

# Smart River - Monitorização e Análise da Qualidade do Rio e Consciencialização Ambiental

**Kenedy Portulez Cabanga - a36074**

Trabalho realizado sob a orientação de  
**José Luís Sousa de Magalhães Lima**  
**Ana Isabel Pinheiro Nunes Pereira**

Engenharia Industrial - Eletrotécnica

2021-2022



# Smart River - Monitorização e Análise da Qualidade do Rio e Consciencialização Ambiental

Dissertação de  
Mestrado em Engenharia Industrial - Eletrotécnica  
Escola Superior de Tecnologia e Gestão

Kenedy Portulez Cabanga - a36074

2021-2022

A Escola Superior de Tecnologia e de Gestão não se responsabiliza pelas opiniões expressas neste relatório.

Declaro que o trabalho descrito nesta dissertação é da minha autoria e é da minha vontade que o mesmo seja submetido a avaliação.

---

Kenedy Portulez Cabanga - a36074

---



# Dedicatória

*Dedico este trabalho a Centro de Ciência Viva de Bragança (CCVB), exemplos de vida, que em uma sociedade mergulhada em alegria, e entretenimento, com muita fé e dedicação ao trabalho, esforço, honestidade, simplicidade e respeito ao próximo venceram um conjunto de dificuldades, até chegarem a este auge que se encontram.*

# Agradecimentos

Agradeço primeiramente a Deus que me guiou e me abençoou até aqui, aos meus pais que me deram força e confiança, tendo em conta meus amigos que sempre estiveram ao meu lado e a minha namorada que me apoiou...

Agradeço aos meus orientadores, que sempre acreditaram na minha competência, que me deram força, por todos os conselhos, pela paciência e ajuda nesse período.

Aos professores que sempre estiveram ali para dar conselhos...

Ao CCVB, meus familiares e colegas, que deram apoio moral.

# Resumo

Na era da comunicação de Tecnologia da Informação (TI), o uso da Internet das Coisas (IoT) tornou-se popular entre outras soluções digitais, uma vez que oferece a integração de informações de várias áreas. Desta forma, pode-se aceder dados de locais distantes e a qualquer momento. No caso específico da monitorização de água, os métodos convencionais de medição são desatualizados podendo originar a problemas de baixa eficiência e complexidade. Assim, os sistemas inteligentes surgem como uma solução para uma ampla gama de casos. O Smart River é uma plataforma com sistema inteligente desenvolvida para otimizar os recursos e monitorizar os parâmetros de qualidade da água do rio Fervença. A solução central está localizada no Ciência Viva de Bragança (CCVB), um dos 21 centros de ciência em Portugal que visa promover a preservação e consciência ambiental da população. Ao utilizar as tecnologias IoT, o sistema permite a recolha de dados em tempo real com baixo custo e baixo consumo de energia, sendo um complemento dos projetos existentes que estão a ser desenvolvidos para promover os recursos naturais. Esta dissertação aborda a seleção do módulo sensor para a recolha de dados do rio e armazenamento de dados. Os parâmetros do rio são visualizados usando um programa desenvolvido na plataforma Unity3D para apresentar médias de dados e comparação entre semanas, meses e anos.

**Palavras-chave:** Sistemas inteligentes, Internet das Coisas (IoT), Parâmetros da água.

# Abstract

In the age of Information Technology (IT) communication, the use of communication and Internet of Things (IoT) has become popular among other digital solutions, since it offers the integration of information from several organisms and at several sources. By means of this, it can access data from distant locations and at any time. In the specific case of water monitoring, the conventional outdated measurement methods can lead to low efficiency and complexity issues. Hence, Smart systems rise as a solution for a broad of cases. Smart River is a platform developed to optimize the resources and monitoring the quality of water parameters of the Fervença river. The central solution is based at *Centro Ciência Viva de Bragança* (CCVB), one of the 21 science centers in Portugal that aims to promote the preservation and environmental awareness for the population. By using the IoT technologies, the system allows real-time data collection with low cost and low energy consumption, being a complement of existing projects that are being developed to promote natural resources. This paper covers sensor module selection for data collection inside the river and data storage. The parameters of the river are visualized using a program developed in Unity3D engine to present statistical data and comparison between weeks, months, and years.

**Keywords:** Smart systems, Internet of Things (IoT), Water parameters.

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Enquadramento . . . . .	2
1.2	Objetivos . . . . .	2
1.3	Estrutura do Documento . . . . .	3
<b>2</b>	<b>Estado da Arte</b>	<b>5</b>
2.1	Internet das Coisas . . . . .	5
2.1.1	Tecnologias que Tornaram Possível a IoT . . . . .	7
2.1.2	Onde Podem Ser Usadas a IoT . . . . .	8
2.2	Infraestrutura <i>The Things Network</i> . . . . .	9
2.3	Tecnologia LoRa . . . . .	12
2.3.1	LoRaWAN ou LPWAN . . . . .	12
2.3.2	Gateway LoRa . . . . .	13
2.3.3	Classes de Dispositivos para o uso da Tecnologia LoRa . . . . .	15
2.4	Microcontroladores . . . . .	15
2.5	Técnicas de Visualização . . . . .	18
2.6	Trabalhos Relacionados . . . . .	20
<b>3</b>	<b>Ferramentas e Tecnologias</b>	<b>23</b>
3.1	Hardware . . . . .	23
3.1.1	Microcontrolador ATMEGA328p-pu . . . . .	23
3.1.2	Transmissor LoRa RFM95 . . . . .	24

3.1.3	Gateway RAK7243C . . . . .	25
3.2	Sensores . . . . .	25
3.2.1	Sensor pH . . . . .	25
3.2.2	Sensor Condutividade Elétrica . . . . .	26
3.2.3	Sensor de Temperatura da Água . . . . .	27
3.2.4	Sensor DHT11 . . . . .	28
3.3	Caixa Protetora . . . . .	28
3.3.1	Caixa Protetora IP66 . . . . .	28
3.3.2	Caixa Protetora IP55 . . . . .	29
3.4	Software . . . . .	29
3.4.1	<i>BalenaEtcher</i> . . . . .	29
3.4.2	Plataforma WEB TTN . . . . .	30
3.4.3	Arduino IDE . . . . .	31
3.4.4	Unity 3D . . . . .	31
3.5	Tecnologia . . . . .	31
3.5.1	JavaScripts . . . . .	32
3.5.2	Node.js . . . . .	34
3.5.3	Prisma . . . . .	35
3.5.4	MQTT . . . . .	36
<b>4</b>	<b>Plataforma Smart River</b>	<b>37</b>
4.1	Localização . . . . .	37
4.2	Estrutura . . . . .	38
4.2.1	Dispositivo . . . . .	38
4.2.2	Exposição Interativa . . . . .	40
4.3	Comunicação . . . . .	43
4.3.1	Configuração da Plataforma . . . . .	43
4.3.2	Comunicação do Dispositivos com a Plataforma . . . . .	44
4.3.3	Comunicação da Plataforma com o Software de Interação . . . . .	45

4.3.4	Esquema de Comunicação . . . . .	46
4.4	Análise de Dados . . . . .	47
4.4.1	Cálculos em JavaScript . . . . .	48
4.4.2	Unity 3D Cálculo . . . . .	49
4.5	Node.js Supervisão . . . . .	50
<b>5</b>	<b>Teste, Avaliação e Discussão</b>	<b>53</b>
5.1	Dispositivo Implementado . . . . .	53
5.1.1	Códigos Relevantes . . . . .	53
5.1.2	PCI desenvolvida . . . . .	56
5.2	Dispositivo Virtual . . . . .	58
5.3	Dispositivo Final . . . . .	59
5.4	Dados Obtidos . . . . .	59
5.5	Monitorização . . . . .	60
5.5.1	Sistema de Supervisão Substituta . . . . .	60
5.5.2	Aplicação de Exposição Interativa . . . . .	61
5.6	Discussão . . . . .	64
5.6.1	Avaliação da Plataforma Smart River . . . . .	64
<b>6</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>67</b>
6.1	Conclusão . . . . .	67
6.2	Trabalhos Futuros . . . . .	68

# Lista de Tabelas

4.1	Tabela de Envio de Dados . . . . .	48
5.1	Tabela de Dados Obtidos . . . . .	60

# Lista de Figuras

2.1	IoT [11]. . . . .	6
2.2	Arquitetura MQTT [16]. . . . .	11
2.3	LoRaWAN arquitetura de rede [20]. . . . .	14
2.4	Arquitetura do LoRa [21]. . . . .	15
2.5	Arquitetura do microcontrolador [23]. . . . .	18
2.6	Imagem 3D [26]. . . . .	19
3.1	Estrutura ATmega328 [23]. . . . .	24
3.2	Estrutura LoRa RFM95 [32]. . . . .	24
3.3	Estrutura Gateway RAK7243C [33]. . . . .	25
3.4	Sensor de pH [35]. . . . .	26
3.5	Sensor de Condutividade Elétrica [37]. . . . .	27
3.6	Sensor de temperatura da água DS18B20 [39]. . . . .	27
3.7	Sensor DHT11 [40]. . . . .	28
3.8	Caixa Protetora IP66 [41]. . . . .	29
3.9	Caixa Protetora IP44, IP55, IP56 [42]. . . . .	30
3.10	<i>BalenaEtcher</i> Logótipo [43]. . . . .	30
3.11	<i>The Things Network</i> Logótipo [44]. . . . .	30
3.12	Arduino Logótipo [45]. . . . .	31
3.13	Unity Logótipo [46]. . . . .	31
3.14	<i>JavaScript</i> Logótipo [48]. . . . .	32
3.15	<i>Vue.js</i> Logótipo [50]. . . . .	33

3.16	EJS Logótipo [52]. . . . .	33
3.17	<i>AxiosJS</i> Logótipo [54]. . . . .	33
3.18	Socket.io Logótipo [56]. . . . .	34
3.19	<i>Node.js</i> Logótipo [58]. . . . .	35
3.20	Prisma Logótipo [59]. . . . .	35
3.21	MQTT Logótipo [61]. . . . .	36
4.1	Localização dos sensores [26]. . . . .	38
4.2	Circuito [26]. . . . .	39
4.3	Separadores Smart River[62]. . . . .	40
4.4	Fundo de apresentação gráfica[62]. . . . .	41
4.5	Botões Circulares[62]. . . . .	41
4.6	Botões Personalizados[62]. . . . .	42
4.7	Separadores De Informações[62]. . . . .	43
4.8	Comunicação do dispositivos com a plataforma [26]. . . . .	45
4.9	Comunicação da Plataforma com o Software de Interação [26]. . . . .	46
4.10	Esquema de Comunicação [26]. . . . .	47
4.11	Designer do Sistema de Supervisão [26]. . . . .	50
5.1	Código de envio [26]. . . . .	54
5.2	Função Dormir [26]. . . . .	55
5.3	Função Dormir [26]. . . . .	56
5.4	PCI [26]. . . . .	56
5.5	Monitor serial porta COM11 [26]. . . . .	57
5.6	Aplicação [26]. . . . .	58
5.7	Dispositivos Finais [26]. . . . .	58
5.8	Dispositivo Final [26]. . . . .	59
5.9	Sistema de Supervisão Substituta [26]. . . . .	60
5.10	Tela Principal da Aplicação [26]. . . . .	61
5.11	Tela Secundária de Dados-Casa da Seda [26]. . . . .	62

5.12	Tela Secundária de Dados-Castro de Avelãs [26]. . . . .	62
5.13	Tela Secundária de Dados-ETAR [26]. . . . .	63
5.14	Tela Secundária de Dados-IPB [26]. . . . .	63

# Siglas

**ABP** Activation By Personalization. 9, 44

**AES-CMAC** Advanced Encryption Standard-Cipher-based Message Authentication Code.  
44

**API** Interface de Programação de Aplicações. 33, 34

**APPSKEY** App Session Key (). 44, 45, 59

**CC** Corrente Contínua. 38

**CCVB** Centro de Ciência Viva de Bragança. vii, viii, 2, 59

**DEVADDR** Device Address. 44, 45, 59

**EJS** Embedded JavaScript Templating. 33

**ETAR** Estação de Tratamento de Águas Residuais. 40, 63

**IFTTT** If This Then That. 21

**IoT** Internet das Coisas. ix, 1, 5, 7, 8, 12, 24

**IPB** Instituto Politécnico de Bragança. 3, 40, 63

**LoRa** Long Range. 12, 13

**LPWA** low power wide area. 21

**MIT** Massachusetts Institute of Technology. 5

**NWKSKEY** Network Session Key (). 44, 45, 59

**ORM** Object-Relational Mapping. 35

**OTAA** Over-The-Air-Activation. 9, 10, 44

**PCB** Printed Circuit Board. 53

**PCI** Placas de Circuito Impresso. 53, 59

**RFID** Radio-Frequency IDentification. 5

**SO** Sistema Operacional. 25, 29

**SRMS** smart river monitoring system. 21

**TI** Tecnologia da Informação. ix

**TTN** The Things Network. 3, 5, 9, 10, 13, 25, 30, 36, 39, 43, 45, 46, 53, 55, 57

**VANT** veículos aéreos não tripulados. 21, 22

**WSNs** Wireless Sensor Networks. 21



# Capítulo 1

## Introdução

Atualmente, a quarta revolução industrial conhecida como Indústria 4.0 está-se a tornar cada vez mais popular entre a população por causa de suas amplas contribuições em diferentes setores. Entre as suas diferentes áreas (realidade virtual e aumentada, inteligência artificial, robôs avançados, veículos autônomos, computação em nuvem, *big data* e *blockchain* entre outras) [1], [2], a Internet das Coisas (IoT) tem vindo a destacar-se e crescendo rapidamente.

No geral, a principal vantagem da tecnologia IoT são as conexões, pois permite a comunicação entre vários dispositivos para compartilhar informações dentro de um sistema de dados baseado em nuvem. Muitos pontos podem ser conectados em qualquer lugar, a qualquer distância e a qualquer momento através da internet. Isso permite uma maior eficiência para entrega rápida de informações com baixo custo e esforço na construção de soluções inteligentes usando software, sensores, veículos e dispositivos eletrônicos interconectados [2]. Deste modo o Smart River tem sido utilizado para obter alguns dados dos rios para a monitorização da poluição ambiental.

O estudo realizado do rio Fervença, um afluente fronteiro do rio Sabor, tem a sua nascente localizada na aldeia de Fontes Barrosas e atravessa a cidade de Bragança, sendo um dos principais recursos hídricos para a atividade agropecuária nas pequenas comunidades da região. É também uma extensa fonte de biodiversidade nas suas águas e ribeirinhas, dentro e fora do território da cidade de Bragança.

Segundo a Rodrigues, o rio Fervença apresenta significativos níveis de perturbação que afetam decisivamente a qualidade e a integridade ecológica deste ecossistema aquático [3]. Pois a poluição doméstica e industrial, alarmam para a necessidade de assegurar ações que visem a reparação de habitats e melhoria da qualidade da água. [3].

## 1.1 Enquadramento

O rio Fervença é um recurso inestimável, mas a poluição das áreas urbanas e agrícolas representam uma ameaça à qualidade da água. Por outro lado, devem ser implementadas ações de sensibilização e educação ambiental que modifiquem comportamentos e contribuam, no futuro, para a melhoria da integridade ecológica deste rio e sua conservação [3]. Para entender o valor da qualidade da água, gerir e proteger efetivamente os recursos hídricos, é fundamental conhecer o status atual das condições de qualidade da água e como e porquê essas condições estão mudando ao longo do tempo. É necessário saber como está a qualidade do rio para tentar diminuir o crescente número de rios poluídos com dispositivos eletrónicos capazes de medir alguns parâmetros no rio fazendo monitorização dos mesmos. Já que os métodos convencionais para monitorizar a qualidade da água precisam deslocar-se ao local para efetuar uma amostragem da água diretamente ou analisando-a em laboratório [4].

## 1.2 Objetivos

O sistema proposto aproveita as tecnologias de IoT para construir uma solução inteligente. A plataforma Smart River é um conjunto de sistemas inteligentes e interligados através da nuvem para medir os parâmetros e a qualidade da água do rio Fervença em Bragança, Portugal. Deste modo tem-se como objetivo medir parâmetros físicos e químicos do rio Fervença, armazenar todos os dados num servidor baseado em nuvem e exibi-los num monitor touchscreen [5] em uma das exposições do CCVB. A exposição tem como objetivo oferecer uma experiência interativa, científica e educativa sobre a importância de realizar

a monitorização ambiental aliado às mais recentes tecnologias de comunicação. Todo o sistema deverá ser desenvolvido para atender as seguintes especificações:

- Dispositivos eletrónicos capazes de obter o pH da água, a temperatura da água e a condutividade da água, a temperatura ambiente, e a humidade relativa do ar;
- Ter dispositivos virtuais na plataforma The Things Network (TTN);
- Conexão dos dispositivos virtuais com o eletrónico para a recolha de dados;
- Os dados armazenados no computador em um arquivo TXT;
- Apresentação dos dados de forma interativa.

### 1.3 Estrutura do Documento

Este documento foi desenvolvido em 6 Capítulos. A integração destes relatam todo o trabalho desenvolvido no Instituto Politécnico de Bragança (IPB), a seguir as sinopses de cada um destes elementos.

