publication/318666453_CALIBRACAO_DE_UM_SENSOR_DE_UMIDADE_DO_SOLO_DE_BAIXO_CUSTO/links/5aec463eaca2727bc003fa1a/CALIBRACAO-DE-UM-SENSOR-DE-UMIDADE-DO-SOLO-DE-BAIXO-CUSTO.pdf. Acesso em: 5 out. 2024.

GOMES, L. L. A.; CAMEIRA, R. F. **Internet of things: conceitos e aplicações baseadas em uma revisão bibliográfica**. In: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, 42., 2022, Curitiba. Anais... Curitiba: ABEPRO, 2018. Disponível em: https://abepro.org.br/biblioteca/TN_STO_258_486_35576.pdf. Acesso em: 27 out. 2024.

INSTITUTO PROPAGUE. **Internet das coisas: conheça a tecnologia que promete revolucionar os serviços financeiros**. Disponível em: https://institutopropague.org/tecnologia-e-dados/internet-das-coisas-conheca-a-tecnologia-que-promete-revolucionar-os-servicos-financeiros/?utm_source=google&utm_medium=cpc&utm_campaign=internet-das-coisas&utm_term=o-que-e&gad_source=1&gclid=CjwKCAjwyfe4BhAWEiwAkIL8sKpWTaa-oYDdMyXVnsN_vAR3IBgpbFYc-Wg0lvSMHlgm_b1DZLKxXhoCr9AQAvD_BwE. Acesso em: 27 out. 2024.

JAVA. **O que é Java?**. Disponível em: https://www.java.com/pt-BR/download/help/whatis_java.html. Acesso em: 15 out. 2024.

JÚNIOR, Cláudio José Martins. **RIVERSENSE: Um sistema para monitoramento de rios através de redes de sensores sem fio**. Disponível em: <https://repositorio.ufsc.br/bitstream/handle/123456789/107449/319595.pdf?sequence=1&isAllowed=y>. Acesso em: 3 out. 2024.

LEAL, G. F. **WaterManna: a IoT na gestão de recursos hídricos**. 2019. Dissertação (Mestrado em Gestão Ambiental) - Universidade Estadual de Maringá, Maringá, 2019. Disponível em: http://repositorio.uem.br:8080/jspui/bitstream/1/5751/1/Gesiane%20Ferreira%20Leal_2019.pdf. Acesso em: 27 out. 2024.

LORA ALLIANCE. **About LoRaWAN**. Disponível em: <https://lora-alliance.org/about-lorawan/>. Acesso em: 22 ago. 2024.

MARQUES, D. M. L. M.; LOUREIRO, S. R. **Hidrologia Aplicada**. São Paulo: Prentice Hall, 2001. 18 p.

MCROBERTS, Michael. **Arduino Básico - 2ª edição: tudo sobre o popular microcontrolador Arduino**. 2015. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=EqOoBwAAQBAJ&oi=fnd&pg=PA6&dq=Arduino&ots=DfdC47vE85&sig=qox7Kz83TXPgcC0vnXZOHBVh-ug#v=onepage&q&f=false>. Acesso em: 15 ago. 2024.

MDR. **Sistema Nacional de Gerenciamento de Recursos Hídricos**. Disponível em: <https://www.gov.br/mdr/pt-br/assuntos/seguranca-hidrica/cnrh/sistema-nacional-de-gerenciamento-de-recursos-hidricos>. Acesso em: 5 set. 2024.

MENDES, Paulo Cesar de Souza. **Caracterização de um sensor para medição de umidade do solo com termo-resistor a temperatura constante**. 2006. Disponível em: <http://www.ppgee.eng.ufba.br/teses/ac6732a9863784f9755479cd130662e4.pdf>. Acesso em: 3 out. 2024.

MENDES, Reisila Simone Migliorini. **Estudo dos indicadores de qualidade do solo para recuperação de nascentes em região de cerrado mineiro**. 2017. Disponível em: <https://repositorio.ufmg.br/handle/1843/35306>. Acesso em: 10 set. 2024.

MINISTÉRIO DAS CIDADES. **Gestão de recursos hídricos ganhará reforço com acordos internacionais**. Disponível em: <https://www.gov.br/cidades/pt-br/noticias/anterior/gestao-de-recursos-hidricos-ganhara-reforco-com-acordos-internacionais>. Acesso em: 22 ago. 2024.

MINISTÉRIO DO DESENVOLVIMENTO REGIONAL. **Plano Nacional de Recursos Hídricos**. Disponível em: <https://www.gov.br/mdr/pt-br/assuntos/seguranca-hidrica/plano-nacional-de-recursos-hidricos-1/o-pnrh>. Acesso em: 15 out. 2024.

MINISTÉRIO PÚBLICO FEDERAL. **Principais acordos e tratados assinados pelo Brasil com interferência em recursos hídricos**. Disponível em: <https://conexaogua.mpf.mp.br/acervo/acordos/principais-acordos-e-tratados-assinados-pelo-brasil-com-interferencia-em-recursos-hidricos>. Acesso em: 18 set. 2024.

NEVES, M. A. T. **Internet das coisas (IoT): introdução e visão geral de aplicações**. 2021. Disponível em: <https://acervodigital.ufpr.br/xmlui/bitstream/handle/1884/70452/MATEUS-APARECIDO-TONIN-NEVES.pdf?sequence=1&isAllowed=y>. Acesso em: 27 out. 2024.

