

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO  
INSTITUTO DE COMPUTAÇÃO  
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

LUIZ HUMBERTO REIS FORNARI

DESENVOLVIMENTO DE UM SISTEMA PARA A COLETA DE RESÍDUOS DE  
EQUIPAMENTOS ELETRO-ELETRÔNICOS

RIO DE JANEIRO  
2022

LUIZ HUMBERTO REIS FORNARI

DESENVOLVIMENTO DE UM SISTEMA PARA A COLETA DE RESÍDUOS DE  
EQUIPAMENTOS ELETRO-ELETRÔNICOS

Trabalho de conclusão de curso de graduação  
apresentado ao Instituto de Computação da  
Universidade Federal do Rio de Janeiro como  
parte dos requisitos para obtenção do grau de  
Bacharel em Ciência da Computação.

Orientador: Profa. Valeria Bastos  
Co-orientador: Prof. Claudio Miceli

RIO DE JANEIRO

2022

## CIP - Catalogação na Publicação

F727d      Fornari, Luiz Humberto Reis  
Desenvolvimento de um sistema para a coleta de  
resíduos de equipamentos eletro-eletrônicos / Luiz  
Humberto Reis Fornari. -- Rio de Janeiro, 2022.  
54 f.

Orientadora: Valeria Menezes Bastos.  
Trabalho de conclusão de curso (graduação) -  
Universidade Federal do Rio de Janeiro, Instituto  
de Computação, Bacharel em Ciência da Computação,  
2022.

1. Reciclagem. 2. Resíduo eletrônico. 3. IoT. 4.  
Rede LoRa. 5. Sensores. I. Bastos, Valeria Menezes,  
orient. II. Título.

LUIZ HUMBERTO REIS FORNARI

DESENVOLVIMENTO DE UM SISTEMA PARA A COLETA DE RESÍDUOS DE  
EQUIPAMENTOS ELETRO-ELETRÔNICOS

Trabalho de conclusão de curso de graduação  
apresentado ao Instituto de Computação da  
Universidade Federal do Rio de Janeiro como  
parte dos requisitos para obtenção do grau de  
Bacharel em Ciência da Computação.

Aprovado em 29 de março de 2022

BANCA EXAMINADORA:

Documento assinado digitalmente  
**gov.br** VALERIA MENEZES BASTOS  
Data: 03/04/2024 07:09:41-0300  
Verifique em <https://validar.iti.gov.br>

---

Valéria Bastos  
Universidade Federal do Rio de Janeiro

  
Claudio Miceli  
Universidade Federal do Rio de Janeiro

---

Participação por videoconferencia

Silvana Rossetto  
Universidade Federal do Rio de Janeiro

---

Participação por videoconferencia

Claudio Fernandes  
Universidade Federal Fluminense

A conclusão deste importante ciclo, representada por este trabalho, só foi possível com o apoio incondicional de minha família. Deles recebi o suporte e estímulo determinantes a esta conquista. Grato, honro mãe, pai e irmãos e dedico a eles este trabalho.

## AGRADECIMENTOS

Agradeço a Reciclotron, empresa que inspirou esta pesquisa, por proporcionar a oportunidade de realizar um trabalho relevante para a sociedade e contribuir diretamente com meu crescimento profissional e pessoal; aos orientadores Valeria e Claudio, e aos professores Aguiar, Claudson e Juliana, por, com muito carinho e atenção, terem me orientado durante minha trajetória acadêmica, por suas disciplinas e por oferecerem apoio e suporte muito além do acadêmico; ao GRIS, grupo acadêmico de extensão, que teve papel determinante na permanência e conclusão do curso, na minha trajetória profissional, na formação de amizades e crescimento pessoal.

Agradeço também aos amigos Vanessa e Leon, que me deram suporte e contribuíram diretamente na minha formação profissional; Pipe, Thamy e Ana, que dedicaram apoio e carinho; ao JP, que embarca comigo nas aventuras profissionais e estimula a aprender mais. Obrigado pela amizade, por caminharem junto e por contribuírem nessa jornada.

Por último e em especial, agradeço novamente ao Claudio por ter sido um PROFESSOR. Sua orientação e atenção incondicional ao aluno, sua dedicação e interesse são, além de inspirador, decisivos na minha formação.