O Capítulo 1, salienta a caracterização do problema abordado, a motivação ao qual se pretendeu realizar este trabalho, e por sua vez são esclarecidos os objetivos proposto para solução.

No Capítulo 2, aborda-se a revisão literária necessária para o desenvolvimento do trabalho, onde são demonstrados os estudos realizados para o desenvolvimento dos dispositivos, e a conexão com a nuvem.

No Capítulo 3, focou-se nas ferramentas disponíveis para o desenvolvimento deste trabalho, tanto teóricas quanto as práticas, como o *hardware*, *software* e materiais, não esquecendo de abordar sobre a suas aplicações.

No Capítulo 4, observa-se nesta seção o desenvolvimento do dispositivo proposto, e também os processos realizados para alcançar os objetivos, desde a criação, como sua elaboração desde o *hardware* até a comunicação com TTN

No Capítulo 5, expõem-se os testes e avaliações realizada do protótipo desenvolvido, e também a discussão dos valores e resultados obtidos no decorrer do projeto até a sua finalidade.

O Capítulo 6, é dedicado à conclusão do projeto, e também dos trabalhos futuros.

# Capítulo 2

## Estado da Arte

Neste capítulo serão mencionados conceitos relevantes e trabalhos acadêmicos para o contexto da dissertação, tecnologias e serviços de IoT, e desenvolvimento do dispositivo eletrônico. Onde estes serão divididos em seções, a primeira seção descreveu-se o IoT, na segunda seção foi observada a infraestrutura da TTN, a terceira seção foi abordada a tecnologia LoRa, a quarta seção foi dedicada aos micro-controladores, a quinta seção abordou-se as técnicas de visualização e a última seção foi realizado as pesquisas coerentes aos trabalhos relacionados mostrando o essência deste projeto.

### 2.1 Internet das Coisas

O termo *Internet of Things* (IoT) ou Internet das Coisas, foi apresentado primeiro por Kevin Asthon do *Massachusetts Institute of Technology (MIT)* em uma apresentação sobre *Radio-Frequency IDentification (RFID)* e a cadeia de suprimentos da Procter e Gamble (PeG) em 1999 (ASHTON, 2009) [6].

Segundo Madakam et al. a *Internet of Things* em inglês, e em português Internet das coisas é uma revolução tecnológica que representa o futuro da computação e das comunicações, e seu desenvolvimento depende de inovação técnica dinâmica em vários campos importantes, desde sensores sem fio até nanotecnologia. A estratégia passa por etiquetar cada objeto para identificar, automatizar, monitorar e controlar [7], [8].

A frase “Internet das Coisas”, que também é brevemente conhecida como IoT, é uma rede de redes que consiste em milhões de redes privadas, públicas, acadêmicas, empresariais e governamentais, de âmbito local a global, que estão ligadas por uma ampla gama de tecnologias de redes eletrônicas, sem fio e ópticas [7].

A melhor noção a Internet das Coisas seria: “Uma rede aberta e abrangente de objetos inteligentes que têm a capacidade de se auto-organizar, compartilhar informações, dados e recursos, reagindo e agindo diante de circunstâncias e alternativas no ambiente”[7].

De acordo com a oracle a Internet das Coisas descreve a rede de objetos físicos – “coisas” – que são incorporados a sensores, software e outras tecnologias com o objetivo de conectar e trocar dados com outros dispositivos e sistemas pela Internet [9].

Com a observação de Colombo e de Filho, a Internet das Coisas, é uma infraestrutura de comunicação que permite conectar o mundo real e o virtual, criando um “novo mundo” mais inteligente nos diversos segmentos da sociedade moderna [10].

A Figura 2.1 observa-se a ampla comunicações entre vários dispositivos, e objetos, em que é conhecida como IoT.



Figura 2.1: IoT [11].

### **2.1.1 Tecnologias que Tornaram Possível a IoT**

Para a IoT estar em um grande avanço foi necessário um avanço nas tecnologias que tornaram a IoT possíveis. Nesta subseção abordou-se sobre essas tecnologias.

#### **Acesso à Tecnologia de Sensores de Baixo Custo e de Baixa Potência.**

Sensores acessíveis e leais estão tornando a tecnologia IoT possível para mais fabricantes.

#### **Conectividade.**

Uma série de protocolos de rede para a internet tornou fácil associar sensores à nuvem e a distintas "coisas" para uma transição apropriado de dados.

#### **Plataformas de Computação em Nuvem.**

O crescimento da disponibilidade de plataformas em nuvem aprova que empresas e consumidores acedam a infraestrutura de que necessitam para escalar sem deveras ter que gerenciar tudo.

#### **Aprendizagem de Máquina e Análise.**

Com as evoluções na aprendizagem de máquina e análise, simultaneamente com o disponibilidade a importância variadas e vastas de dados acomodados na nuvem, as empresas podem recolher percepções mais rapidamente e com mais facilidade. O nascimento dessas tecnologias aliadas segue a ultrapassar os limites da IoT e os dados alcançados pela IoT também consomem essas tecnologias.

#### **Inteligência Artificial Conversacional (IA).**

Os progressos nas redes neuronais nutriram o processamento de linguagem natural para dispositivos IoT (como assistentes pessoais digitais Alexa, Cortana e Siri) e os tornaram sedutores, coerentes e capazes para uso doméstico.

### 2.1.2 Onde Podem Ser Usadas a IoT

É provável que, pelo menos hoje em dia, alguns não manifestem muito entusiasmo em ter uma casa abertamente conectada. Em conformidade com esse ponto de perspectiva, a Internet das Coisas pode não parecer lá muito pertinente. Mas é uma discrepância cogitar que o conceito serve meramente para o habitação: há muitas serventias não ligadas ao ambiente doméstico em que o conceito é capaz de trazer ganho de produtividade ou amenizar custos de produção, segundo as pesquisas feitas, a IoT pode ser usadas em:

- **Hospitais e clínicas:** pacientes podem usufruir de equipamentos conectados que medem batimentos cardíacos ou pressão sanguínea, por exemplo, e os dados recebidos serem enviados em tempo real para o sistema que controla os exames [12], [13];
- **Agropecuária:** sensores espalhados em plantações podem dar informações deveras precisas acerca de temperatura, humidade do solo, probabilidade de chuvas, velocidade do vento e outras informações essenciais para o bom rendimento do plantio. De igual forma, sensores conectados aos animais conseguem colaborar no controle do gado: um chip distribuído na orelha do boi pode fazer o rastreamento do animal, instruir seu histórico de vacinas e logo por diante [12], [13];
- **Fábricas:** a Internet das Coisas pode ajudar a medir, em tempo real, a rendimento de equipamentos ou esclarecer quais setores da planta carecem de mais equipamentos ou suprimentos [12], [13];
- **Lojas:** armários inteligentes podem instruir, em tempo real, quando algum item está provindo a faltar, que produto está tendo menos saída (exigindo medidas como reposicionamento físico ou criação de promoções) ou em quais turnos determinados produtos vendem mais (ajudando na produção de estratégias de vendas) [12], [13];
- **Transporte público:** clientes podem saber, pelo smartphone isto é em telas instaladas nos sítios de embarque, qual a localização de certo ônibus. Os sensores

inclusive podem ajudar a empresa a constatar que um veículo apresenta problemas mecânicos, assim como saber como está o desempenho de horários;

- **Logística:** dados de sensores instalados em caminhões, contêineres e até caixas específicos estabelecidos com informações do trânsito, por exemplo, podem amparar uma empresa de logística a descrever as excelentes caminhos, seletar os veículos mais propícios para determinada área, definir quais encomendas distribuir entre a frota ativa e assim por diante [12], [13];
- **Serviços públicos:** sensores em lixeiras podem colaborar a prefeitura a otimizar a recolha de lixo; carros podem se conectar a uma central de acompanhamento de trânsito para atingir a excelente rota para aquele momento, oxalá como para ajudar o setor de controle de tráfego a saber quais vias da cidade estão mais movimentadas naquele momento. Este fator nos leva ao temática das "cidades inteligentes"[12], [13].

## 2.2 Infraestrutura *The Things Network*

A TTN visa permitir que os dispositivos de baixo consumo de energia usem gateways de longo alcance para se conectarem a uma rede descentralizada de código aberto para trocar dados com aplicações. A TTN é a primeira infraestrutura descentralizada de código aberto para a Internet das coisas. Essa infraestrutura pode comunicar com o dispositivos que deve ser registado em uma rede antes de enviar e receber mensagens. Este procedimento é conhecido como ativação. Existem dois métodos de ativação disponíveis:

1. *Over-The-Air-Activation (OTAA)* - o método de ativação mais seguro e recomendado para dispositivos finais. Os dispositivos executam um procedimento de junção com a rede, durante o qual um endereço de dispositivo dinâmico é atribuído e as chaves de segurança são negociadas com o dispositivo [14].
2. *Activation By Personalization (ABP)* - requer a codificação do endereço do dispositivo, bem como as chaves de segurança no dispositivo. O ABP é menos seguro que

o OTAA e também tem a desvantagem de que os dispositivos não podem trocar de provedor de rede sem alterar manualmente as chaves no dispositivo [14].

## Protocolo MQTT

Segundo a plataforma TTN o protocolo MQTT é um dos protocolos utilizado pela plataforma para publicar ativações de dispositivos e mensagens, mas também permite que você publique uma mensagem para um dispositivo específico em resposta [14].

O MQTT é um protocolo de conectividade máquina a máquina. Ele foi ilustrado como um transporte de mensagens de publicação/assinatura muito leve. Basicamente, o protocolo permite o envio de mensagens e comandos entre os dispositivos que comunicam através de MQTT por meio de TCP/IP [14].

Os clientes MQTT necessitam de recursos mínimos e podem ser consumidos em microcontroladores pequenos. Os cabeçalhos das mensagens MQTT são pequenos para aprimorar a largura de banda da rede [15].

O protocolo MQTT foi projetado pelos engenheiros Andy Stanford-Clark e Arlen Nipper em 1999. A ideia era conectar sistemas de telemetria de oleodutos por satélite e, hoje, possui incontáveis possibilidades. No começo, este era um protocolo particular, porém, em 2010, foi lançado em royalties e se tornou um padrão OASIS em 2014 [15].

## Subscribe e Publish

O essencial conceito a saber é o de *Subscribe* e *Publish*. Um dispositivo é capaz de ser “inscrito= *subscribed*” ou uma “publicação=*publish*”. Enquanto tal dispositivo é um *Publish*, ele publica informações pré-estabelecidas. Logo o dispositivo *Subscribe* recebe todas essas informações que estão sendo publicadas [16].

Se o dispositivo produz alguma informação que não está cadastrada, esta jamais será publicada. Por isso que a maior vantagem deste protocolo é a economia de banda de internet, já que não há tráfego de informações sem interesse [16].

## Broker

Este é um dos elementos essenciais para que o protocolo MQTT suceda. O Broker é um servidor que gere as informações aos registrados e reciprocamente, trabalhando, pois, como um mediador entre Subscribers e Publishers [16].

Além de armazenar, o Broker também filtra as informações. Os dados dentro deste servidor estão armazenados em tópicos e, desta forma, os Subscribers indicam quais os tópicos querem se inscrever e receber unicamente os que os convém. Em distintas palavras, o Broker recebe todas as mensagens, filtra e define quem está interessado e inscrito nela ou não [16].

O Broker é capaz ser tanto um servidor local, como uma estrutura em nuvem.

## Arquitetura MQTT

A Figura 2.2 apresenta a arquitetura MQTT, onde as informações de temperatura que são recolhidas e armazenadas no Broker no tópico “temperatura”, por exemplo. A partir do momento que você escolher os Subscriber, eles vão receber apenas as informações relativas a este tópico nos dispositivos cadastrados, facilitando o acesso aos dados de temperatura para as análises necessárias [16].

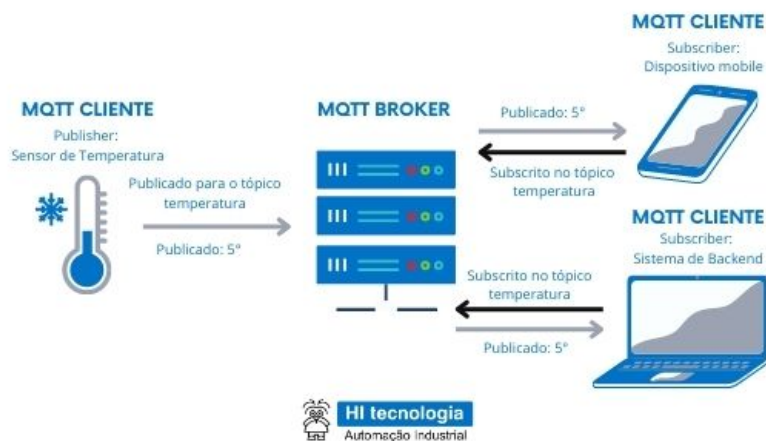


Figura 2.2: Arquitetura MQTT [16].

## 2.3 Tecnologia LoRa

Surgiram novas tecnologias de transceptores que permitem uma comunicação eficiente em termos de energia a longas distâncias. Exemplos dessas tecnologias de rede de área ampla de baixa potência são Long Range (LoRa), Sigfox e Weightless. Um cenário típico de aplicação para essas tecnologias é a recolha de leitura de medidores em toda a cidade, na qual os dispositivos enviam leituras em frequência muito baixa por uma longa distância para um concentrador de dados (redes de um salto). Argumentamos que esses transceptores são potencialmente muito úteis para construir redes mais genéricas da IoT, incorporando comunicação bidireccional multip hop, permitindo detecção e atuação. Além disso, esses transceptores têm recursos interessantes indisponíveis nos transceptores mais tradicionais usados para redes IoT, que permitem a construção de novos elementos de protocolo [17].

### 2.3.1 LoRaWAN ou LPWAN

Com o rápido crescimento do paradigma da Internet das Coisas (IoT), outra importante tecnologia surge no campo de pesquisa que auxilia na obtenção de uma comunicação sem fio de alto desempenho entre dispositivos, é chamada de rede de longa distância de baixa potência (LPWAN). LoRa não é apenas uma rede de longo alcance, também permite a transferência de informações com baixa taxa de transferência e baixo consumo de energia. LoRaWAN é uma tecnologia promissora desenvolvida para superar os desafios de aplicativos de IoT [18], [19].

LoRaWAN é o nome dado ao protocolo que define a arquitetura do sistema bem como os parâmetros de comunicação usando a tecnologia LoRa [18], [19], ele é muito usado na plataforma descrita na seção 2.2.

O protocolo LoRaWAN implementa os detalhes de funcionamento, segurança, qualidade do serviço, ajustes de potência visando maximizar a duração da bateria dos módulos, e os tipos de aplicações tanto do lado do módulo quanto do servidor [18], [19].

Uma das tecnologias que seria necessária para a integração da Smart River é o Lora para a recolher as leituras de medidores em todo o rio de uma cidade, podendo fazer

comunicações com a rede enviando fazendo trocas de dados.

## Arquitetura LoRaWAN

De acordo com a TTN uma rede LoRaWAN típica consiste nos seguintes elementos que são:

- *End Devices* - sensores ou atuadores enviam mensagens sem fio moduladas LoRa para os gateways ou recebem mensagens sem fio de volta dos gateways.
- Gateways - recebe mensagens de dispositivos finais e as encaminhe para o Network Server.
- *Network Server* - um software executado em um servidor que gerencia toda a rede.
- *Application servers* - um software executado em um servidor responsável pelo processamento seguro de dados de aplicativos .
- *Join Server* - um software executado em um servidor que processa mensagens de solicitação de adesão enviadas por dispositivos finais (o Join Server não é mostrado na figura em baixo) [20].

A Figura 2.3 demonstra uma típica arquitetura de rede loRaWAN com os seus elementos [20].

### 2.3.2 Gateway LoRa

Os gateways formam a ponte entre os dispositivos e a TTN. Os dispositivos usam redes de baixa energia, como o LoRa, para conectar-se ao Gateway, enquanto o Gateway usa redes de alta largura de banda, como WiFi, Ethernet ou Celular, para conectar-se à TTN [14]. Todos os gateways ao alcance de um dispositivo receberão as mensagens do dispositivo e as encaminharão para a TTN. A rede eliminará mensagens duplicadas e selecionará o melhor gateway para encaminhar as mensagens na fila para o *downlink*. Um único gateway pode servir milhares de dispositivos [14]. A Figura 2.4 apresenta a arquitetura

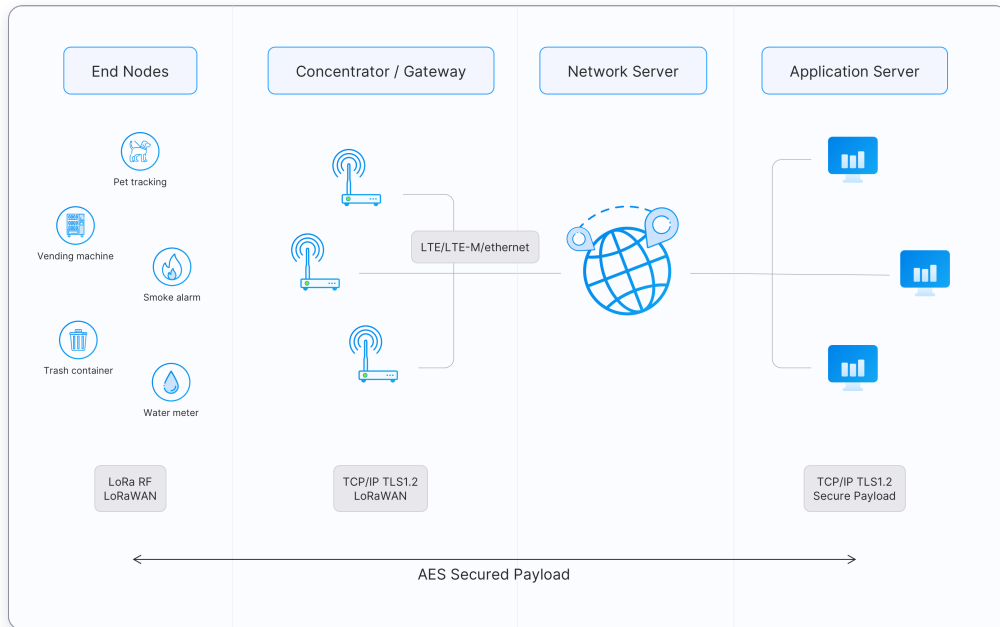


Figura 2.3: LoRaWAN arquitetura de rede [20].

do LoRa onde Servidor de Aplicação (*Application server*) são programas específicos que recebem os pacotes dos servidores de rede e de acordo com a informação executam uma ou mais ações específicas, Servidor de Rede (*Network server*) que é responsável pela gestão das informações enviadas pelos gateways, Gateway é responsável por receber os dados de milhares de dispositivos e encaminhá-los para o servidor de rede, e por fim *End-Devices* que são elementos básicos de rede como Sensores, e etc[21].