OLIVEIRA, C. A. N.; CONTI, D. M. **Sistema de automação para prevenção em áreas de risco de enchentes em cidades inteligentes: estudo de caso em São Carlos, SP**. In: SIMPÓSIO INTERNACIONAL DE GESTÃO DE PROJETOS, INOVAÇÃO E SUSTENTABILIDADE, 7., 2019, São Paulo. Anais... São Paulo: UNINOVE, 2019. Disponível em: <https://www.singep.org.br/7singep/resultado/84.pdf>. Acesso em: 27 out. 2024.

OLIVEIRA, Márcio Clei Silva de. **Estudo sobre sistemas de análise e monitoramento de recursos hídricos com tecnologias de código aberto: caso do projeto Mãe D'Água da Rede InfoAmazonia**. Disponível em: https://ppgca.unifap.br/wp-content/uploads/2021/07/Dissertacao_Mestrado_Marcio-Clei_2019.pdf. Acesso em: 20 ago. 2024.

OPR. **Parceria internacional desenvolve sistema de alerta para o risco de brusone no trigo.** 2024. Disponível em: <https://opresenterural.com.br/parceria-internacional-desenvolve-sistema-de-alerta-para-o-risco-de-brusone-no-trigo/>. Acesso em: 12 ago. 2024.

SANTOS, Alberto C. P.; MAGALHÃES, Amarilton L.; ALMEIDA, André L. F. de; MACIEL, Tarcisio F. **Integração de redes de sensores sem fio à plataforma LoRa® para serviços de monitoramento e controle inteligentes.** 2020. Disponível em: <https://www.sbrt.org.br/sbrt2020/papers/1570661669.pdf>. Acesso em: 25 ago. 2024.

SANTOS, B. P.; SILVA, L. A. M.; CELES, C. S. F. S.; BORGES NETO, J. B.; PERES, B. S.; VIEIRA, M. A. M.; VIEIRA, L. F. M.; GOUSSEVSKAIA, O. N.; LOUREIRO, A. A. F. **Internet das coisas: da teoria à prática.** Disponível em: <https://homepages.dcc.ufmg.br/~mmvieira/cc/papers/internet-das-coisas.pdf>. Acesso em: 27 out. 2024.

SANTOS, Renato Batista. **Monitoramento de hidrômetro utilizando tecnologia LoRa para comunicação.** 2022. Disponível em: <https://repositorio.utfpr.edu.br/jspui/bitstream/1/30718/1/monitoramentohidrometrolora.pdf>. Acesso em: 5 out. 2024.

SIGMA SENSORS. **Sensor de umidade do solo.** Disponível em: <https://sigmasensors.com.br/sensor-de-umidade-do-solo#:~:text=Um%20Sensor%20de%20Umidade%20do,medi%C3%A7%C3%A3o%20da%20umidade%20da%20terra>. Acesso em: 12 set. 2024.

SILVA, A. M. **Gestão de Recursos Hídricos.** São Paulo: Editora Manole, 2010. 247 p.

SILVA, Daniela Rocha; SOBRAL, Luis Felipe Bentin. **Um estudo em larga escala sobre a estrutura do código-fonte de pacotes JavaScript.** 2017. Disponível em: <https://bsi.uniriotec.br/wp-content/uploads/sites/31/2020/05/201707DanielaRochaLuisSobral.pdf>. Acesso em: 10 set. 2024.

SILVA, Marcel Salvioni da; FRUETT, Fabiano. **Monitoramento Ambiental através de Rede de Sensores Sem Fio de Baixo Custo.** Disponível em: <http://marte2.sid.inpe.br/col/dpi.inpe.br/marte2/2013/05.29.01.13/doc/p1706.pdf>. Acesso em: 5 set. 2024.

SMITH, K.; WARD, R. **Floods: Physical Processes and Human Impacts.** New York: Wiley, 1998. 955 p.

SOARES, Natália Cristino. **Sistema para monitoramento de nascentes e cursos de água via SMS.** Disponível em: https://www.monografias.ufop.br/bitstream/35400000/1209/1/MONOGRAFIA_SistemaMonitoramentoNascentes.pdf. Acesso em: 29 set. 2024.

SOUZA, Saymon Castro de; VIANNA, Leonardo Félix; GUTTER, Reginaldo Radinz; GARBERCHT, Andreia Lucia; BALDEZ, Victor Cristian de Aguiar. **Uma infraestrutura para monitoramento de rios para predição de enchentes.** Disponível em: <https://www.aedb.br/seget/arquivos/artigos17/18125266.pdf>. Acesso em: 25 ago. 2024.

TERRA AMBIENTAL. **O que é a Política Nacional de Recursos Hídricos e a sua importância para o meio ambiente.** Disponível em: <https://www.teraambiental.com.br/blog-da-tera-ambiental/o-que-e-a-politica-nacional-de-recursos-hidricos-e-a-sua-importancia-para-o-meio-ambiente>. Acesso em: 12 out. 2024.

TUCCI, C. E. M. **Hidrologia: Ciência e Aplicação.** Porto Alegre: Ed. da UFRGS, 2007. 938 p.