## RESUMO

Os baixos níveis de reciclagem de resíduos eletrônicos praticados no país causam um passivo ambiental extremamente danoso. Neste contexto, este trabalho propõe o desenvolvimento de uma sistema para permitir a adoção de coletores de resíduos eletrônicos com características IoT que estimulem a prática da reciclagem e, por consequência, reduza o impacto ambiental. Para atingir este objetivo são relacionados diversos tipos de sensores que identificam o volume armazenado no coletor. As informações sensoreadas são transmitidas através de uma rede LoRa para um dispositivo de borda que, por sua vez, retransmite via WiFi e HTTP Rest ao serviço remoto. Desta forma é possível obter relatórios que auxiliem na identificação do estado atual da reciclagem e no planejamento de ações a serem tomadas.

**Palavras-chave:** reciclagem; resíduo eletrônico; iot; rede LoRa; sensores;

## ABSTRACT

The low levels of electronic waste recycling practiced in the country cause an extremely harmful environmental liability. In this context, this work proposes the development of a system to allow the adoption of electronic waste collectors with IoT characteristics that stimulate the practice of recycling and, consequently, reduce the environmental impact. To achieve this goal, several types of sensors are listed that identify the volume stored in the collector. The sensed information is transmitted over a LoRa network to an edge device which, in turn, relays via WiFi and HTTP Rest to the remote service. In this way, it is possible to obtain reports that help identify the current status of recycling and plan actions to be taken.

**Keywords:** recycling; e-waste; iot; LoRa network; sensors;

## LISTA DE ILUSTRAÇÕES

Figura 1 – Topologia física em malha . . . . .	18
Figura 2 – Topologia física em estrela . . . . .	19
Figura 3 – Topologia física em árvore . . . . .	19
Figura 4 – Ponte de Wheatstone . . . . .	23
Figura 5 – Croqui do sensor óptico . . . . .	24
Figura 6 – Croqui do coletor com os sensores . . . . .	24
Figura 7 – Topologia proposta neste trabalho . . . . .	27
Figura 8 – Comparativo entre as pilhas OSI, TCP/IP e LoRa . . . . .	28
Figura 9 – Representação da implantação na cidade de Nova Friburgo . . . . .	29
Figura 10 – Diagrama esquemático do sensor ultrassônico . . . . .	33
Figura 11 – Fluxograma de funcionamento do nó coletor . . . . .	33
Figura 12 – Diagrama esquemático do sensor de abertura e fechamento da porta do coletor . . . . .	34
Figura 13 – Diagrama esquemático do nó coletor . . . . .	35
Figura 14 – Fluxograma de funcionamento do <i>gateway</i> . . . . .	35
Figura 15 – Fluxograma de funcionamento do serviço remoto . . . . .	36
Figura 16 – Configuração do experimento . . . . .	36
Figura 17 – Imagem da montagem na <i>proto-board</i> do conjunto do nó coletor com- posto pelo microcontrolador e sensores. . . . .	39
Figura 18 – Imagem da montagem do microcontrolador como <i>gateway</i> . . . . .	40
Figura 19 – Imagem das informações exibidas no terminal pelo serviço remoto. . . . .	40

## LISTA DE CÓDIGOS

Código 1	Código do coletor para sensorar e enviar a informação ao gateway.	45
Código 2	Código do gateway para receber as mensagens dos coletores e re- passar ao serviço remoto. . . . .	48
Código 3	Serviço HTTP Rest para receber as mensagens e exibir o conteúdo no terminal. . . . .	53

## LISTA DE TABELAS

Tabela 1 – Tecnologias de infraestrutura para comunicação de dados . . . . .	17
Tabela 2 – Consumo de corrente por modo de operação . . . . .	37
Tabela 3 – Tempos de duração da bateria com base numa corrente de 17,6mAh . .	37
Tabela 4 – Tempos de duração da bateria com base numa corrente de 14,3mAh . .	38