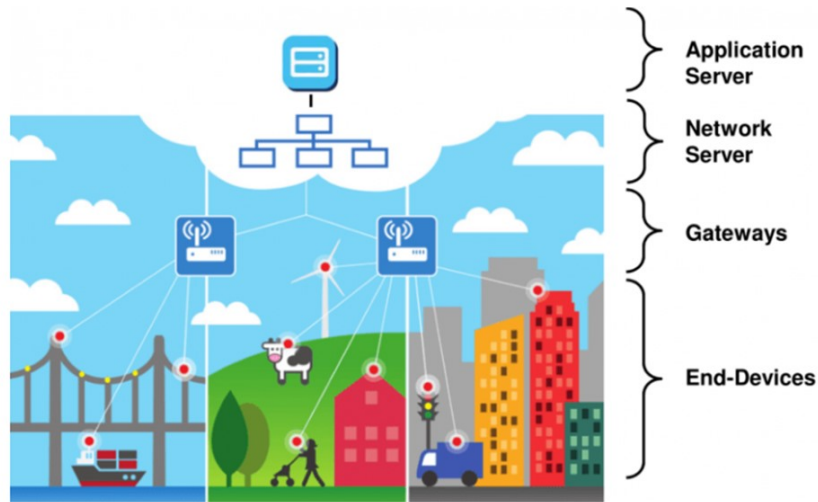


Figura 2.4: Arquitetura do LoRa [21].

### 2.3.3 Classes de Dispositivos para o uso da Tecnologia LoRa

Para atender as mais diversas necessidades no protocolo LoRaWAN foram definidas três classes de dispositivos (módulos):

Classe A - Sensores Comunicação bi-direcional, recepção após transmissão. Os módulos podem receber dados em janelas de tempo pré determinadas e imediatamente após realizarem uma transmissão [21];

Classe B - Atuadores Comunicação bi-direcional, com janelas de recepção agendadas [21];

Classe C - Bidirecional, recepção de dados praticamente sem atraso. Nesta classe o módulo sempre esta apto a receber dados do gateway [21].

## 2.4 Microcontroladores

Para entender o microcontrolador foi observado um sistema embarcado onde trata se de um sistema computacional completamente independente, geralmente fabricado para

um propósito único, e com o objetivo de executar tarefas específicas. Esses sistemas possuem seus próprios recursos computacionais como processamento, memória e portas de comunicação de entrada e saída. Este também pode ser programado e reprogramado varias vezes, e executar tarefas predefinidas e geralmente de uso limitadas [22].

A programação das instruções de um sistema embarcado é feita através de uma encriptação geralmente chamada de *firmware*, é por meio dela que seus desenvolvedores determinam as serventias dos dispositivos. O *firmware* regularmente é armazenado em uma memória ROM ou memória FLASH diferentemente de um computador pessoal no qual o armazenamento é completo por um disco rígido [22].

Os **microcontroladores** eram produzidos a partir de componentes individuais como resistências, condensadores, díodos e transístores entre outros. Porém, o uso desses componentes tornavam circuitos mais complexos muito grandes, o que tornavam os projetos eletrônicos caros e até mesmo poderiam inviabilizar alguns projetos segundo o Araujo, Cavalcante, e o Silva [22].

Com o surgimento dos circuitos integrados isso mudou. Circuito integrado é um circuito eletrônico minúsculo inventado com semicondutores que substitui os elementos outrora apontados, encapsulado sobre o substrato de um material muito fino, regularmente o silício. Com isso foi plausível abreviar significativamente o tamanho dos equipamentos eletrônico [22].

O microcontrolador é considerado um computador completo em miniatura acrescentado em um único circuito integrado. No interior de um microcontrolador uma unidade de processamento e os elementos importantes para que ele possa cumprir funções de forma autónoma, conforme memórias de leitura, escrita de dados e armazenamento da firmware, conversores de sinais analógico/digital e digital/analógico, e portas programáveis de entrada e saída para diversas aplicabilidade como dirigir outros dispositivos, fornecer uma interação com o usuário entre outras [22].

De acordo com a Figura 2.5 está descrita a função de cada um dos blocos que formam o microcontrolador:

\* **CPU**– É a Unidade Central de Processamento, assim como nos computadores,

pode-se dizer que é o cérebro da máquina. É neste local onde são tomadas as decisões e o processamento dos dados [23].

- \* **ULA** – Chamada de Unidade Lógica e Aritmética é responsável por realizar os cálculos matemáticos e de lógica. Geralmente faz parte interna da CPU [23].

- \* **RAM** – É a Memória de Acesso aleatório, assim como nos computadores, serve para realizar o armazenamento de informações momentâneas durante um cálculo [23].

- \* **PORT A, B e C** – São os terminais chamados de portas do microcontrolador, também conhecidas por portas digitais e são por elas que as informações digitais entram ou saem [23].

- \* **Entradas Analógicas** – São portas unicamente de entrada que servem para registrar sinais analógicos do meio externo. Esses sinais podem ser os mais diversos como som, temperatura, luminosidade, humidade entre outros [23].

- \* **EPROM** – É a memória usada capaz de armazenar *firmware*, que é o programa criado pelo desenvolvedor, e pode ser escrita e apagada cerca de 10.000 vezes a 100.000 vezes. Esse programa é que vai estipular as funcionalidades e forma de trabalho do microcontrolador [23].

- \* **EEPROM** – Memória interna que consegue ser usada para guardar informação. Ela é apto de guardar os dados e mantê-los embora se o microcontrolador for desligado. Dependendo do microcontrolador os dados poderão ficar retidos em torno de 40 anos, depois de esse tempo não se garante que as informações continuaram intactas [23].

- \* **Program Counter** – Serve para descrever ao processador interno do microcontrolador a ordem de execução do código e no caso de uma rotina de interrupção ele armazena a linha seguinte para quando o processador retornar da interrupção e logo poder permanecer com a execução sequencial do código [23].

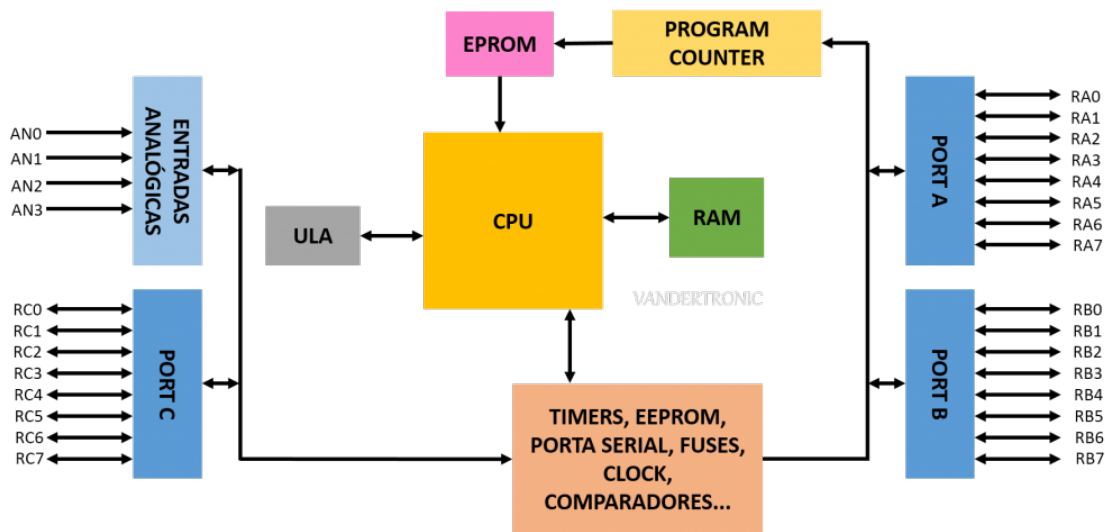


Figura 2.5: Arquitetura do microcontrolador [23].

## 2.5 Técnicas de Visualização

De acordo com o Pillat técnicas de visualização de informações constituem graficamente dados de um certo domínio de aplicação e disponibilizam mecanismos para a interação com a representação exercida, a fim de que o usuário consiga analisar e compreender as informações apresentadas [24].

Segundo Valiati e Almeida as técnicas de visualização de informações podem ser classificadas de diversos feitios, baseadas: nas funções a serem suportadas, na estrutura do conjunto de dados, na natureza dos dados a serem visualizados, na dimensionalidade dos dados ou do espaço em que serão representados, na interpelação de mapeamento assumido pelas técnicas quer nos métodos de manipulação e interação [25].

As táticas de visualização multidimensionais, em original, referem-se à visualização de informações onde cada elemento no conjunto de dados é referido, ou descrito, por múltiplas variáveis (atributos), as quais carecem ser codificadas em uma única estrutura visual [24].

As peças visuais do conhecimento são adicionalmente categorizadas de cautela em relação a o tipo de estrutura visual acolhida, a que se menciona à dimensionalidade das representações de dados aproveitada. Logo, entre as estruturas visuais básicas estão as:

- físicas (ou naturais): referentes à representação de dados típicos da Visualização Científica, que possuem uma correspondência direta com objetos, fenômenos ou posições do mundo real [25]
- 1D, 2D, 3D: relacionadas a representações visuais, que criptografam informações durante do posicionamento de símbolos ou marcas em superfícies uni, bi ou tridimensionais. Estruturas 1D podem ser observadas na representação de dados temporais. Estruturas visuais 2D e 3D são, regularmente, vistas em gráficos usuais (de barras, pontos, linhas e círculos) e utilizadas para evidenciar dados físicos, porquanto também, informações abstratas [25].
- técnicas de visualização multidimensionais: em original, referem-se à visualização de informações onde cada elemento no conjunto de dados é referido, ou descrito, por múltiplas variáveis (atributos), as quais carecem ser codificadas em uma única estrutura visual [24].
- árvores e redes: correspondem a quaisquer as técnicas que constituem uma hierarquia para a exibição dos dados, independente destes revelarem ou não uma estrutura hierárquica pertinente. Equivalem, regularmente, de nodos representando dados e ligações ostentando o relacionamento entre nodos. Tais estruturas são colocadas para evidenciar, por amostra, taxonomias, organizações, gestão de tráfego, espaço em disco ou estruturas de informação na web [25].

A Figura 2.6 demonstra a técnica de visualização 3D que é umas das mais utilizadas neste tempo de tecnologia.

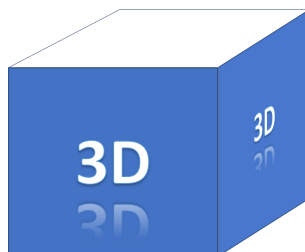


Figura 2.6: Imagem 3D [26].

## 2.6 Trabalhos Relacionados

O trabalho realizado por Adu-Manu, Katsriku, Abdulai e Engmann, cujo o tema é "***Smart River Monitoring Using Wireless Sensor Networks***", eles utilizaram abordagem de um cronograma de transmissão de dados com eficiência energética e energia colhida usando painéis solares para prolongar a vida útil do nó, este estudo segundo o artigo deles foi realizado na entrada de Weija, na região da Grande Accra, em Gana em uma Barragem que serve como uma fonte de água significativa para a estação de tratamento de Weija, que fornece água tratada para o povo da Grande Accra e partes das regiões centrais de Gana [27].

Foram usados sensores de água inteligentes e dispositivos de sensores de íons de água inteligentes da Libelium que foram implantados na entrada para medir parâmetros físicos e químicos, e com isso obtiveram uns resultados significativos, cujo dados obtidos no repositório central revelaram um valor de pH de 7, os níveis de condutividade aumentaram de  $196.S/cm$  para  $225.S/cm$ , os níveis de cálcio aumentaram para cerca de  $3,5.mg/L$  e caíram para cerca de  $0,16.mg/L$ , a temperatura do rio era principalmente em torno de  $35^{\circ}C$  a  $36^{\circ}C$ . Eles observaram também níveis de flúor entre  $1,24.mg/L$  e  $1,9.mg/L$ , e o conteúdo de oxigênio subiu do OD negativo para atingir  $8.mg/L$ . Estes dados foram guardados na nuvem em um portal da web em um centro de gestão por meio de uma infraestrutura de comunicação 4G [27].

Segundo o trabalho realizado por Ahmad e Kumar, "***IOT Based Smart River Monitoring System***", foi envolvido a abordagem da Internet das Coisas (IoT) que será feita por meio de um microcontrolador, desse modo é utilizado o sensor para monitorar os parâmetros da água como pH, turbidez, temperatura, potencial de oxidação-redução, Oxigênio dissolvido, resíduos industriais, resíduos hospitalares, estes dados serão armazenado em um servidor na internet, o microcontrolador enviara as informações no banco de dados usando o protocolo TCP para *Thingspeak* e do *Thingspeak* usando i-frame colocam dados no site deles de gestão inteligente (smart city), e salvam informações sobre a hora em que os dados são registrados criando um arquivo de log na nuvem do *Thingspeak* [28].

Foi criado um gráfico que é um gráfico de dados do sensor em relação ao tempo e usando o protocolo *If This Then That (IFTTT)* e, em seguida, um SMS é enviado para o usuário final. Este banco de dados é acessível ao usuário autorizado sempre que ele precisar de dados para análise [28].

De acordo com o trabalho de Kadir, Siswanto, Rosa, Syukur, Irie e Othman, cujo o tema é "***Smart Sensor Node of WSNs for River Water Pollution Monitoring System***", foi utilizado Wireless Sensor Networks (WSNs) em português (Redes de Sensores Sem Fio), devido às vantagens do sistema WSNs, um nó sensor acoplado a diversos sensores, como temperatura da água, pH, oxigênio dissolvido (OD) e condutividade elétrica. O sistema de monitorização de água do rio projetado pode monitorar o nível de água e a vazão para alerta ambiental e de inundação, um nó de sensor personalizado e projetado para cumprir o padrão de qualidade da água, além de adicionar outros sensores para alerta de inundação, dados recolhidos pelo nó sensor são encaminhados para o nó coletor da WSNs que é incorporado com uma unidade de microcontrolador e memória como banco de dados local antes de enviar para o sistema back-end [29].

A monitorização no sistema de back-end é exibido por alguns monitores para facilitar a visualização pela instituição representativa ou autoridade local, bem como ações serão tomadas se ocorrer algum caso ou anormalidade relatada pelo sistema de gestão. Protótipo de nó de WSNs foi projetado e testado, os resultados mostram que o nó sensor é capaz de detetar parâmetros de água poluída e nível de água, bem como taxa de vazão [29].

Conforme o trabalho do Elijah, Abdul, Leow, Yeen, Sarijari, Aris, Salleh e Chua, com o tema "***A Concept Paper on Smart River Monitoring System for Sustainability in River***", foi elaborado o smart river monitoring system (SRMS) ou em português (sistema inteligente de gestão de rios), que utiliza veículos aéreos não tripulados (VANT) ou drones e low power wide area (LPWA) em português (tecnologia de comunicação de área ampla de baixa potência), devido à extensão do rio que precisa ser monitorada, as tecnologias de comunicação convencionais como Wi-Fi, Zigbee, Bluetooth não são adequadas, segundo o que eles relataram no artigo. Desse modo existe a necessidade de tecnologia de comunicação LPWA [30].

De acordo com resultados preliminares que eles adquiriram mostram que o uso de VANT aumentará a eficiência de medição dos parâmetros de qualidade da água em comparação com o método de gestão manual, o gestão em tempo real lhes permitiu estudar as mudanças na qualidade da água [30].

# Capítulo 3

## Ferramentas e Tecnologias

Neste capítulo apresenta-se as ferramentas e tecnologias usadas para montagem do dispositivo, onde abordou-se o hardware que compõe a parte física, os sensores usados para a recolha de dados, e o software que seria a parte de configuração, e programação para o controle do sistema, na última seção as tecnologias necessária para a comunicação.