UNDP. **MHEWS: Multi-Hazard Early Warning Systems Framework.** 2022. Disponível em: https://www.undp.org/sites/g/files/zskgke326/files/2024-02/mhewas_framework_2022_pt.pdf. Acesso em: 25 ago. 2024.

VITAL S. JÚNIOR, F.; CUNHA, M. X. C.; VIEIRA, L. B. **IoTflood: uma solução baseada em internet das coisas para monitoramento de enchentes de grandes proporções em tempo real.** Disponível em: <https://www2.ifal.edu.br/ppgtec/publicacoes/arquivos/publicacoes-2020/iotflood-uma-solucao-baseada-em-internet-das-coisas-para-monitoramento-de-enchentes-de-grandes-proporcoes-em-tempo-real.pdf>. Acesso em: 27 out. 2024.

WARD, R. C.; ROBINSON, M. **Principles of Hydrology.** 4th ed. New York: McGraw-Hill, 1990. 402 p.

BIBLIOGRAFIA CONSULTADA

BERTOLETI, Pedro. **Projetos com ESP32 e LoRa**. 2019. Disponível em: <https://books.google.com.br/books?id=Doi0DwAAQBAJ>. Acesso em: 19 ago. 2024.

CALMON, Ana Teresa; MÜLLER, Mauricio; BAUNGARTEN, Ana Paula; FELIX, Daniele; MARTINS, Marlo Antonio Ribeiro. **Tecnologia de mapeamento digital a laser e suas aplicações em hidrologia urbana**. Disponível em: <https://files.abrhidro.org.br/Eventos/Trabalhos/154/317.pdf>. Acesso em: 10 set. 2024.

CEMADEN. **Concurso público para provimento de vagas em cargos de nível superior da carreira de desenvolvimento tecnológico**. Edital de concurso público nº 02/2023 – CEMADEN/MCTI, de 09 de outubro de 2023. Disponível em: <https://www.novaconcursos.com.br/portal/wp-content/uploads/2023/12/tecnologia.pdf>. Acesso em: 12 ago. 2024.

GOULART, Victor. **Sistema IoT para monitoramento de porteiros utilizando LoRa e LoRaWAN**. 2018. Disponível em: <https://publicacao.cimatech.com.br/index.php/cimatech/article/view/108>. Acesso em: 15 out. 2024.

INTERNET SOCIETY. **The Internet Of Things: An Overview**. 2015. Disponível em: https://d1wqtxts1xzle7.cloudfront.net/48790442/ISOC-IoT-Overview-20151014_0-libre.pdf. Acesso em: 3 out. 2024.

KOBIYAMA, Masato; MONTEIRO, Leonardo Romero; GOERL, Roberto Fabris. **Integração das ciências e das tecnologias para redução de desastres naturais: sócio-hidrologia e sócio-tecnologia**. 2018. Disponível em: <https://lume.ufrgs.br/handle/10183/184477>. Acesso em: 30 ago. 2024.

LIMA, João L. M. P. de; LIMA, M. Isabel P. de; RODRIGUES, Rui. **Hidrologia**. 2019. Disponível em: <https://ap1.sib.uc.pt/handle/10316.2/45879>. Acesso em: 1 out. 2024.

MARTINS, Mário Henrique da Mata; SPINK, Mary Jane Paris. **O uso de tecnologias de comunicação de riscos de desastres como prática preventiva em saúde**. 2015. Disponível em: <https://www.scielo.br/j/icse/a/5c6srmRzK3wWk5YBCM6qNCm/>. Acesso em: 14 set. 2024.

NUNES, Eduardo Elias; SOUZA, Rodrigo Alencar de; COSTA, Arlindo Messias Mendes da; AMARAL, Hiram Carlos Costa; REZENDE, Diogo Pena; MOURÃO, Andreza. **IoT Real-Time Soil Monitoring Based on LoRa for Palm Oil Plantation**. 2020. Disponível em: https://lcv.fee.unicamp.br/images/BTSym-22-Brasil/papers/BTSym2022_032.pdf. Acesso em: 25 set. 2024.

QUIMBITA, Michael Andrés Moya. **Evaluación de pasarela LoRa/LoRaWAN en entornos urbanos**. 2018. Disponível em: <https://riunet.upv.es/handle/10251/109791>. Acesso em: 5 out. 2024.

ROSA, Alexandre. **INTERNET OF THINGS (IOT) Aplicações, desafios, tendências**. Março/2020. Disponível em: https://edisciplinas.usp.br/pluginfile.php/5199901/mod_resource/content/1/IoT%20-%20Alexandre%20Rosa%20-%20Angoera.pdf. Acesso em: 17 set. 2024.

SALEH, Muhammad Syarafuddin Md; SAJAK, Aznida Abu Bakar; MOHAMAD, Rafidei; ZAABA, Mohd Ariff Majmi. **LoRa and server-based home automation using the internet of things (IoT)**. 2020. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1319157820306285>. Acesso em: 8 out. 2024.

SANTOS, Renato Batista. **Monitoramento de hidrômetro utilizando tecnologia LoRa para comunicação**. 2022. Disponível em: https://bdtd.ibict.br/vufind/Record/UTFPR-12_a09734989d428cabd3147c554c4fdcf4. Acesso em: 3 set. 2024.

SAUSEN, Tania Maria; LACRUZ, María Silvia Pardi. **Sensoriamento remoto para desastres**. 2015. Disponível em: <https://books.google.com.br/books?id=Y0EuCQAAQBAJ>. Acesso em: 22 out. 2024.