## LISTA DE ABREVIATURAS E SIGLAS

REEE	Resíduos de Equipamentos Eletro-Eletrônicos
LDR	Light Dependent Resistor
GSM	Global System for Mobile
API	Application Programming Interface
HTTP	Hypertext Transfer Protocol
CoAP	Constrained Application Protocol
MQTT	Message Queuing Telemetry Transport
gRPC	Google Remote Procedure Call
GPIO	General Pin Input Output

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>12</b>
1.1	MOTIVAÇÃO . . . . .	12
1.2	CONTEXTUALIZAÇÃO . . . . .	12
1.3	OBJETIVO . . . . .	13
1.4	ORGANIZAÇÃO DO TEXTO . . . . .	15
<b>2</b>	<b>CONCEITOS BÁSICOS . . . . .</b>	<b>16</b>
2.1	INTERNET OF THINGS - IOT . . . . .	16
2.2	INFRAESTRUTURA DE REDE . . . . .	17
2.3	TOPOLOGIAS FÍSICAS . . . . .	18
<b>3</b>	<b>TRABALHOS RELACIONADOS . . . . .</b>	<b>20</b>
<b>4</b>	<b>PROPOSTA . . . . .</b>	<b>22</b>
4.1	SENSOREAMENTO . . . . .	22
4.2	COMUNICAÇÃO DE DADOS . . . . .	25
4.3	TOPOLOGIA . . . . .	27
4.4	SERVIÇOS REMOTOS . . . . .	29
<b>5</b>	<b>EXPERIMENTO . . . . .</b>	<b>32</b>
5.1	MONTAGEM DO EXPERIMENTO . . . . .	32
5.2	EXECUÇÃO DO EXPERIMENTO . . . . .	35
<b>6</b>	<b>CONCLUSÃO . . . . .</b>	<b>41</b>
<b>7</b>	<b>TRABALHOS FUTUROS . . . . .</b>	<b>42</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>43</b>
	<b>APÊNDICE A – CÓDIGOS DOS MICROCONTROLADORES. . . . .</b>	<b>45</b>
	<b>APÊNDICE B – CÓDIGO DO SERVIÇO REMOTO. . . . .</b>	<b>53</b>

# 1 INTRODUÇÃO

## 1.1 MOTIVAÇÃO

A evolução social e tecnológica agrega, além dos benefícios objetivados, o ônus da grande geração de resíduos de equipamentos eletro-eletrônicos (REEE). De acordo com a ONU, são gerados por ano, no mundo inteiro, 50 milhões de toneladas de resíduos eletrônicos e projeções indicam que no ano de 2030 o volume ultrapassará 75 milhões de toneladas. Estes resíduos, quando não são tratados de forma correta, agredem o meio ambiente com a contaminação por metais pesados e outras substâncias tóxicas, como por exemplo chumbo, mercúrio, bário e cádmio.

Não fosse suficiente o aspecto poluente destes resíduos, o custo relacionado a eles também é relevante. Metais preciosos ou raros como ouro, prata, cobre e paládio estão presentes nestes dispositivos e a obtenção destes materiais na natureza é agressiva e cada vez mais custosa. Neste contexto, o reuso e a reciclagem dos resíduos eletrônicos se apresentam como uma alternativa atraente por serem menos custosos tanto em aspectos financeiros quanto em aspectos ambientais.

No Brasil, apesar da Política Nacional de Resíduos Sólidos, instituída pela lei Nº 12.305 de 2 de agosto de 2010, e do decreto Nº 10.240 de 12 de fevereiro de 2020, que regulamenta o sistema de logística reversa de REEE, apenas 2% de todo REEE gerado é reciclado. Esta performance baixa é atribuída a fatores como custos de logística, manufatura reversa, e aspectos culturais.

## 1.2 CONTEXTUALIZAÇÃO

Conforme apresentado, o tratamento dos resíduos, independente de sua natureza, é de grande relevância para todo o planeta. Olhando especificamente para os resíduos de equipamentos eletro-eletrônicos (REEE) percebemos uma vacância de soluções que elevem o percentual tratado deste tipo de resíduo a níveis saudáveis tanto para o meio ambiente quanto para o setor industrial. A logística reversa destes resíduos, isto é, o recolhimento após seu uso e a destinação final para reciclagem, é um processo caro no aspecto financeiro porque envolve parques industriais para armazenar e tratar estes resíduos e envolve também o transporte deles. Além disso, a falta de locais adequados para o descarte é um fator limitante.