### 3.1 Hardware

Nesta seção foi observado, o microcontrolador usado, as ferramentas de comunicação, e a alimentação do dispositivo.

#### 3.1.1 Microcontrolador ATMEGA328p-pu

Este microcontrolador foi usado por ser de fácil acesso, de baixo preço e de fácil programação, onde este é uma série ATmega AVR conhecida como microcontrolador embalado em 28-DIP. Possui o processador de 8 bits com a frequência de 20 MHz. Tem conectividade por I2C, SPI, UART/USART. Memória flash de 32 Kb, 8 Kb EEPROM e 16 Kb RAM [31]. A Figura 3.1 demonstra a sua estrutura física com várias entradas e saídas, onde estes já dizem a forma que estão de acordo com os pinos de ligação de uma placa Arduino.

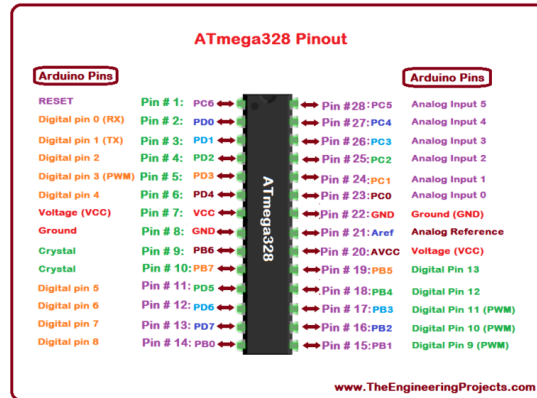


Figura 3.1: Estrutura ATmega328 [23].

### 3.1.2 Transmissor LoRa RFM95

O transmissor LoRa RFM95 foi utilizado porque é muito usado no mercado, fazendo parte de uma tecnologia muito avançada na área do IoT conhecida como tecnologia LoRa, este também é de fácil acesso, de baixo custo, e pode ser usado por ATmega328 por pinos de Arduino, deste modo, é uma das ferramentas que foi possível estabelecer a comunicação com a plataforma. Na Figura 3.2 pode ser observado o transceptor RFM95 868MHz que possui o modem LoRa de longo alcance que oferece comunicação de espectro disperso de alcance ultra-longo e alta imunidade a interferências, minimizando o consumo de corrente [32].



Figura 3.2: Estrutura LoRa RFM95 [32].

### 3.1.3 Gateway RAK7243C

O gateway RAK7243C é a ferramenta mais essencial para a comunicação entre o dispositivo e a plataforma web TTN, onde este é de configuração fácil, e de baixo preço no mercado, este também vem acompanhado de um cartão de memória, uma LTE antena, uma LoRa antena, um GPS antena, e também uma documentação apropriada para fazer a instalação de um Sistema Operacional (SO), para conseguir fazer a configuração [33]. Na Figura 3.3 observa-se o Gateway utilizado, que faz a ponte entre o dispositivo e a plataforma.



Figura 3.3: Estrutura Gateway RAK7243C [33].

## 3.2 Sensores

Nesta seção foram observados os sensores para análise de dados do rio, em que estes são o sensor de pH, o sensor de condutividade elétrica, o sensor de temperatura da água, e o sensor DHT11.

### 3.2.1 Sensor pH

O pH é a sigla usada para potencial (ou potência) hidrogeniônico, porque se refere à concentração de  $[H^+]$  (ou de  $H_3O^+$ ) em uma solução. Este sensor mede o nível de pH

na água e para indicar se uma solução é ácida, neutra ou básica, a escala de pH varia entre 0 e 14 na temperatura de 25°C. Se o valor do pH for igual a 7 (pH da água), o meio da solução (ou do líquido) será neutro. Mas se o pH for menor que 7, será ácido, e se for maior que 7, básico, deste modo pode-se observar a qualidade da água, este sensor representado na Figura 3.4, foi escolhido por ser de baixo preço, e pode ser encontrado no mercado facilmente [34].



Figura 3.4: Sensor de pH [35].

### 3.2.2 Sensor Condutividade Elétrica

O sensor de condutividade elétrica (sensor EC) mede a condutividade elétrica em uma solução que normalmente é usada para testes de agricultura e qualidade da água. O Sensor de Condutividade Elétrica foi especialmente projetado para um sistema de baixo custo com uma precisão relativamente alta que pode cobrir a maioria das aplicações, por este motivo foi escolhido este sensor, que esta representado na Figura 3.5. O conector sonda BNC facilitam o uso e são muito adequados para o projeto Arduino e Raspberry Pi [36].

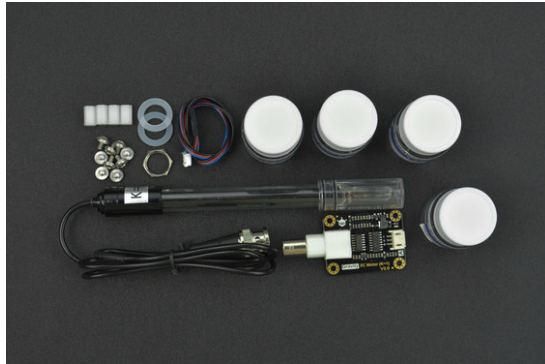


Figura 3.5: Sensor de Condutividade Elétrica [37].

### 3.2.3 Sensor de Temperatura da Água

Esta é uma versão à prova de água do sensor de temperatura DS18B20. Útil para quando precisa para medir algo distante, ou num ambiente molhado. Este sensor suporta até 125°C uma vez que o cabo é revestido em PVC mas na documentação é sugerido que o mantém abaixo de 100°C. A qualidade do sinal do sensor não se degrada com a distância uma vez que é digital [38]. O DS18B20 que esta representado na Figura 3.6, foi escolhido por ter estas capacidades referida anteriormente, porque desta maneira é possível medir a temperatura da água sem hesitação.

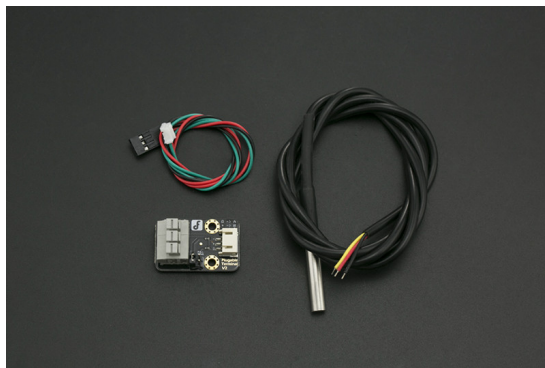


Figura 3.6: Sensor de temperatura da água DS18B20 [39].

### 3.2.4 Sensor DHT11

Este sensor DHT11 possui uma saída de sinal digital calibrada com o complexo de sensores de temperatura e humidade do ambiente. Sua tecnologia garante alta confiabilidade e excelente estabilidade a longo prazo. Este inclui um elemento resistivo e um sensor de medição de temperatura húmida NTC. Tem excelente qualidade, resposta rápida, capacidade anti-interferência e vantagens de desempenho de alto custo, desta maneira foi escolhido este sensor, que está representado na Figura 3.7. Estes sensores ajudam a refletir sobre os dados do ambiente com relação à água [40].

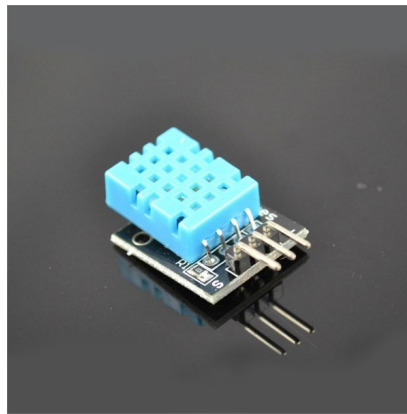


Figura 3.7: Sensor DHT11 [40].

## 3.3 Caixa Protetora

Nesta seção foi observado a Caixa em que protege a outra caixa, que se encontra o circuito, foi colocada uma caixa dentro da outra para garantir uma proteção elevada.

### 3.3.1 Caixa Protetora IP66

Esta caixa protetora representada na Figura 3.8, é a caixa principal à que protege a segunda caixa onde se encontra o circuito, onde este está protegida de água, humidade

elevada, e a temperatura muito elevada [41].



Figura 3.8: Caixa Protetora IP66 [41].

### 3.3.2 Caixa Protetora IP55

Esta caixa protetora representada na Figura 3.9, é a caixa onde se encontra o circuito, onde este impede toda a malícia que pode danificar o circuito [42].

## 3.4 Software

Nesta seção foi observado os softwares que foram utilizados para configurar e programar as ferramentas descritas acima.

### 3.4.1 *BalenaEtcher*

Este software é usado para instalar SO nos dispositivos de armazenamento, em que este foi instalado Linux no Gateway, para ser feito a configuração desejada. Foi escolhido este software porque é de fácil acesso e de rápido processo de instalação do SO, na Figura 3.10 está representada o logótipo do software *BalenaEtcher* [43].



Figura 3.9: Caixa Protetora IP44, IP55, IP56 [42].



Figura 3.10: *BalenaEtcher* Logótipo [43].

### 3.4.2 Plataforma WEB TTN

Esta plataforma serve para podermos ter dados na nuvem, onde estes criam-se dispositivos virtuais que são ligados aos dispositivos físicos por meios de endereços de seção de informações, assim como os Gateways também são conectados à plataforma TTN. Esta plataforma foi escolhida por ser a mais usada para o uso da tecnologia LoRa, é uma ferramenta *Open Source* (código aberto) e de fácil configuração. Na Figura 3.11 apresentamos o logótipo da plataforma [44].



Figura 3.11: *The Things Network* Logótipo [44].

### 3.4.3 Arduino IDE

O Software Arduino de código aberto (IDE) facilita a gravação de código e o upload para a placa. Este software pode ser usado com qualquer placa Arduino. Este software é um dos mais usados em todo o mundo para o uso do Arduino, existem diversas bibliotecas para este software, por este motivo foi escolhido. Na Figura 3.12 esta representado o logótipo da ferramenta [45].

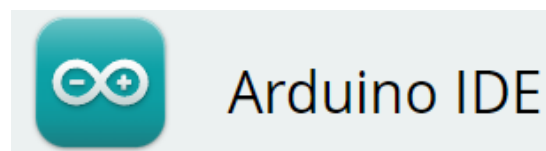


Figura 3.12: Arduino Logótipo [45].

### 3.4.4 Unity 3D

Este software é conhecido como um software capaz de criar ambientes 3D como jogos entre outras coisas, este foi escolhido pela sua capacidade de criar imagens ou ambientes 3D, desse modo será usado para criar o ambiente desejado [46]. Na Figura 3.13 esta representada o logótipo do software.



Figura 3.13: Unity Logótipo [46].

## 3.5 Tecnologia

Nesta seção estão descritas as tecnologias usadas para chegar o resultado desejado.

### 3.5.1 JavaScripts

*JavaScript* (às vezes abreviado para JS) é uma linguagem leve de multi-paradigma e dinâmica, interpretada e baseada em objetos [47]. Este foi escolhido por ser simples, foi utilizado para fazer o servidor de recepção de dados e cálculo de previsão por média, na Figura 3.14 demonstra o logótipo da linguagem.



Figura 3.14: *JavaScript* Logótipo [48].

#### Algumas Bibliotecas Usadas

Estas bibliotecas foram usadas para a comunicação do sistema, que são as seguintes:

1. **VUE.js** é um *framework Javascript open source*, e muito utilizado para criar aplicações de *single page* (página única) e também para desenvolver vários tipos de interfaces, que possuem necessidades de maior interação e experiência mais valorosa para o utilizador . É importante frisar que *Vue.js* se enquadra em *framework Javascript* progressivo, isto é, *Vue.js* pode ser conectado numa aplicação server-side que precisa otimizar a interface de utilizador [49] (o logótipo é apresentado na Figura 3.15).



Figura 3.15: *Vue.js* Logótipo [50].

2. O ***Embedded JavaScript Templating (EJS)*** é uma *Template Engine* que se pode utilizar em aplicações Node.js de forma dinâmica sem depender das limitações do *HTML*. Por outro lado, é um dos mecanismos de modelo mais populares para *JavaScript*, e como o nome sugere, nos permite incorporar código *JavaScript* em *HTML* [51]. A Figura 3.16 apresenta o logo do EJS .



Figura 3.16: EJS Logótipo [52].

3. ***AXIOS.js*** é um *cliente HTTP* modesto para o navegador, e *node.js*, fornece uma interface modesta muito extensível [53], a Figura 3.17.



Figura 3.17: *AxiosJS* Logótipo [54].

4. ***SOCKET.IO*** é uma Interface de Programação de Aplicações (API) que cria uma interface entre o *software*. É uma biblioteca *JavaScript* orientada a eventos para

aplicativos da Web em tempo real. A aplicação permite a comunicação bidirecional em tempo real entre clientes e servidores web, tendo portanto duas partes: uma biblioteca do lado do cliente que é executada no navegador e uma biblioteca do lado do servidor para *Node.js*. Ambos os componentes têm uma API quase idêntica [55]. Na Figura 3.18, apresenta o logo do Socket.IO.

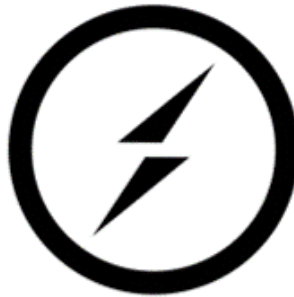


Figura 3.18: Socket.io Logótipo [56].

### 3.5.2 Node.js

É um interpretador de *JavaScript* assíncrono com código aberto orientado a eventos, criado por *Ryan Dahl* em 2009, focado em migrar a programação do *JavaScript* do cliente (*frontend*) para os servidores, criando aplicações de alta escalabilidade (como um servidor *web*), manipulando milhares de conexões/eventos simultâneas em tempo real numa única máquina física. O *Node.js* (ambiente de execução *JavaScript* no servidor) foi implementado baseado no interpretador V8 *JavaScript Engine* (interpretador de *JavaScript* em C++ com código aberto do *Google*, utilizado no *Chrome*), com desenvolvimento mantido pela fundação *Node.js* em parceria com a *Linux Foundation*, conforme Figura 3.19 [57].



Figura 3.19: *Node.js* Logótipo [58].

### 3.5.3 Prisma

Prisma é um Object-Relational Mapping (ORM), em português, mapeamento objeto-relacional, de última geração de código aberto [59]. Esta dividido em seguintes partes:

- **Cliente Prisma:** Construtor de consultas gerado automaticamente e com segurança de tipo para Node.js e TypeScript;
- **Prisma Migrar:** Sistema de migração;
- **Prisma Studio:** GUI para visualizar e editar dados em seu banco de dados

Todo projeto que usa uma ferramenta do kit de ferramentas Prisma começa com um arquivo de esquema Prisma. O esquema prisma permite que os desenvolvedores definam seus modelos de aplicativos em uma linguagem intuitiva de modelagem de dados [59]. Na Figura 3.20 está representado o logótipo do prisma.



Figura 3.20: Prisma Logótipo [59].

### 3.5.4 MQTT

Esta ferramenta tecnológica foi escolhida, segundo o que está descrito no capítulo 2, conhecida como *Message Queuing Telemetry Transport*, usualmente designado por *MQTT* nos restantes dos países, é um protocolo de mensagens leve para sensores, e pequenos dispositivos móveis otimizado para redes *TCP/IP*. O procedimento de troca de mensagens, e é fundamentado no modelo Publicador-Subscritor, extremamente simples e leve, está tecnologia esta mesmo por dentro da plataforma TTN onde estes o usam para trocas de mensagens, deste modo, foi possibilitado a receção dos dados que foram enviados a plataforma [60]. Na Figura 3.21, apresenta o logótipo do *MQTT*.



Figura 3.21: MQTT Logótipo [61].

# Capítulo 4

## Plataforma Smart River

Neste capítulo foi feita uma abordagem, análise e modelação usando diagramas, mapas, imagens de circuitos, ou tabelas para perceber as dificuldades associadas à resolução do problema, deste modo chegar aos resultados coerentes.

### 4.1 Localização

A plataforma Smart River é um módulo interativo que consiste em sensores ligados através da rede, que recolhem dados sobre os parâmetros ambientais do rio principal de Bragança - rio *Fervença*. A Figura 4.1 ilustra o posicionamento dos sensores na região. Os sensores estão localizados em pontos estratégicos que permitem a recolha de dados em quatro ( $n = 4$ ) pontos diferentes:

1. antes de chegar à cidade de Bragança (perto da vila de Castro de Avelãs),
2. na cidade de Bragança no Instituto Politécnico de Bragança (IPB),
3. na cidade de Bragança na Casa-da-Seda, edifício do Centro de Ciência Viva de Bragança,
4. depois da cidade de Bragança, junto à ETAR do Município.