SHAHIDIAN, Shakib; GUIMARÃES, Rita; RODRIGUES, Carlos. **Hidrologia Agrícola**. 2012. Disponível em: <https://dspace.uevora.pt/rdpc/handle/10174/7224>. Acesso em: 21 set. 2024.

WILKE, Humberto; LORA, Electo Eduardo Silva. **Desenvolvimento de um módulo combustor biomassa-motor stirling aplicado a sistemas de geração isolada e baseados em gerador de indução**. 2004. Disponível em: http://www.proceedings.scielo.br/scielo.php?pid=MSC0000000022004000100046&script=sci_arttext&tlng=pt. Acesso em: 30 set. 2024.

APÊNDICE A – Programação do Sistema Virtual Tinkercad

/*

The LCD Circuit:

- * LCD RS pin to digital pin 12
- * LCD Enable pin to digital pin 11
- * LCD D4 pin to digital pin 5
- * LCD D5 pin to digital pin 4
- * LCD D6 pin to digital pin 3
- * LCD D7 pin to digital pin 6
- * LCD R/W pin to ground
- * LCD VSS pin to ground
- * LCD VCC pin to 5V
- * 10K resistor:
- * ends to +5V and ground
- * wiper to LCD VO pin (pin 3)

*/

// ### BIBLIOTECAS ###

#include <LiquidCrystal.h>

// ### DEFINIÇÃO DA POSIÇÃO DOS PINOS

#define waterPinVcc A0 // Sensor de Água (Chuva) conectado ao Pino A0 do Arduino

#define waterPinSignal A1 // Sensor de Água (Chuva) conectado ao Pino A1 do Arduino

#define hallPin 2 // Sensor de Efeito Hall (Vazão) conectado ao Pino D2 do Arduino

#define buzzerPin 8 // Sirene conectada ao Pino D8 do Arduino

#define echoPin 9 // Retorno (Eco) do Sensor de Distância Ultrassônico (HC-SR04) conectado ao Pino D9 do Arduino

```

#define triggerPin 10 // Disparo do Sensor de Distância Ultrassônico (HC-SR04) conectado
ao Pino D10 do Arduino

// Inicializa com os Pinos do LCD

LiquidCrystal lcd(12, 11, 5, 4, 3, 6);

// Para Cálculo da vazão

#define PI 3.1415926535897932384626433832795

#define intervaloPulso 500

// Tempo de apresentação/delay no LCD e Serial

#define apresentaResultado 2000

// ### DECLARAÇÕES DAS VARIÁVEIS###

// Nível do Rio

unsigned long duracao; // Tempo para o eco

float distancia; // Distância calculada

// Intensidade da Chuva

unsigned int intensidadeChuva; // Valor da Intensidade

// Vazão do Rio

const float diametroEixo = 0.15; // Diâmetro da roda em [m]

volatile unsigned long pulso = 0; // Número de voltas/pulsos lidos pelo Arduino

unsigned long ultimaVerificacao = 0; // Millis da Última Verificação

unsigned long proximaVerificacao = 0; // Millis da Próxima Verificação

unsigned long varTempo = 0; // Diferença entre o Millis Atual a última verificação

float ajusteEncoder = 185.21; // 313 rpm a 12V (conforme configurado no motor) / 1.69
pulsos

```

```
float rpm = 0;           // Frequência em [RPM] = pulsos/voltas por minuto
float Hz = 0;           // Frequencia em [Hz = 1/s] = pulsos/voltas por segundo
float Periodo = 0;      // Período(T) em [m/s]
float kmh = 0;          // Velocidade em [m/s]

// Demais

unsigned int tamanhoChar; // Tamanho do número a ser apresentado no LCD
unsigned int cont=0;      // Contador de medições

// Trigger

// 0 - sem medição
// 1 - sem risco
// 2 - baixo risco
// 3 - médio risco
// 4 - alto risco
// 5 - inundação

// Structure para análise
struct Tipo
{
    float Valor;
    String Descritivo;
    int Grau;
};
```

```

// #### MEDIÇÃO DO NÍVEL DO RIO ####

struct Tipo NivelRio()
{
    // Declara o Struct com as chaves que deverão
    // receber os devidos valores

    Tipo T;

    // Retorno o pino de disparo do Sensor para posição inicial
    digitalWrite(triggerPin, LOW);

    delayMicroseconds(2);

    // Aciona o pino de disparo (HIGH) por 10 ms
    digitalWrite(triggerPin, HIGH);

    delayMicroseconds(10);

    digitalWrite(triggerPin, LOW);

    // Efetua a leitura do pino de Retorno (Eco)

    // Retorna o tempo que a onda sonora viajou até retornar ao Sensor
    duracao = pulseIn(echoPin, HIGH); // em milissegundos

    // Calculando a distância em cm

    // Velocidade do Som (343 m/s) dividido por 2 (tempo de ida e volta)

    distancia = duracao * 0.03549 / 2; // pequeno ajuste de ambiente para adequar aos dados apre-
    sentados no simulador

```

```
// Tradutor

T.Valor = distancia;

switch(int(T.Valor))
{
    case 0 ... 10:

        T.Descritivo = "Inundacao";

        T.Grau = 5;

        break;

    case 11 ... 50:

        T.Descritivo = "Nivel Alto";

        T.Grau = 4;

        break;

    case 51 ... 100:

        T.Descritivo = "Nivel Medio";

        T.Grau = 3;

        break;

    case 101 ... 150:

        T.Descritivo = "Nivel Baixo";

        T.Grau = 2;

        break;

    case 151 ... 250:

        T.Descritivo = "Nivel Mto Baixo";

        T.Grau = 1;

        break;
```

```
        default:
            T.Descritivo = "Nao Medido";
            T.Grau = 0;
            break;
    }

//Validador
Serial.print("Nivel do Rio: ");
Serial.print(T.Descritivo);
Serial.print(" (");
Serial.print(T.Valor);
Serial.println(" cm)");

return T;
}

// ### MEDIÇÃO DA VAZÃO DO RIO ###
// Tratamento do Pino 2 (Interrupção 0)
void contaPulso()
{
    pulso++;
}
```

```

//Método do Cálculo da Vazão (Velocidade)

struct Tipo VazaoRio()

{
    // Declara o Struct com as chaves que deverão
    // receber os devidos valores

    Tipo T;

    if (millis() >= proximaVerificacao)
    {
        varTempo = millis() - ultimaVerificacao;

        noInterrupts(); //Don't process interrupts during calculations

        rpm = ajusteEncoder * pulso / varTempo;

        Hz = rpm * 0.0166666666667;

        Periodo = 1 / Hz;

        kmh = 2 * PI * (diametroEixo/2) * Hz * 3.6;

        pulso = 0L;

        interrupts(); // Restart the interrupt processing

        ultimaVerificacao = millis();

        proximaVerificacao = ultimaVerificacao + intervaloPulso;
    }

    // Tradutor

    T.Valor = kmh;

    switch((int)T.Valor)
    {

```

case 0 ... 3:

T.Descritivo = "Estacionado";

T.Grau = 0;

break;

case 4 ... 6:

T.Descritivo = "Muito Baixa Vazao";

T.Grau = 1;

break;

case 7 ... 9:

T.Descritivo = "Baixa Vazao";

T.Grau = 2;

break;

case 10 ... 12:

T.Descritivo = "Media Vazao";

T.Grau = 3;

break;

case 13 ... 15:

T.Descritivo = "Alta Vazao";

T.Grau = 4;

break;

default:

T.Descritivo = "Turbulencia";

T.Grau = 5;

break;

}


```

// Validador

Serial.print("Vazao do Rio: ");

Serial.print(T.Descritivo); //frequência em RPM

Serial.print(" (f=");

Serial.print(rpm); //frequência em RPM

Serial.print(" RPM ou ");

Serial.print(Hz); //frequência em Hertz (SI)

Serial.print(" Hz | T=");

Serial.print(Periodo); // período em s (SI)

Serial.print(" s | v=");

Serial.print(kmh); // velocidade em km/h

Serial.print(" km/h");

Serial.println(")");

return T;
}

// ### MEDIÇÃO DA INTENSIDADE DA CHUVA ###

struct Tipo IntensidadeChuva()

{

// Declara o Struct com as chaves que deverão

// receber os devidos valores

Tipo T;