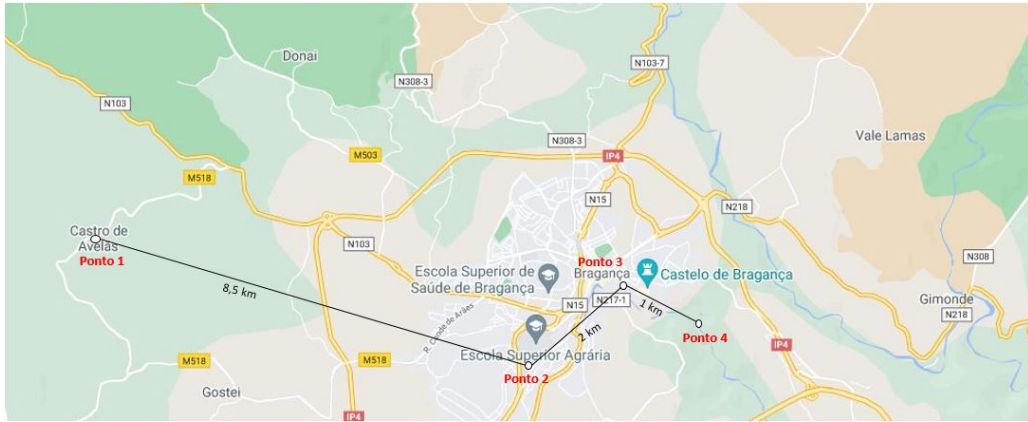


Figura 4.1: Localização dos sensores [26].

O posicionamento estratégico dos sensores permite avaliar e correlacionar os dados recolhidos com a atividade urbana, e assim propor soluções que melhorem a qualidade da água do rio.

## 4.2 Estrutura

Nesta seção será abordada a estrutura da plataforma em que na primeira sub-seção está centrado no circuito eletrónico, em que este é o corpo do projeto, e a segunda sub-seção que seria a imagem do projeto onde será mostrado os dados, a interação, e significados destes dados obtidos do rio.

### 4.2.1 Dispositivo

Para medir a qualidade da água do rio Fervença, foi necessário analisar o circuito que será montado, para poder ter uma eficiência boa e eficaz, pensando no consumo dos sensores, e quando os técnicos podem ir dar manutenção no dispositivo. Deste modo a Figura 4.2, representa o circuito criado. A área número (1) se refere ao microcontrolador, este está alimentado por uma bateria de lítio na área (2). Tem também um conversor Corrente Contínua (CC)-CC 5V-3.3V, que está na área (3), porque os sensores podem ser alimentados desde 3V á 5V, deste modo foi convertido os 5V para 3V para aumentar

a capacidade de duração da bateria. Observar-se na área quatro(4) o transmissor LoRa RFM95 para comunicar com a plataforma TTN através de um Gateway. A seguir na área cinco(5) tem pinos analógicos para ligar os sensores. Na última área é uma das mais necessárias são os pinos de ligações RX, TX, RST e GND para poder compilar e enviar o programa ao microcontrolador.

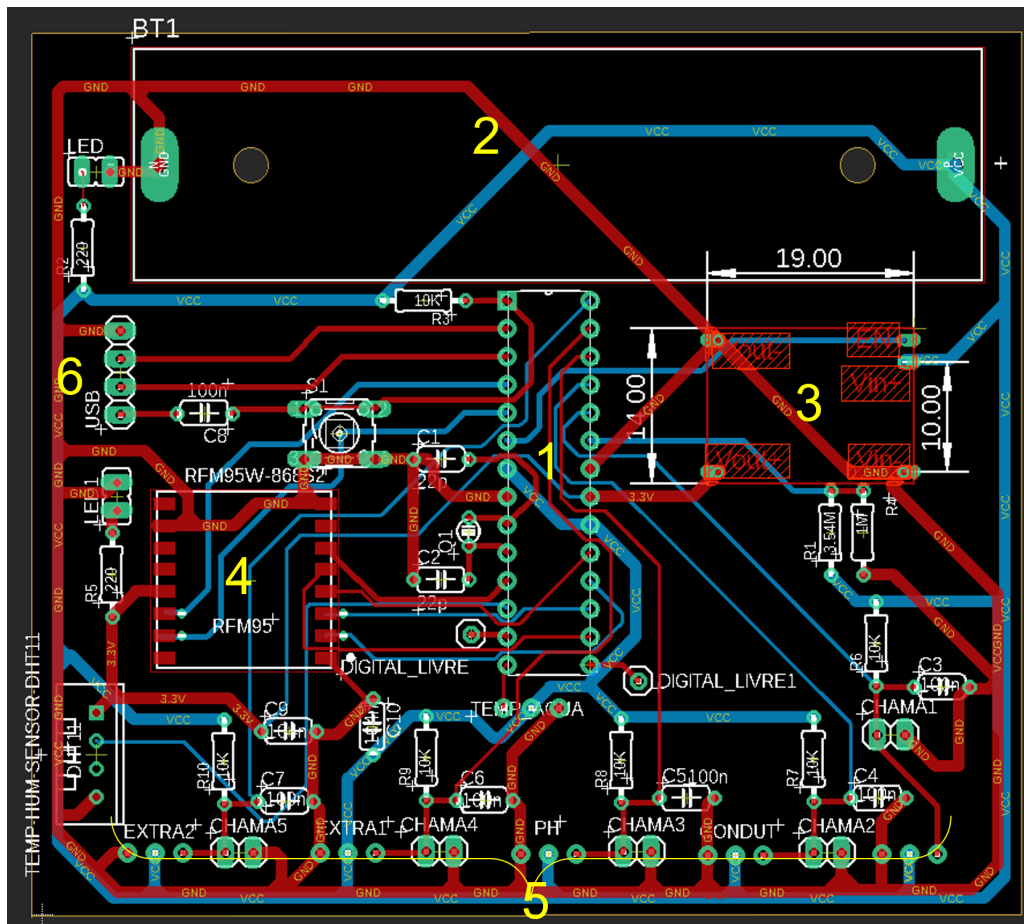


Figura 4.2: Circuito [26].

Os componentes foram seleccionados levando em consideração o custo de aquisição (evitando sofrer vandalismo ou danos externos), fácil acesso aos dados um sistema externo (por exemplo, um microcontrolador) e robustez (como é uma aplicação que requer a exposição de sensores a condições ambientais o ano todo). A estrutura para ser completada foi idealizada duas caixas de proteção do circuito ilustrado na Figura 4.2 e também um

tubo PVC para fazer a proteção dos cabos dos sensores.

## 4.2.2 Exposição Interativa

Para a composição completa da estrutura do sistema foi criado um ambiente em Unity 3D, onde foi disponibilizado pelo designer algumas imagens para poder ser criado o ambiente, isso foi de grande ajuda para não dificultar na programação, as imagens disponibilizadas são:

- **Separadores Smart River/Página Principal** → Figura 4.3 diz respeito a imagem com os quatro pontos falado na seção 4.1, onde o primeiro é o Castro de Avelãs, o segundo é o IPB, o terceiro é a Casa da Seda, o quarto e o último a Estação de Tratamento de Águas Residuais (ETAR).



Figura 4.3: Separadores Smart River[62].

- **Fundo de apresentação gráfica/Página Secundária** → Figura 4.4 é onde se pode visualizar os gráficos com valores em média por semana, por trimestres, informações dos dados, e algumas informações de propriedades químicas.

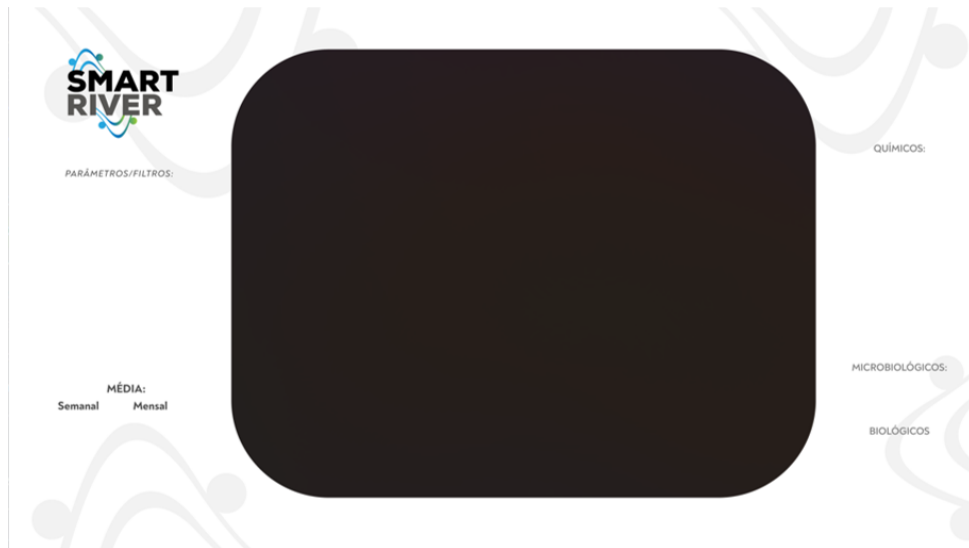


Figura 4.4: Fundo de apresentação gráfica[62].

- **Botões circulares** → Figura 4.5 representa os tipos de botões que será usado, onde tem-se o botão de fechar o separador de informação que for aberto, o de seguinte/anterior para ir a data ou semana seguinte e vice-versa, tem também o botão de atualizar, que seria de atualizar o separador, tem os dois botões de informações, o azul é a informação dos dados, o que esta com info é de informação da localização, e por último tem-se o botão de casa que seria voltar para o separador inicial.

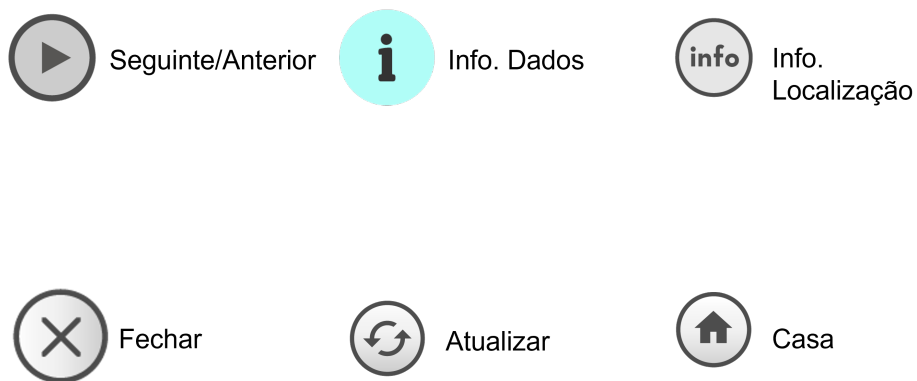


Figura 4.5: Botões Circulares[62].

- **Botões personalizados** → Figura 4.6 representa os tipos de botões personalizados, em que, o botão da casa da seda, e assim também será para as outras localizações,

ao clicar nele será enviado para o separador secundário com os dados daquela localização não esquecendo que o separador secundário terá uma label ou seja uma tela de indicação do local, tem o botão de oxigênio dissolvido onde mostra a informação do mesmo lembrando que também terá outros botões com essas similaridades, e por fim tem o botão de humidade relativa do ar onde este, ao ser clicado aparece um gráfico da humidade relativa do ar, também tem-se outros botões de dados com essas similaridades.



Figura 4.6: Botões Personalizados[62].

- **Separadores de informações** → Figura 4.7 está relacionado com o que foi dito no ponto anterior, esta figura representa os separadores de informações.

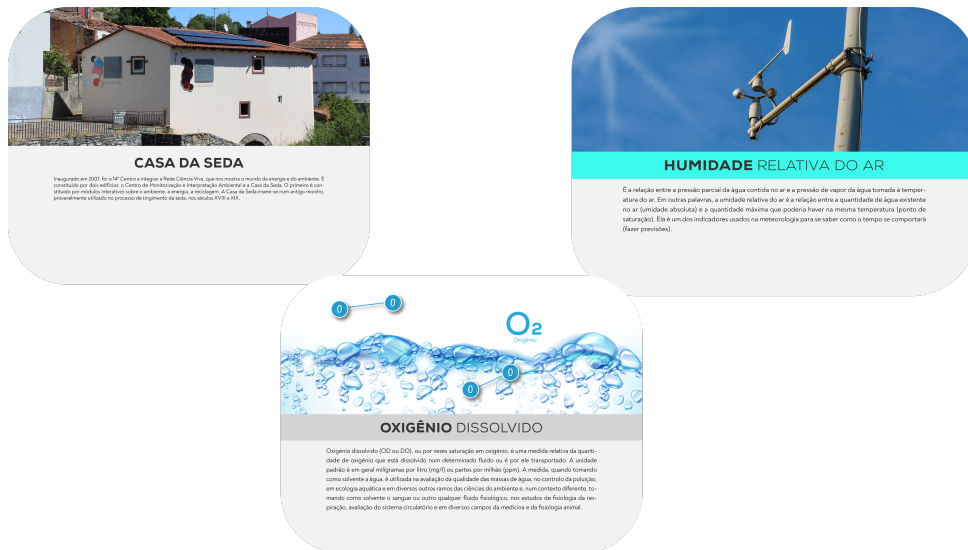


Figura 4.7: Separadores De Informações[62].

## 4.3 Comunicação

Nesta seção vai ser abordado como que será feita a comunicação do dispositivo com a plataforma TTN e suas configurações, e como recebem estes dados no computador através da plataforma TTN.

### 4.3.1 Configuração da Plataforma

Segundo a documentação, é necessário, registar um Gateway, uma Aplicação, e os dispositivos dentro da aplicação, para está configuração tem-se de ir a plataforma TTN e configurar da seguinte maneira:

1. Adicionar um Gateway;
2. Adicionar uma Aplicação;
3. Adicionar os Dispositivos dentro da Aplicação [14].

**OBS:** Estas configurações são feitas seguindo a documentação disponibilizada no "site : <https://www.thethingsnetwork.org/docs/lorawan/>", foi simplificada a descrição neste documento uma vez que essa informação está disponibilizada pela TTN.

Depois de realizar estas configurações são estabelecidos alguns parâmetros para comunicação do dispositivos com a plataforma para finalizar a configuração, que são, a *Network Session Key () (NWKSKEY)* ou em português (**Chave e Sessão de Rede()**) que é usada para interação entre o Nó e o Servidor de Rede. Esta chave é usada para validar a integridade de cada mensagem pelo código de integridade da mensagem (verificação MIC) [14]. Este MIC é semelhante a um cheque, exceto que evita adulteração intencional de uma mensagem [14]. Para isso, a LoRaWAN usa *Advanced Encryption Standard-Cipher-based Message Authentication Code (AES-CMAC)*. No *backend da The Things Network* esta validação também é usada para mapear um endereço de dispositivo não único () para um único e a *App Session Key () (APPSKEY)* ou em português (**Chave de Sessão de Aplicação ()**) é usada para criptografia e descriptografia da carga útil [14]. A carga é totalmente criptografada entre o Nó e o componente Manipulador/Servidor de Aplicativos da Rede de Coisas (que poderá executar em seu próprio servidor). Isso significa que ninguém, exceto , é capaz de ler o conteúdo das mensagens que envia ou recebe [14]. Estas duas chaves de sessão são únicas por dispositivo, por sessão. Se ativar dinamicamente o seu dispositivo (*OTAA*), essas chaves são geradas em todas as ativações [14]. Se ativar estaticamente seu dispositivo (*ABP*), essas chaves permanecerão as mesmas até que as altere. Foi também verificado no código o *Device Address (DEVADDR)* ou em português(**Endereço do Dispositivo ()**) , de trinta e dois(32) bits (não único), onde estes identificamos os dispositivos que enviaram a mensagem [14].

### 4.3.2 Comunicação do Dispositivos com a Plataforma

A comunicação do dispositivos com a plataforma foi realizada através de três componentes essencial o Transmissor LoRa RFM95, Gateway RAK7243C e o microcontrolador. O transmissor funciona como um comunicador de espectro disperso que seria transmissão de dados digitais em radiofrequências tendo um alcance de longa distância e específico de largura de banda, ele comunica com o gateway que deixa passar a informação como meio de conexão com a TTN, todos os gateways ao alcance de um dispositivo receberão

as mensagens do dispositivo e as encaminharão para a TTN. A rede eliminará mensagens duplicadas e selecionará o melhor gateway para encaminhar as mensagens na fila para o *downlink*, o microcontrolador está como a base do envio, porque ao programar o *firmware*, é usado a NWKKEY, APPSKEY e a DEVADDR para saber qual dispositivo enviou os dados. Na Figura 4.8 ilustra a comunicação do dispositivo com a plataforma.

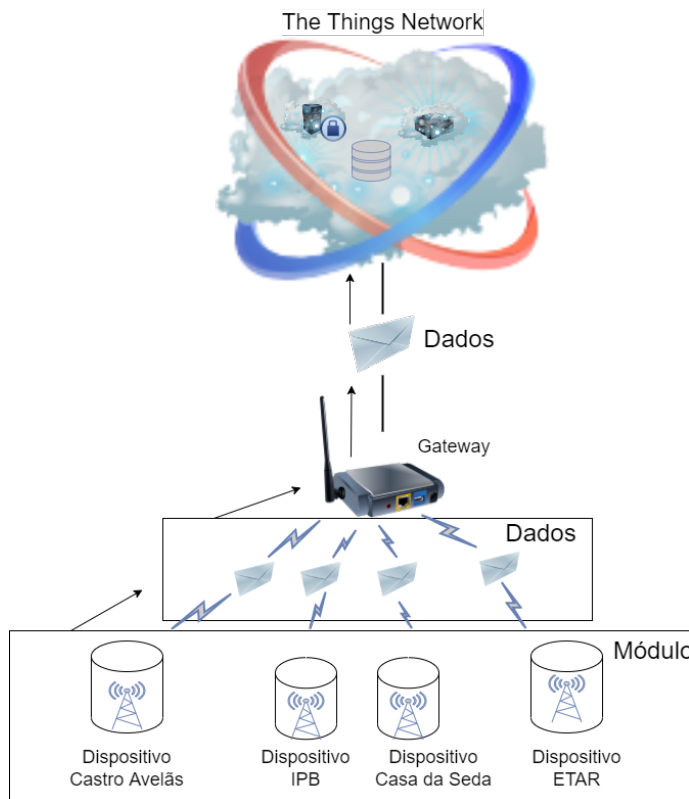


Figura 4.8: Comunicação do dispositivos com a plataforma [26].

### 4.3.3 Comunicação da Plataforma com o Software de Interação

Na comunicação da Plataforma com o Software de Interação, sabendo que a TTN já tem um servidor MQTT, que o *Things Network* usa para publicar ativações e mensagens de dispositivos, mas também permite que publique uma mensagem para um dispositivo específico em resposta, deste modo foi necessário criar um cliente Mqtt no Node.js programado por JavaScript para receber os dados da plataforma TTN e transformar estes dados no

arquivo .txt, para ser trabalhado na aplicação Unity3D e apresentado graficamente. Na Figura 4.9 representa a comunicação da Plataforma com o Software de Interação.

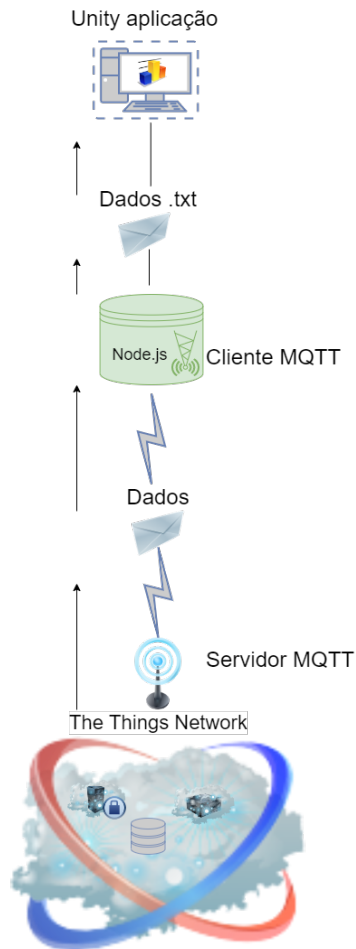


Figura 4.9: Comunicação da Plataforma com o Software de Interação [26].

#### 4.3.4 Esquema de Comunicação

De acordo com as subseções 4.3.2 e 4.3.3, pode-se verificar o seguinte esquema que está apresentado na Figura 4.10, observa-se que o módulo envia os dados como mensagem por meio do Transmissor LoRa e o gateway deixar passar as mensagens eliminando as que estão duplicadas. Estes dados comunicam com plataforma TTN, que está preparada para receber estes dados de acordo com as **Chaves de Sessões e Endereço de Dispositivo**, e é disponibilizado pelo servidor MQTT, onde estes publicam as ativações e mensagens

dos dispositivos, em que o cliente recebe estes dados como cliente assinantes que foi criado no Node.js, e disponibilizam os dados num arquivo .txt para a aplicação criada no Unity apresentar em gráficos.

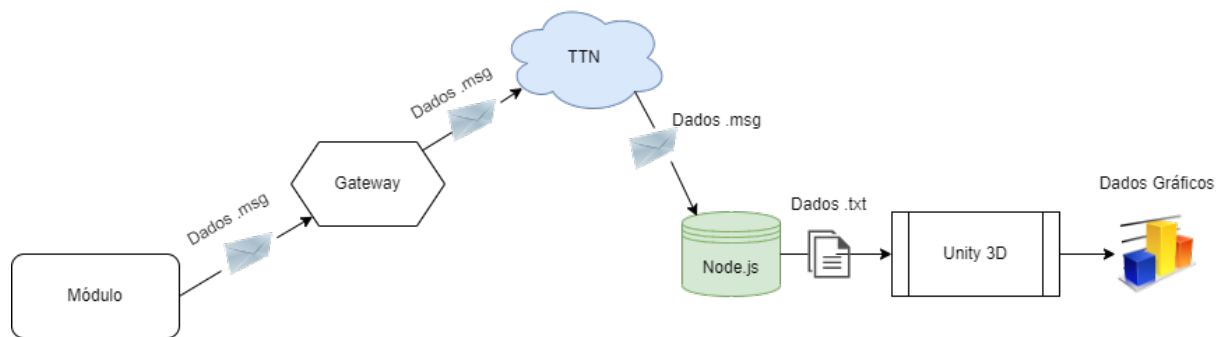


Figura 4.10: Esquema de Comunicação [26].

## 4.4 Análise de Dados

O dados que são fornecidos foram estabelecidos para serem enviados de uma hora (1h) a uma hora e meia (1h30), ou seja, os dados são enviados e o dispositivo fica a dormir durante o restante tempo. Os dados enviados são: pH, Condutividade elétrica da água, humidade relativa do ar, temperatura ambiente, bateria, e a temperatura da água. Isso realizado em cada localização descrita na seção 4.1. Ao obter estes dados em tempo real, foram realizadas algumas operações em JavaScript e em Unity 3D.

A Tabela 4.1 diz respeito ao exemplo de como os dados chegam, por exemplo quando chega um dado verifica-se a localização e qual dados são, depois cria-se o arquivo .txt na ordem certa, e de acordo com a data e a hora organiza-se os dados por ordem de chegada do mais recente para o antigo.

**OBS:** Os Dados que consta na tabela é um exemplo criado aleatoriamente para exprimir como os dados são separados de acordo com cada dispositivo, lembra-se que os dados de cada localização vêm por completo todos os cinco dados de uma vez e o estado da bateria.

Dados	Localização	Data	Hora	Valor
pH	Seda	dd/mm/aa	hh/mm	###
Condutividade(eC)	Castro	dd/mm/aa	hh/mm	###
Humidade R. Ar	IPB	dd/mm/aa	hh/mm	###
Temperatura Amb.(°C)	ETAR	dd/mm/aa	hh/mm	###
Temperatura Água(°C)	Seda	dd/mm/aa	hh/mm	###
Bateria	IPB	dd/mm/aa	hh/mm	###

Tabela 4.1: Tabela de Envio de Dados

#### 4.4.1 Cálculos em JavaScript

Nesta subsecção será representado os cálculos realizados para previsão de dados por média que foi feita em JavaScript, e os cálculos por semana e trimestral feita em Unity para apresentação gráfica.

##### JavaScript Cálculo de Previsão Por Média

Este cálculo será realizado por causa do tempo de espera de manutenção dos dispositivo, ou seja na hora de trocar a bateria. Uma vez que a bateria esgotar automaticamente, em que a condição estipulada, é que, se for menor que três é enviada uma mensagem no email do supervisor. Desse modo ele deve ir trocar a bateria do dispositivo, porém alguns dispositivos se encontram em uma distância longa de acordo com a localização de onde o supervisor trabalha. Nesse caso ao demorar um ou mais dias não vai haver envio de dados, para está solução foi aplicado este cálculo de previsão por média em que é verificado que se em duas horas não for enviado nenhum dados da plataforma, a aplicação em JavaScript gera uns dados fazendo a média dos dados que já existem.

Para este feito foi criado um banco de dados no servidor da Prisma Studio, onde este guarda os dados dos sensores localmente, onde é possibilitado pegar estes dados e fazer cálculos, foi utilizado o Socket.io para criar um evento dessa maneira saber-se-a, se em cada duas horas chegam os dados e se nesse instante estes dados não forem enviados, automaticamente é gerado o dado de previsão por média, a expressão usada se encontra na linha nº 4.1. A expressão para os outros dados, são a mesma que está apresentada na linha nº 4.1.

$$pH_{seda} = \frac{\sum pH_{por.30dias}}{numero.total.de.dados.pH.nos.30dias} \quad (4.1)$$

Salienta-se que o Socket.io também ao verificar a chegada de dados, o código está preparado para criar um arquivo .txt e verificar se já existe o arquivo com este nome, depois colocar ali os dados consoante o dispositivo de cada localização, ou seja o Socket.io nos permite criar um evento desse modo trabalhar tão rápido assim que houver algum evento acontecendo.

#### 4.4.2 Unity 3D Cálculo

A análise feita para a visualização de dados em Unity foi consoante a semana, e por trimestre, porque ao apresentar os dados por hora seria necessário atualizar sempre o separador secundário para apresentar os dados. Dessa maneira foi criado uma expressão que calcula os dados semanalmente e trimestral, estes dados são apresentados por semana um(1), semana dois(2)..., semana quatro(4), e primeiro(1º) trimestre, segundo(2º) trimestre ..., quarto(4º) trimestre.

A expressão para o cálculo semanal é o seguinte: pelo menos dez(10) dados num dia a aplicação calcula a média e apresenta este dados em média do primeiro dia da primeira semana, isto porque foi declarado que não se pode esperar até completar a semana toda para apresentar os dados. Ou seja, assim que completa a semana um é feita a média de todos os dias daquela semana. Seguidamente é feito o somatório de todos os dados em sete(7) dias dividido por número total de dados recolhidos em sete dias. Segundo ao cálculo trimestral dos dados é feito o seguinte, enquanto não completar os três meses para ter o primeiro trimestre nenhum dados são apresentados, bom o cálculo trimestral segue a seguinte expressão na linha nº 4.2, nota que os cálculos realizados para os trimestres restante seguem o mesmo padrão:

$$pH_{seda} = \frac{\sum pH_{por.3meses}}{numero.total.de.dados.pH.por.três.meses} \quad (4.2)$$

## 4.5 Node.js Supervisão

Mesmo tendo uma aplicação que será realizada em Unity 3D foi necessário uma aplicação aplicada ao WEB, porque o tempo de espera para aplicação ser feita em Unity 3D é um pouco estendido, desse modo foi discutido para ter uma aplicação de supervisão e de interação substituta ou reserva, sendo assim usando o conhecimento de JavaScript e suas bibliotecas (VUE.js, EJS, AXIOS.js e SOCKET.IO) analisou-se um seguinte designer para sua implementação. Na Figura 4.11 diz respeito á aplicação que será realizada através do Node.js, em que tem-se basicamente a data e a hora do dia atual com uma imagem no lado superior esquerdo, no lado superior direito tem os quatro pontos onde ao clicar apresentará os dados daquele ponto, depois tem no lado inferior esquerdo os dados deste ponto com o seu valor, e tem também um gráfico de rosca no lado inferior direito.

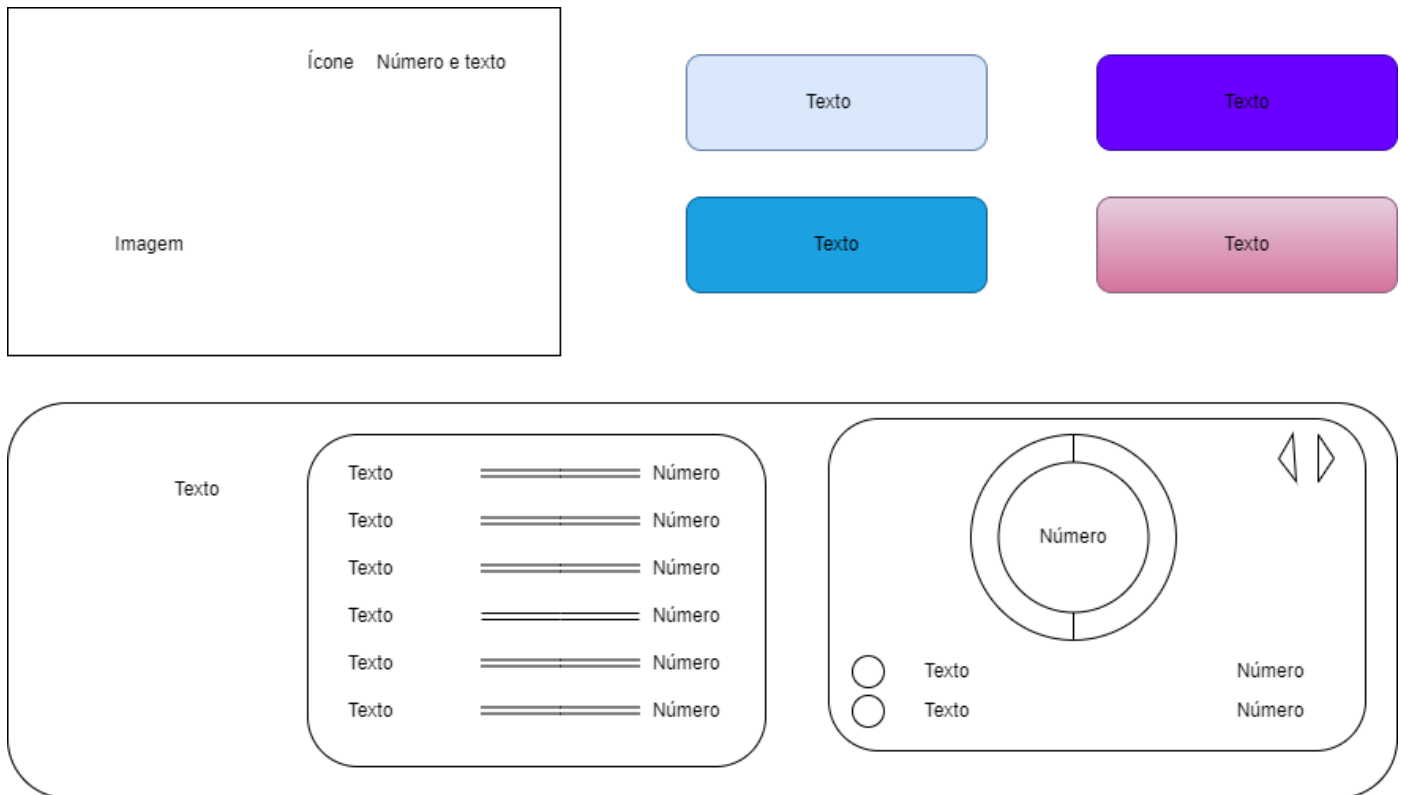


Figura 4.11: Designer do Sistema de Supervisão [26].

**OBS:** Este designer com o tempo pode ser atualizado para uma outra versão, por isso

foi criado para suprir a necessidade da aplicação que será realizada no Unity 3D.



# Capítulo 5

## Teste, Avaliação e Discussão

Este capítulo apresenta os testes realizados para verificar que o projeto desenvolvido cumpre os objetivos assumidos e resolve, de facto, o problema descrito, salientando os pontos mais relevantes da mesma, dificuldades encontradas ou soluções técnicas inovadoras desenvolvidas ou aplicadas.

### 5.1 Dispositivo Implementado

Nesta seção demonstra-se a *Printed Circuit Board (PCB)* ou em português (Placas de Circuito Impresso (PCI)), também foi observados algumas imagens de códigos importantes para o uso do dispositivo, e a imagem do teste deste.

#### 5.1.1 Códigos Relevantes

Nestes códigos que serão apresentados foram usadas algumas bibliotecas como `LowPower.h` que seria para economia da bateria, o `lmic.h` e o `hal/hal.h` para comunicação com a TTN, utilizou-se `SPI.h` para comunicação LoRa.

## Envio de mensagem

Nesta função *dosend* envia-se os dados através das portas TXRX, onde é verificado, se este, está enviando. A primeira vez que o dispositivo é ligado este não envia os dados simplesmente é verificado se ele consegue enviar, os dados dos sensores são enviado depois de ligar o lora RFM95, depois são lidos os sensores, e assim, é enviado o *mydata* um vetor com os dados, feito isso ele volta a dormir 1 hora. A Figura 5.1 observa-se o código de envio.

```
void do_send(osjob_t* j) {
    // Check if there is not a current TX/RX job running
    if (LMIC.opmode & OP_TXRXPEND) {
        Serial.println(F("OP_TXRXPEND, not sending"));
    } else {
        delay(1000);
        lmicSeqNumber = LMIC.seqnoUp;
        //desliga_rfm95();

        read_sensors();
        sleeping();

        //início*****

        //Religando o RFM95
        liga_rfm95();
        LMIC.seqnoUp = lmicSeqNumber;
        delay(1500);
        LMIC_setTxData2(1, mydata, sizeof(mydata), 0);
        Serial.println(F("Packet queued"));
    }
}
```

Figura 5.1: Código de envio [26].

## Função *sleeping*

Esta função foi criada para aumentar a durabilidade da Bateria, onde este desliga o LoRa RFM95, depois desliga o pino 4 que é do abaixador de tensão para desabilitar o mesmo para não haver consumo dos sensores entre outros, isto foi feito só para garantir que não vai haver mesmo nenhum consumo dos sensores, depois verifica-se, se a tensão da bateria estiver menor que 3 volts coloca-se o dispositivo para dormir para sempre, isso para que a bateria não fique em baixo que não é muito bom para vida útil da bateria. Se a tensão

da bateria for maior que 3 volts, ele dorme 2 segundos e é feito um **for** onde a contagem vai de 0 á 450 vezes 8 segundos que é igual 3600 segundos que equivale a 1 hora, depois disso ele acorda e volta a ligar o pino 4 lendo os sensores. A Figura 5.2 está o código da função que foi implementada.

```
void sleeping() {
  desliga_rfm95();
  digitalWrite(4, LOW);
  delay(50);
  Serial.println("bat");
  byte i;
  if (Vbatt < 30) {
    Serial.println("bat<30");
    LowPower.powerDown(SLEEP_FOREVER, ADC_OFF, BOD_OFF);
  } else {
    Serial.println("bat>30");
    LowPower.powerDown(SLEEP_2S, ADC_OFF, BOD_OFF);
    delay(50);
    for (i = 0; i < 450; i++) {
      digitalWrite(4, LOW);
      LowPower.powerDown(SLEEP_8S, ADC_OFF, BOD_OFF);
      delay(50);
    }

    delay(50);
  }
  digitalWrite(4, HIGH);
  Serial.println("Acordou");
}
```

Figura 5.2: Função Dormir [26].