```

```
// Ativa a energia no Sensor de Chuva (estamos utilizando
// o sensor de umidade do solo no Tinkercad para efeito
// didático)
digitalWrite(A0, HIGH);

// Aguarda por 10ms
delay(10);

// Lê o valor registrado pelo Sensor
intensidadeChuva = analogRead(A1);

// Desliga o sensor para reduzir a corrosão do metal
// ao longo do tempo
digitalWrite(A0, LOW);

// Tradutor
T.Valor = intensidadeChuva;
switch((int)T.Valor)
{
  case 0 ... 164:
    T.Descritivo = "Sem chuva";
    T.Grau = 0;
    break;
  case 165 ... 327:
    T.Descritivo = "Forte Nevoeiro";
    T.Grau = 1;
```

```
        break;
    case 328 ... 511:
        T.Descritivo = "Chuvisco";
        T.Grau = 2;
        break;
    case 512 ... 700:
        T.Descritivo = "Chuva Mediana";
        T.Grau = 3;
        break;
    case 701 ... 800:
        T.Descritivo = "Chuva Forte";
        T.Grau = 4;
        break;
    default:
        T.Descritivo = "Tempestade";
        T.Grau = 5;
        break;
}

// Validador
Serial.print("Intensidade da Chuva: ");
Serial.print(T.Descritivo);
Serial.print(" (");
Serial.print(T.Valor);
Serial.println(")");
```

```
return T;
}

// ### AFERIÇÃO DO RISCO DE ALAGAMENTO ###

struct Tipo Risco(int Nivel, int Chuva, int Vazao)
{
    Tipo T;

    T.Grau = (3*Nivel) + (1*Chuva) + (2*Vazao);

    T.Descritivo = "Nao Medido";

    // Resposta

    if(Nivel == 5)
    {
        T.Descritivo = "Inundacao";
    }

    else if(T.Grau>=0 && T.Grau<=5)
    {
        T.Descritivo = "Nao Medido";
    }

    else if(T.Grau>5 && T.Grau<=9)
    {
        T.Descritivo = "Baixo Risco";
    }

    else if(T.Grau>9 && T.Grau<=13)
```

```
{  
    T.Descritivo = "Medio Risco";  
}  
else if(T.Grau>13 && T.Grau<=16)  
{  
    T.Descritivo = "Alto Risco";  
}  
else  
{  
    T.Descritivo = "Inundacao";  
}  
//status=status + " (" + String(resultado) + ")";  
  
return T;  
}  
void setup()  
{  
    // Inicializa comunicação Serial:  
    Serial.flush();  
    Serial.begin(9600);  
  
    // Sensor de Chuva (estamos utilizando o sensor  
    // de umidade do solo no Tinkercad para efeito didático)  
    pinMode(waterPinVcc, OUTPUT);  
    pinMode(waterPinSignal, INPUT);
```

```
// Sirene de Alerta Local

pinMode(buzzerPin,OUTPUT);

// Sensor do Nível do Rio

pinMode(echoPin, INPUT); // Retorno (Echo)

pinMode(triggerPin, OUTPUT); // Disparo

// Sensor de Efeito Hall (Encoder)

pinMode(hallPin, INPUT); // Configura o pino D2 (disparo) como INPUT

attachInterrupt(0, contaPulso, FALLING); // Interrupção por borda de descida no pino
2 (interrupção 0)

proximaVerificacao = ultimaVerificacao + millis(); //Inicia Contagem Tempo

// Configura o layout do LCD (16 colunas x 2 linhas):

lcd.begin(16, 2);

// Imprime mensagem inicial no LCD.

lcd.setCursor(0, 0);

lcd.print("BEM VINDO!!!");

lcd.setCursor(0, 1);

lcd.print("INICIANDO...");

delay(1500);

lcd.clear();

}
```

```

void loop()
{
  // ### MARCA QUANTOS LOOPINGS/MEDIÇÕES JÁ SE PASSARAM ###
  // Pula uma linha para leitura ficar mais visual
  Serial.println("=====");
  // Inicia marcação
  Serial.print("Medicao #");
  Serial.print(cont++);
  Serial.println(":");

  // ### APRESENTA O NÍVEL DO RIO NO DISPLAY
  Tipo medicaoNivelRio;
  medicaoNivelRio = NivelRio();
  lcd.setCursor(0, 0);
  lcd.print("N");
  lcd.setCursor(1, 0);
  lcd.print(char(161)); //="í"
  lcd.setCursor(2, 0);
  lcd.print("vel do rio:");
  lcd.setCursor(0, 1);
  lcd.print(medicaoNivelRio.Descritivo);
  /*
  lcd.print(medicaoNivelRio.Valor);

```

```
tamanhoChar = String(medicacaoNivelRio.Valor).length();  
lcd.setCursor(tamanhoChar, 1);  
lcd.print(" cm");  
lcd.setCursor(tamanhoChar + 3,1);  
lcd.print(" ");  
*/  
delay(apresentaResultado);  
lcd.clear();  
  
// ### APRESENTA A INTENSIDADE DA CHUVA NO DISPLAY ###  
Tipo medicacaoIntensidadeChuva;  
medicacaoIntensidadeChuva = IntensidadeChuva();  
lcd.setCursor(0, 0);  
lcd.print("Intens. da Chuva");  
lcd.setCursor(0, 1);  
lcd.print(medicacaoIntensidadeChuva.Descritivo);  
delay(apresentaResultado);  
lcd.clear();  
  
///  
// # APRESENTA A VAZÃO DO RIO NO DISPLAY  
Tipo medicacaoVazaoRio;  
medicacaoVazaoRio = VazaoRio();  
lcd.setCursor(0, 0);
```



```

lcd.print("Vazao do rio:");

lcd.setCursor(0, 1);

lcd.print(medicaoVazaoRio.Descritivo);

/*

lcd.print(medicaoVazaoRio.Valor);

tamanhoChar = String(medicaoVazaoRio.Valor).length();

lcd.setCursor(tamanhoChar, 1);

lcd.print(" km/h");

lcd.setCursor(tamanhoChar + 5,1);

lcd.print("  ");

*/

delay(apresentaResultado);

lcd.clear();

//#ACIONA O BUZZER PARA ALERTA LOCAL

Tipo Resultado;

Resultado.Descritivo = "Não Medido";

Resultado = Risco(medicaoNivelRio.Grau, medicaoIntensidadeChuva.Grau, medicaoVazao-
Rio.Grau);

digitalWrite(buzzerPin,LOW);

if(Resultado.Descritivo=="Alagado" || Resultado.Descritivo=="Alto Risco")

{

    digitalWrite(buzzerPin,HIGH);

}

```

```
else

{

    digitalWrite(buzzerPin,LOW);

}

//#APRESENTA O RESULTADO FINAL NO DISPLAY

//Validador

Serial.print("Resultado Final: ");

Serial.print(Resultado.Descritivo);

Serial.print(" (");

Serial.print(Resultado.Grau);

Serial.print(") [Nivel=");

Serial.print(medicaoNivelRio.Grau);

Serial.print(" | Chuva=");

Serial.print(medicaoIntensidadeChuva.Grau);

Serial.print(" | Vazao=");

Serial.print(medicaoVazaoRio.Grau);

Serial.println("]");

//LCD

lcd.setCursor(0, 0);

lcd.print("Cenario Consol.");

lcd.setCursor(0, 1);

lcd.print(Resultado.Descritivo);

delay(apresentaResultado);
```

```
lcd.clear();  
}
```

APÊNDICE B – Código Arduino e Sensores do Sistema Físico

```

#include <Ultrasonic.h>

// Pinos do sensor ultrassônico
#define TRIG_PIN 3
#define ECHO_PIN 4

// Pino do sensor de fluxo
#define FLOW_SENSOR_PIN 7

// Inicializa o sensor ultrassônico
Ultrasonic ultrasonic(TRIG_PIN, ECHO_PIN);

// Altura máxima do recipiente
#define ALTURA_MAXIMA_RECIPIENTE 147 // altura atualizada para 147 cm

// Variáveis para o fluxo de água
int pulseCount = 0;
unsigned long previousMillis = 0;
float fluxoAgua = 0.0;

void setup() {
  Serial.begin(9600);
  pinMode(FLOW_SENSOR_PIN, INPUT);
  Serial.println("Iniciando medição do nível e fluxo de água...");
}

void loop() {
  unsigned long currentMillis = millis();

  // =====
  // Medição do Nível de Água
  // =====
  if (currentMillis - previousMillis >= 2000) { // Intervalo de 2 segundos para atualização
    previousMillis = currentMillis;

    // Lê a distância medida pelo sensor ultrassônico e ajusta a leitura
    float distancia = ultrasonic.read() + 1; // Ajuste de 1 cm para compensação de leitura
    float alturaAgua = ALTURA_MAXIMA_RECIPIENTE - distancia;

    // Garante que o valor da altura da água nunca seja negativo
    if (alturaAgua < 0) {
      alturaAgua = 0;
    }

    // Calcula a porcentagem de enchimento
    float porcentagemAgua = (alturaAgua / ALTURA_MAXIMA_RECIPIENTE) * 100;

    // Cálculo do fluxo em L/min (pulsos por segundo / 7.5)

```

```

fluxoAgua = pulseCount / 7.5;

// Reseta a contagem de pulsos
pulseCount = 0;

// =====
// Exibição dos Dados no Monitor Serial
// =====
Serial.print("Altura da água: ");
Serial.print(alturaAgua);
Serial.println(" cm");

Serial.print("Porcentagem de água: ");
Serial.print(porcentagemAgua);
Serial.println(" %");

Serial.print("Fluxo de água: ");
Serial.print(fluxoAgua);
Serial.println(" L/min");

// Envia os dados via Serial no formato "altura,porcentagem,fluxo"
Serial.print(alturaAgua);
Serial.print(",");
Serial.print(porcentagemAgua);
Serial.print(",");
Serial.println(fluxoAgua);
}

// =====
// Contagem de Pulsos para o Sensor de Fluxo
// =====
if (digitalRead(FLOW_SENSOR_PIN) == HIGH) {
  while (digitalRead(FLOW_SENSOR_PIN) == HIGH); // Aguarda o pulso terminar
  pulseCount++;
}
}

```

APÊNDICE C – Código Módulo LoRa Transmissor

```

#include <SPI.h>
#include <LoRa.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

// Definir os pinos LoRa de acordo com o TTGO LoRa32
#define SCK 5
#define MISO 19
#define MOSI 27
#define SS 18
#define RST 14
#define DI0 26
#define OLED_RST -1 // Usar -1 para desativar o reset de hardware da tela OLED

// Definições para a tela OLED
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RST);

// Variáveis para armazenar o último valor recebido
float ultimaAlturaAgua = 0.0;
float ultimaPorcentagemAgua = 0.0;
float ultimoFluxoAgua = 0.0;

// Variáveis para controle do tempo (1 segundo de intervalo)
unsigned long tempoAnterior = 0;
const long intervalo = 1000; // 1 segundo

void setup() {
  Serial.begin(9600);
  while (!Serial);

  // Inicializa o display OLED
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("Falha ao iniciar a tela OLED"));
    while (1);
  }

  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(0, 0);
  display.println("LoRa Receiver Pronto");
  display.display();

  // Inicializa o LoRa
  SPI.begin(SCK, MISO, MOSI, SS);

```

```

LoRa.setPins(SS, RST, DI0);

if (!LoRa.begin(915E6)) { // Frequência de operação 915 MHz
  Serial.println("Falha ao iniciar o LoRa!");
  display.clearDisplay();
  display.setCursor(0, 0);
  display.println("LoRa Init Failed");
  display.display();
  while (1);
}
}

void loop() {
  // Verifica se há pacotes de dados via LoRa
  int packetSize = LoRa.parsePacket();
  if (packetSize) {
    String receivedData = "";

    // Ler o pacote de dados recebido via LoRa
    while (LoRa.available()) {
      receivedData += (char)LoRa.read();
    }

    // Extrair os valores de altura, porcentagem e fluxo dos dados recebidos
    int indexOfFirstComma = receivedData.indexOf(',');
    int indexOfSecondComma = receivedData.indexOf(',', indexOfFirstComma + 1);

    if (indexOfFirstComma > 0 && indexOfSecondComma > indexOfFirstComma) {
      String alturaAguaStr = receivedData.substring(0, indexOfFirstComma);
      String porcentagemAguaStr = receivedData.substring(indexOfFirstComma + 1,
        indexOfSecondComma);
      String fluxoAguaStr = receivedData.substring(indexOfSecondComma + 1);

      // Atualizar os valores somente se o pacote foi lido corretamente
      ultimaAlturaAgua = alturaAguaStr.toFloat();
      ultimaPorcentagemAgua = porcentagemAguaStr.toFloat();
      ultimoFluxoAgua = fluxoAguaStr.toFloat();
    }
  }

  // Obter o tempo atual
  unsigned long tempoAtual = millis();

  // Verifica se 1 segundo se passou
  if (tempoAtual - tempoAnterior >= intervalo) {
    tempoAnterior = tempoAtual;

    // Exibir os dados no Monitor Serial e no OLED
    Serial.print("Altura da água: ");
    Serial.print(ultimaAlturaAgua, 2); // 2 casas decimais
  }
}