## Função Desligar

Esta função desligar é muito importante porque ela garante que o lora RFM95 desligue e deixe de consumir, onde é desligado a comunicação com o TTN, desliga-se a interface SPI e os restantes pinos que implica o consumo. A Figura 5.3 está o código da função que foi implementada.

```

void desliga_rfm95() {
  LMIC_shutdown();
  SPI.end(); // shutdown the SPI interface
  pinMode(SCK, INPUT);
  digitalWrite(SCK, LOW); // shut off pullup resistor
  pinMode(MOSI, INPUT);
  digitalWrite(MOSI, LOW); // shut off pullup resistor
  pinMode(MISO, INPUT);
  digitalWrite(MISO, LOW); // shut off pullup resistor
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
  digitalWrite(10, LOW);
  digitalWrite(11, LOW);
  digitalWrite(4, LOW);

  Serial.println("Desligou");
  Serial.print("pin4->");
  Serial.println(digitalRead(4));
}

```

Figura 5.3: Função Dormir [26].

### 5.1.2 PCI desenvolvida

Na Figura 5.4 está representada a placa de circuito impresso, com os componentes que foram soldados na placa, com as saídas Analógicas.

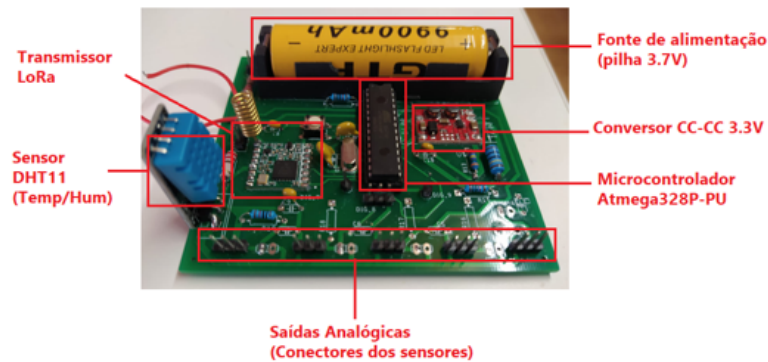
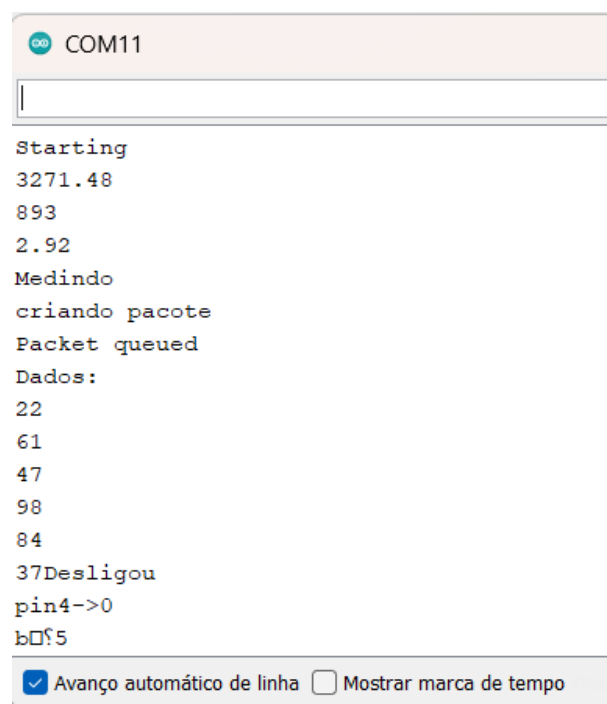


Figura 5.4: PCI [26].

## Teste do Dispositivo

Na Figura 5.5 apresenta-se o monitor série da IDE do arduino, na porta COM11, em que, ao ligar o dispositivo deu Início (*starting*) porque comunicou com o gateway, feito isso ele cria os pacotes que seria colocar os dados dentro do vetor *mydata*, depois disso o pacote é colocado em fila e enviado para TTN, que por sua vez o dispositivo adormece no intervalo de 1 hora á 1 hora 30 minutos.



```
COM11
Starting
3271.48
893
2.92
Medindo
criando pacote
Packet queued
Dados:
22
61
47
98
84
37Desligou
pin4->0
b0f5
 Avanço automático de linha  Mostrar marca de tempo
```

Figura 5.5: Monitor serial porta COM11 [26].

## 5.2 Dispositivo Virtual

Para ter o dispositivo virtual foi criada uma aplicação com um id, e o nome de Natureza\_virtual de acordo com a Figura 5.6. dentro desta aplicação foram criados 4 (quatro) *end devices* (dispositivos finais) que estão apresentado na Figura 5.7.

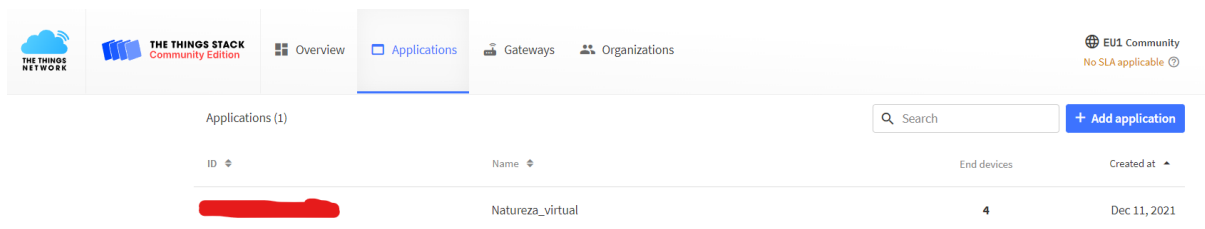


Figura 5.6: Aplicação [26].

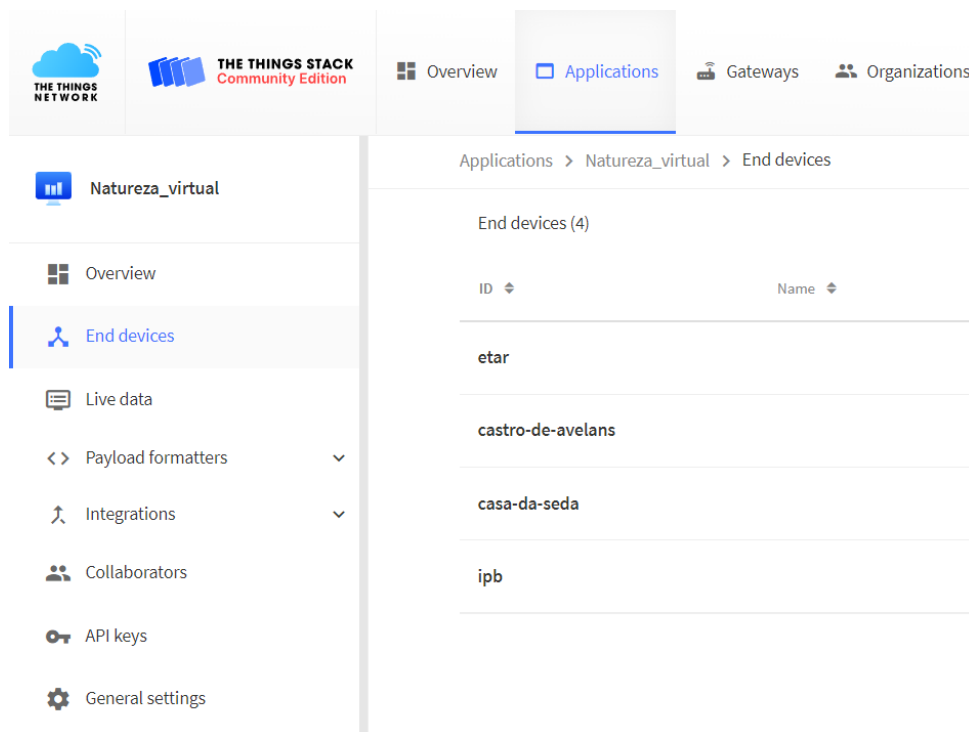


Figura 5.7: Dispositivos Finais [26].

### 5.3 Dispositivo Final

Após criar a PCI, os dispositivos virtuais, e implementado o código utilizando o NWKSKEY, APPSKEY e o DEVADDR, foi criado o dispositivo final onde foi colocado a PCI dentro de duas caixas de proteção, e os cabos do sensor passados em um tubo PVC. Na Figura 5.8 está representado o dispositivo final colocado no Rio com o logótipo do CCVB.



Figura 5.8: Dispositivo Final [26].

### 5.4 Dados Obtidos

Na Tabela 5.1 observa-se os dados dos sensores recolhidos em dois dias do mês de Outubro de 2022 em que estes dados estão sendo avaliados neste ano, que é um período de teste para depois ser definitivo.

Dispositivo	pH	TempH2O(°c)	TempAmb(°c)	HumRelativaAr(%)	Condut(uS/cm)	Data
C. Seda	8.1	18	19	62	110	10/10/2022
	8	20	24	49	104	11/10/2022
C. Avelãs	7.6	16	17	68	110	10/10/2022
	7.1	16	17	69	110	11/10/2022
ETAR	7.7	18	19	62	108	10/10/2022
	8	16	17	69	110	11/10/2022
IPB	8.1	16	17	71	130	10/10/2022
	7.7	16	17	71	116	11/10/2022

Tabela 5.1: Tabela de Dados Obtidos

## 5.5 Monitorização

Nesta seção foi observado a aplicação de supervisão substituta, que foi criada para apresentação, também foi avaliado a aplicação da exposição interativa.

### 5.5.1 Sistema de Supervisão Substituta

Na Figura 5.9, está representada a aplicação que foi criada para substituir a aplicação da exposição interativa durante o tempo do seu desenvolvimento.

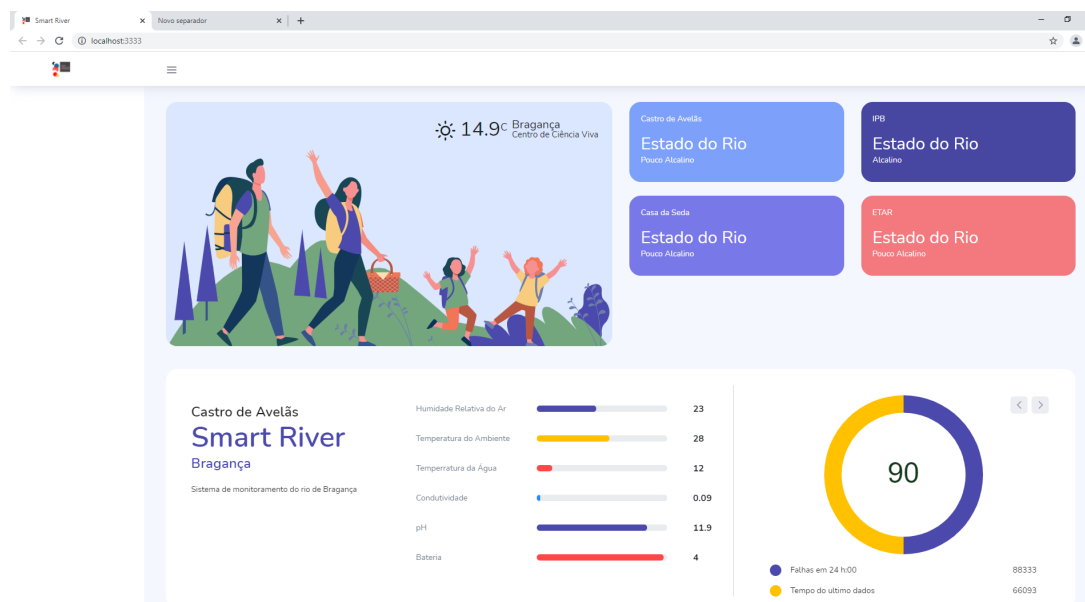


Figura 5.9: Sistema de Supervisão Substituta [26].

## 5.5.2 Aplicação de Exposição Interativa

Para esta aplicação, foram desenvolvidos dois separadores em que primeiro que esta na Figura 5.10, sendo o separador principal, que ao clicar na localização desejada é aberto o separador secundário.



Figura 5.10: Tela Principal da Aplicação [26].

## Separador Secundário

Na Figura 5.11 observa-se a média dos dados enviado do dispositivo da Casa da Seda em Duas semanas de Outubro.

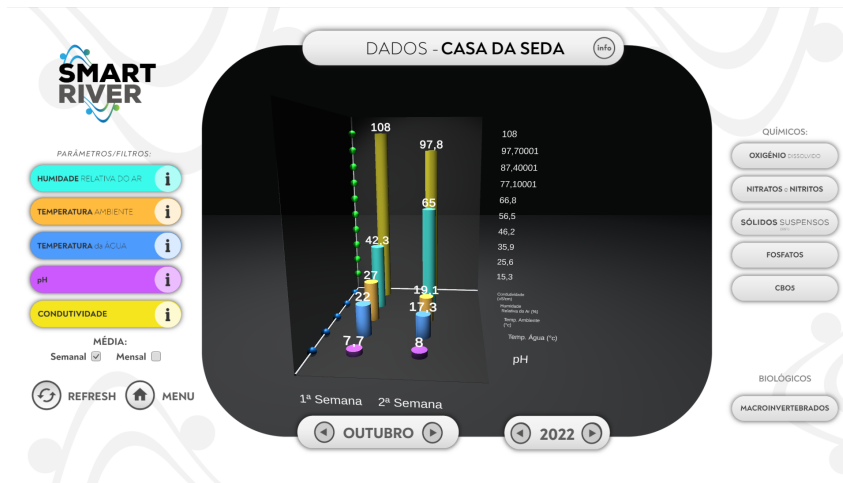


Figura 5.11: Tela Secundária de Dados-Casa da Seda [26].

A Figura 5.12 observa-se a média dos dados enviado do dispositivo da Castro de Avelãs em duas semanas de outubro.

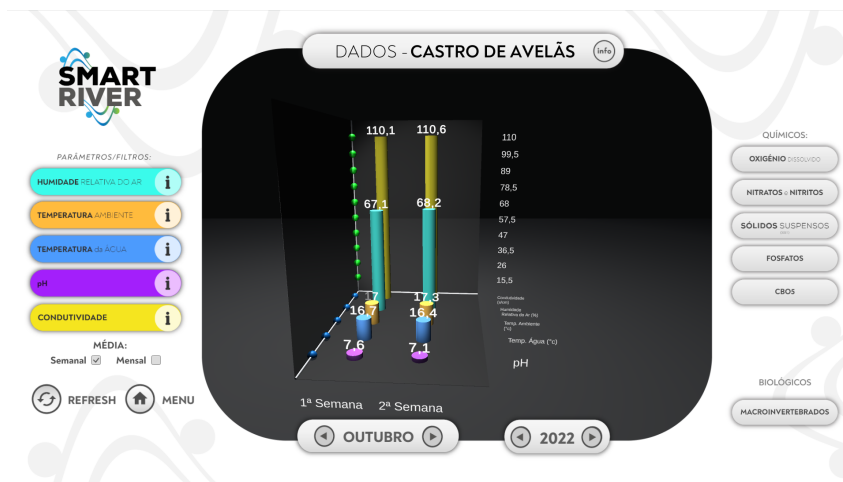


Figura 5.12: Tela Secundária de Dados-Castro de Avelãs [26].

Na Figura 5.13 observa-se a média dos dados enviado do dispositivo da ETAR em duas semanas de outubro.

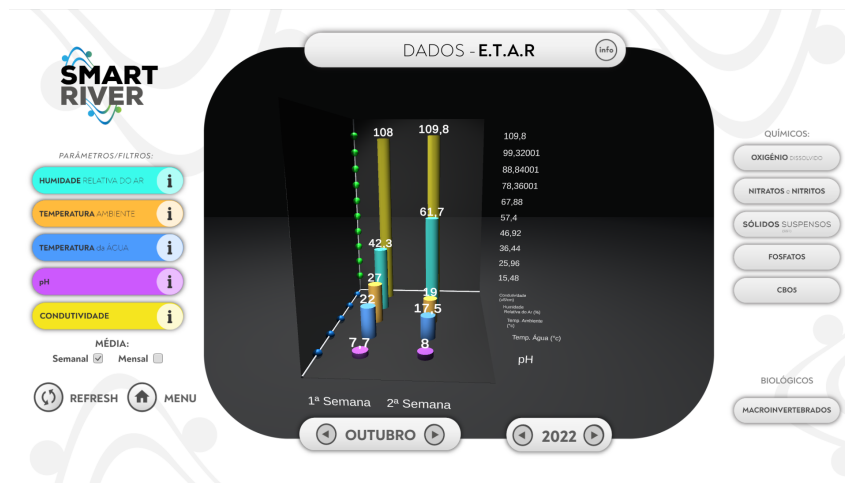


Figura 5.13: Tela Secundária de Dados-ETAR [26].

Na Figura 5.14 observa-se a média dos dados enviado do dispositivo da IPB em duas semanas de outubro.

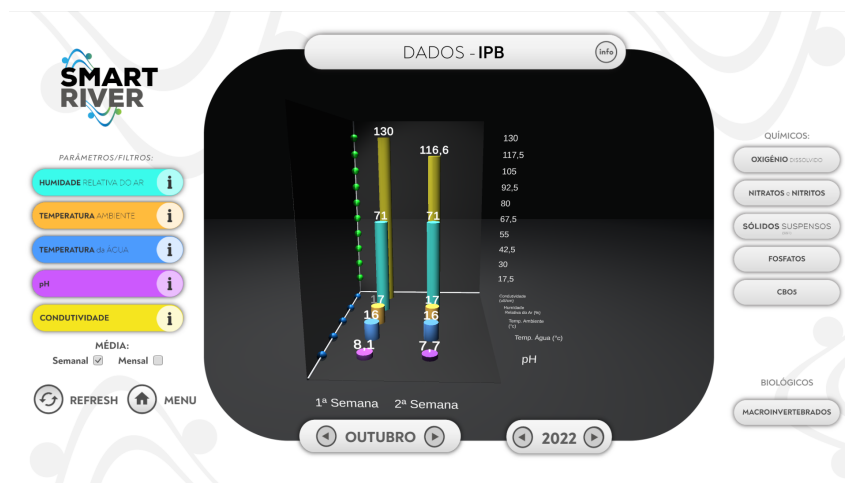


Figura 5.14: Tela Secundária de Dados-IPB [26].

## 5.6 Discussão

A plataforma desenvolvida e abordada nesta dissertação, tem como objetivo significativo para o resultado de um sistema que pode proporcionar a facilidade de aquisição de dados e identificar a qualidade da água do rio. Pode-se observar que não é difícil entender que há várias limitações, e posterior atualizações para serem tomadas.

### 5.6.1 Avaliação da Plataforma Smart River

Com relação a plataforma, o Centro Ciência Viva de Bragança, colocou este à Exposição para os Visitantes na Casa da Seda, como um teste desde Julho de 2021, para saber o retorno positivo e negativo do sistema.

Sendo assim depois de um ano e uns meses realizando a monitorização do sistema foi observado que alguns dos visitantes que não entendem muito de tecnologia ao ver o sistema não entendiam muito bem do porquê os dados do rio. Mais isso foi explicado pelos monitores do Centro Ciência Viva de Bragança que entenderam muito bem do assunto.

A parte positiva foi que os visitantes e os monitores do Centro Ciência Viva de Bragança acharam interessante a forma de recolha de dados, sem ter que dirigir ao rio, e a forma como é apresentado estes dados, e também observaram que os dados recolhidos mostram que a água está em condições normais, em que o pH demonstra-se no intervalo de 6 a 8, ou seja, não muito ácido nem muito alcalino.

Porém com grandes poderes vem grandes responsabilidades sendo assim é necessário que os monitores depois de aperceberem que a bateria do dispositivo esgotasse têm, que se, deslocar ao local desejado para fazer substituição da bateria, e de três em três meses devem dirigir-se ao local novamente para fazer a manutenção dos sensores porque os sensores do pH e da Condutividade Elétrica quando ficam muito tempo na água eles podem avariar.

Para diminuir o número de idas ao rio, o código que foi implementado faz, com que, o consumo do dispositivo seja pouco e a bateria tem uma duração de três a quatro meses, e desse modo fazer a manutenção no mesmo dia que vão substituir a bateria.

Foi observado também que as vezes o rio não ajuda muito no verão, porque é uma

época de seca, ou seja o rio sem água, e os dados não podem ser recolhidos, essa é uma das partes negativas mais relevante da plataforma, porque quando o rio seca não é apercebido isso, já que não foi colocado nenhum sensor que avisa-se que o rio secou, desse modo este fica como um trabalho futuro.



# Capítulo 6

## Conclusões e Trabalhos Futuros

Neste capítulo abordou-se duas seções sendo a primeira conclusão e a segunda os trabalhos futuros, na qual a primeira foi possível concluir com este trabalho a relação e a limitação encontrada no decorrer do do desenvolvimento da plataforma, e na seção seguinte abordou-se sobre as melhorias a serem implementadas para os trabalhos futuros.

### 6.1 Conclusão

A plataforma Smart River pode ser uma das melhores soluções para analisar e informar a qualidade da água do rio utilizando a tecnologia IoT para aquisição de dados. O sistema proposto funciona através do servidor TTN que armazena os dados, e faz uso da tecnologia e protocolos LoRaWAN para envio e recepção de dados pela internet.

Foram propostos dois sistemas independentes: o sistema educacional Unity que foi destinado a exibir as médias de dados de forma animada e com interação touch-screen para os visitantes do CCVB, enquanto o servidor Node.js foi criado para controle e gestão dos dispositivos instalados no rio e para futuras pesquisas científicas. Todos os dados estão sendo adquiridos e avaliados, em ambos os sistemas, que se mostraram coerentes e precisos com a saída de dados dos sensores.

Mesmo observado alguns artigos relacionados a este trabalho pode-se garantir que este é um trabalho único e já realizado em Portugal. Por outro lado, tem algumas partes

negativas onde o rio seca e não há dados coerentes, e também em fazer a manutenção dos sensores de pH e de Condutividade Elétrica de três em três meses, porém estas podem ser resolvidas facilmente, e as partes positivas melhoradas.

Com este trabalho espera-se analisar melhor a qualidade das águas do Rio Fervença, fornecer dados fiáveis, para o desenvolvimento da investigação na área, promover a sensibilização ambiental dos residentes e visitantes do Centro Ciência Viva de Bragança, e proporcionar melhoria da qualidade dos recursos hídricos e, conseqüentemente, do bem-estar do meio ambiente e da população.

## 6.2 Trabalhos Futuros

Para trabalhos futuros, pretende-se otimizar a plataforma, para que a vida útil da bateria possa ser maior, e também alterar os sensores de pH e Condutividade Elétrica com padrões da indústria que tenham uma vida útil mais longa dentro da água.

Pretende-se colocar mais um sensor na água para saber o nível de água, para que possa saber se o rio secou. Implementar também *machine learning* para tratar de dados e fazer previsão de dados. Outras análises de dados e alertas de ambientes também podem ser implementados no sistema Smart River. Atualizar aplicação de visualização para ser mais interativa, de modo a dizer o que acontece com água quando esta no valor do pH estabelecido.

# Bibliografia

- [1] J. Rymarczyk, “Technologies, Opportunities and Challenges of the Industrial Revolution 4.0”, em *Theoretical Considerations*, Entrepreneurial Business e Economics Review, <https://doi.org/10.15678/EBER.2020.080110>, 2020, pp. 185–198.
- [2] M. Y. H. Hasan, “Innovative developments”, em *HCI and future trends. Int. J. Autom. Comput. 14th*, IADIS Press, <https://doi.org/10.1007/s11633-016-1039-6>, 2017, pp. 10–20.
- [3] V. d. C. R. Rodrigues, “Rio Fervença: efeitos da perturbação no ecossistema”, tese de doutoramento, 2013.
- [4] G. USGS, *Water Quality Nations Streams and Rivers Current Conditions*, <https://www.usgs.gov/mission-areas/water-resources/science/water-quality-nation-s-streams-and-rivers-current-conditions/>, Abril de 2022.
- [5] Y. L. Heejin Jeong, “Computational Modeling of Touchscreen Drag Gestures Using a Cognitive Architecture and Motion Tracking”, em *Journal of Human–Computer Interaction*, International Journal of Human–Computer Interaction, DOI: 10.1080/10447318.2018.146682019, pp. 510–520.
- [6] S. Santos, *Introdução a internet das coisas*. SS Trader Editor, 2018.
- [7] S. Madakam, V. Lake, V. Lake, V. Lake et al., “Internet of Things (IoT): A literature review”, *Journal of Computer and Communications*, vol. 3, n.º 05, p. 164, 2015.

- [8] S. Al-Sarawi, M. Anbar, K. Alieyan e M. Alzubaidi, “Internet of Things (IoT) communication protocols”, em *2017 8th International conference on information technology (ICIT)*, IEEE, 2017, pp. 685–690.
- [9] Oracle, *O que é Internet das Coisas (IoT)*, <https://www.oracle.com/pt/internet-of-things/what-is-iot/>, Abril de 2022.
- [10] J. F. Colombo e J. de Lucca Filho, “INTERNET DAS COISAS (IoT) E INDÚSTRIA 4.0: revolucionando o mundo dos negócios”, *Revista Interface Tecnológica*, vol. 15, n.º 2, pp. 72–85, 2018.
- [11] UMVOV, *Internet of Things*, <https://www.umov.me/o-que-e-internet-das-coisas-iot/>, Abril de 2022.
- [12] Infowester, *IoT*, <https://www.infowester.com/iot.php>, Maio de 2022.
- [13] Totvs, *Aplicacoes da Internet das Coisas*, <https://www.totvs.com/blog/inovacoes/aplicacoes-da-internet-das-coisas/>, Maio de 2022.
- [14] T. T. Network, *Rede das Coisas Description*, <https://www.thethingsnetwork.org/docs/>, Abril de 2022.
- [15] Mqtt.org, *Protocolo MQTT*, <https://mqtt.org/>, Abril de 2022.
- [16] N. Fernandes, *Protocolo MQTT*, <https://www.hitecnologia.com.br/blog/o-que-e-protocolo-mqtt/>, Abril de 2022.
- [17] M. Bor, J. E. Vidler e U. Roedig, “LoRa for the Internet of Things”, 2016.
- [18] D. M. Ibrahim, “Internet of Things technology based on LoRaWAN revolution”, em *2019 10th International Conference on Information and Communication Systems (ICICS)*, IEEE, 2019, pp. 234–237.
- [19] J. Navarro-Ortiz, S. Sendra, P. Ameigeiras e J. M. Lopez-Soler, “Integration of LoRaWAN and 4G/5G for the Industrial Internet of Things”, *IEEE Communications Magazine*, vol. 56, n.º 2, pp. 60–67, 2018.
- [20] LoRaWAN, *Protocolo LoRaWAN*, <https://www.thethingsnetwork.org/docs/lorawan/>, Abril de 2022.

- [21] V. P. da Silva Junior, *Tecnologia LoRa*, <https://www.embarcados.com.br/conheca-tecnologia-lora-e-o-protocolo-lorawan/>, Abril de 2022.
- [22] R. O. da Silva, W. M. Araujo e M. M. Cavalcante, “Visão Geral Sobre Microcontroladores e Prototipagem com Arduino”, *Tecnologias Em Projeção*, vol. 10, n.º 1, pp. 36–46, 2019.
- [23] Vandertronic, *Microcontroladores*, <http://www.vandertronic.com/index.php/microcontroladores/?print=print>, Abril de 2022.
- [24] R. M. Pillat, “Coordenação dinâmica de visualizações de dados multidimensionais”, 2006.
- [25] E. R. d. A. Valiati, “Avaliação de usabilidade de técnicas de visualização de informações multidimensionais”, 2008.
- [26] K. Cabanga, “Imagens de Minha Autoria”, Abril de 2022.
- [27] K. S. Adu-Manu, F. A. Katsriku, J.-D. Abdulai e F. Engmann, “Smart river monitoring using wireless sensor networks”, *Wireless Communications and Mobile Computing*, vol. 2020, 2020.
- [28] D. Ahmad e A. Kumar, “IoT based smart river monitoring system”, *International Journal of Advance Research, Ideas and Innovations in Technology*, vol. 4, n.º 2, pp. 60–64, 2018.
- [29] E. A. Kadir, A. Siswanto, S. L. Rosa, A. Syukur, H. Irie e M. Othman, “Smart sensor node of WSNs for river water pollution monitoring system”, em *2019 International Conference on Advanced Communication Technologies and Networking (CommNet)*, IEEE, 2019, pp. 1–5.
- [30] O. Elijah, T. A. Rahman, C. Leow, H. Yeen, M. Sarijari, A. Aris, J. Salleh e T. Chua, “A concept paper on smart river monitoring system for sustainability in river”, *International Journal of Integrated Engineering*, vol. 10, n.º 7, 2018.

- [31] R. A. Rauber, *Introducao Ao Microcontrolador Atmega328p*, <https://visiorob.com.br/index.php/2019/09/15/introducao-ao-microcontrolador-atmega328p/>, jul. de 2022.
- [32] Botnroll, *RFM95 Modulo LoRa 868mhz*, <https://www.botnroll.com/pt/rf-lora/2600-rfm95-m-dulo-lora-868mhz.html>, Maio de 2022.
- [33] L. Alliance, *Lora Products*, [https://lora-alliance.org/lora\\_products/rak7243-lpwan-developer-gateway/](https://lora-alliance.org/lora_products/rak7243-lpwan-developer-gateway/), jun. de 2022.
- [34] J. R. V. Fogaça, *Conceito pH*, <https://www.manualdaquimica.com/fisico-quimica/conceito-ph.htm>, jul. de 2022.
- [35] Botnroll, *pH*, <https://www.botnroll.com/pt/ph/3088-medidor-de-ph-analogico-v2.html>, jun. de 2022.
- [36] MAUSER, *Sensor de Condutividade*, [https://mauser.pt/catalog/product\\_info.php?products\\_id=096-8593](https://mauser.pt/catalog/product_info.php?products_id=096-8593), jul. de 2022.
- [37] DFRobot, *Electrical conductivity Sensor*, <https://www.dfrobot.com/product-1123.html>, jun. de 2022.
- [38] E. Oliveira, *Como Usar Com Arduino Sensor de Temperatura DS18B20 Prova de Água do Tipo Sonda*, <https://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-sensor-de-temperatura-ds18b20-prova-dagua-do-tipo-sonda>, jul. de 2022.
- [39] DFRobot, *Sensor de temperatura da água*, <https://www.dfrobot.com/product-1354.html>, jun. de 2022.
- [40] Electrofun, *Sensor Temperatura Humidade DHT11*, <https://www.electrofun.pt/sensores-arduino/sensor-temperatura-humidade-dht11>, jun. de 2022.
- [41] OBO, *Material elétrico caixas de derivação*, <https://loja.watt.pt/material-electrico-caixas-de-derivacao-caixa-derivacao-estanque-t160-ip66>, jun. de 2022.

- [42] Gewiss, *Material elétrico caixas de derivação*, <https://www.gewiss.com/ww/pt/products/experience-catalogue/catalogs/series/building/44-ce-range-technopolymer-surface-mounting-watertight-junction-boxes>, jun. de 2022.
- [43] Balena, *BalenaEtcher Logótipo*, <https://www.balena.io/etcher/>, jun. de 2022.
- [44] Thethingsnetwork, *Thethingsnetwork Logótipo*, <https://www.thethingsnetwork.org/>, jun. de 2022.
- [45] Arduino, *Arduino Logótipo*, <https://www.arduino.cc/>, jun. de 2022.
- [46] Unity, *Unity Logótipo*, <https://unity.com/>, jun. de 2022.
- [47] D. Flanagan, *JavaScript: o guia definitivo*. Bookman Editora, 2004.
- [48] DroidScript, *JavaScript Logótipo*, <https://geekflare.com/handling-files-in-javascript/>, jun. de 2022.
- [49] L. Picollo, *Vue JS: o que é, como funciona e vantagens*, <https://blog.getbootstrap.com/>, abril de 2022.
- [50] —, *Vue JS: o que é, como funciona e vantagens*, <https://blog.getbootstrap.com/>, abril de 2022.
- [51] S. Olusola, *How to use EJS to template your Node.js application*, <https://blog.logrocket.com/how-to-use-ejs-template-node-js-application/>, jul. de 2022.
- [52] icon-icons, *arquivo-tipo-ejs*, <https://icon-icons.com/pt/icone/arquivo-tipo-ejs/130626>, jul. de 2022.
- [53] C. Innocent, *Understanding Axios POST requests*, <https://blog.logrocket.com/understanding-axios-post-requests/>, jul. de 2022.
- [54] —, *Understanding Axios POST requests*, <https://blog.logrocket.com/understanding-axios-post-requests/>, jul. de 2022.
- [55] M. Pigula, *Socket.io tutorial: Real-time communication in web development*, <https://tsh.io/blog/socket-io-tutorial-real-time-communication/>, jul. de 2022.

- [56] Socket.IO, *Socket.IO*, <https://socket.io/>, jul. de 2022.
- [57] Nodejs, *Node.js*, <https://nodejs.dev/learn>, jul. de 2022.
- [58] P. Pinto, *Node.js: Aprenda a instalar e a desenvolver a primeira app*, <https://pplware.sapo.pt/tutoriais/node-js-aprenda-a-instalar/>, jul. de 2022.
- [59] Prisma.io, *Documentação Prisma*, <https://www.prisma.io/docs/>, jul. de 2022.
- [60] Hivemq, *Introducing the MQTT Protocol - MQTT Essentials*, <https://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt/>, jul. de 2022.
- [61] —, *Introducing the MQTT Protocol - MQTT Essentials*, <https://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt/>, jul. de 2022.
- [62] NOAH, “Smart River”, Abril de 2022.