```

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

Serial.print("Porcentagem de água: ");
Serial.print(ultimaPorcentagemAgua, 2); // 2 casas decimais
Serial.println(" %");

Serial.print("Fluxo de água: ");
Serial.print(ultimoFluxoAgua, 2); // 2 casas decimais
Serial.println(" L/min");

// Exibe os dados no OLED
display.clearDisplay();
display.setCursor(0, 0);
display.print("Altura da agua: ");
display.print(ultimaAlturaAgua, 2); // 2 casas decimais
display.println(" cm");
display.print("Porcentagem: ");
display.print(ultimaPorcentagemAgua, 2); // 2 casas decimais
display.println(" %");
display.print("Fluxo: ");
display.print(ultimoFluxoAgua, 2); // 2 casas decimais
display.println(" L/min");
display.display();
}
}
```


APÊNDICE D – Código Módulo LoRa Receptor

```

#include <SPI.h>
#include <LoRa.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

// Definir os pinos LoRa de acordo com o TTGO LoRa32
#define SCK 5
#define MISO 19
#define MOSI 27
#define SS 18
#define RST 14
#define DI0 26
#define OLED_RST -1 // Usar -1 para desativar o reset de hardware da tela OLED

// Definições para a tela OLED
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RST);

// Variáveis para armazenar o último valor recebido
float ultimaAlturaAgua = 0.0;
float ultimaPorcentagemAgua = 0.0;
float ultimoFluxoAgua = 0.0;

// Variáveis para controle do tempo (1 segundo de intervalo)
unsigned long tempoAnterior = 0;
const long intervalo = 1000; // 1 segundo

void setup() {
  Serial.begin(9600);
  while (!Serial);

  // Configuração do LoRa
  SPI.begin(SCK, MISO, MOSI, SS);
  LoRa.setPins(SS, RST, DI0);

  if (!LoRa.begin(915E6)) { // Frequência de operação 915 MHz
    Serial.println("Falha ao iniciar o LoRa!");
    while (1);
  }

  // Inicializa o display OLED
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println("Falha ao iniciar a tela OLED");
    while (1);
  }
}

```

```

display.clearDisplay();
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(0, 0);
display.println("LoRa Sender Pronto");
display.display();
}

void loop() {
  // Verifica se há dados recebidos do Arduino via serial
  if (Serial.available() > 0) {
    // Lê os dados enviados pelo Arduino (ex: "alturaAgua,porcentagemAgua,fluxoAgua")
    String dataFromArduino = Serial.readStringUntil('\n');

    // Extrair os valores de altura, porcentagem e fluxo dos dados recebidos
    int indexOfFirstComma = dataFromArduino.indexOf(',');
    int indexOfSecondComma = dataFromArduino.indexOf(',', indexOfFirstComma + 1);

    String alturaAguaStr = dataFromArduino.substring(0, indexOfFirstComma);
    String porcentagemAguaStr = dataFromArduino.substring(indexOfFirstComma + 1,
    indexOfSecondComma);
    String fluxoAguaStr = dataFromArduino.substring(indexOfSecondComma + 1);

    // Verificar se os dados recebidos são válidos
    if (alturaAguaStr.length() > 0 && porcentagemAguaStr.length() > 0 &&
    fluxoAguaStr.length() > 0) {
      ultimaAlturaAgua = alturaAguaStr.toFloat();
      ultimaPorcentagemAgua = porcentagemAguaStr.toFloat();
      ultimoFluxoAgua = fluxoAguaStr.toFloat(); // Adiciona o fluxo de água
    }

    // Envia os dados via LoRa
    LoRa.beginPacket();
    LoRa.print(String(ultimaAlturaAgua) + "," + String(ultimaPorcentagemAgua) + "," +
    String(ultimoFluxoAgua));
    LoRa.endPacket();
  }

  // Obter o tempo atual
  unsigned long tempoAtual = millis();

  // Verifica se 1 segundo se passou
  if (tempoAtual - tempoAnterior >= intervalo) {
    // Atualiza o tempo anterior
    tempoAnterior = tempoAtual;

    // Exibe os dados no OLED
    display.clearDisplay();
    display.setCursor(0, 0);
    display.print("Altura da agua: ");
  }
}

```

```
display.print(ultimaAlturaAgua, 2); // 2 casas decimais
display.println(" cm");
display.print("Porcentagem: ");
display.print(ultimaPorcentagemAgua, 2); // 2 casas decimais
display.println(" %");
display.print("Fluxo: ");
display.print(ultimoFluxoAgua, 2); // 2 casas decimais
display.println(" L/min");
display.display();
}
}
```