Desta necessidade nasceu a Reciclotron, uma empresa cuja função social é interligar os recicladores de REEE com os produtores deste tipo de resíduo, fomentando assim o tratamento responsável e sustentável, por meio de bonificação e campanhas educacionais. Como consequência desta atuação os índices de captação e tratamento destes recursos

aumentarão e a agressão ambiental diminuirá na medida em que reduz a poluição e a extração de metais na natureza.

A Reciclotron <sup>1</sup> iniciou sua operação em 2019 na cidade de Nova Friburgo, estado do Rio de Janeiro, como um projeto estratégico de cidades inteligentes apoiado pela Secretaria de Ciência e Tecnologia de Nova Friburgo. Foram distribuídos 21 coletores de 50 e 200 litros em alguns locais estratégicos da cidade, como prédios da administração pública, rodoviária e comércios, para receber os resíduos descartados. Os usuários podem descartar nos coletores todo tipo de REEE, desde celulares, mouses ou cartões de memória até gabinetes de computador, monitores e televisores. As únicas limitações são que os resíduos descartados sejam REEE e que o coletor esteja apto a armazenar determinado resíduo. Os coletores de 50 litros, por exemplo, recebem apenas itens pequenos como celulares e tablets.

Cada coletor atinge sua capacidade de armazenamento em momentos distintos e muitas vezes o mesmo coletor varia a velocidade com que atinge sua capacidade, o que torna mais difícil a definição das rotas de visita. Como os coletores são meras caixas de madeira ou polímero, um funcionário precisa rotineiramente verificar o volume armazenado e transportar o conteúdo para uma central de armazenamento, de onde as recicladoras por sua vez retiram os resíduos. Em alguns casos, onde o coletor está em um local constantemente vigiado, o funcionário responsável pela coleta é avisado da necessidade de esvaziamento e então visita o coletor. Visto que essa rotina é custosa sob o aspecto financeiro e demanda tempo e esforço que poderiam ser melhor empenhados, dotar os coletores com capacidade de sensorar o que foi depositado e comunicar essa informação aos sistemas da Reciclotron é uma solução para o caso em tela.

De igual importância é estimular que os REEE sejam depositados nos coletores. Para isso, campanhas educativas são veiculadas nas mídias sociais e benefícios são oferecidos aos usuários. Cada vez que um resíduo é depositado, o usuário cadastra este descarte no sistema próprio da empresa, que calcula e credita um valor correspondente em pontos. Estes pontos podem ser trocados por produtos ou serviços na rede de parceiros da Reciclotron. Atualmente o usuário que deposita o REEE cadastra no sistema esse descarte e um funcionário verifica posteriormente a veracidade da informação.

### 1.3 OBJETIVO

Tomando como base o cenário exposto, este trabalho visa apresentar uma proposta que contribua para o aumento da taxa de reciclagem de REEE através do desenvolvimento de um sistema para a coleta destes resíduos. Espera-se com isso que o custo de logística seja reduzido de forma atrativa para a indústria e comércio. Além disso, espera-se também

---

<sup>1</sup> Website Reciclotron: [www.reciclotron.com.br](http://www.reciclotron.com.br)

obter maior engajamento da sociedade no descarte de resíduos uma vez que a oferta de pontos de coleta deixa de ser um problema.

Os custos de logística citados se devem a alguns fatores. Um deles é a necessidade atual de vistoriar visualmente o volume armazenado em cada coletor. Mesmo em locais onde há mão de obra acessível, esta tarefa é custosa e suscetível a erros podendo até mesmo não ser realizada. Também é um custo de logística a visita para esvaziamento de um ponto de coleta. Embora a visita seja uma tarefa necessária, o problema aqui é um ponto de coleta ser visitado sem que ele tenha atingido um volume armazenado de resíduos que justifique esta ação. Outro fator que pode influenciar de forma a diminuir os custos de logística é calcular uma estimativa de volume com base em informações pregressas dos coletores. Assim, é possível refinar para cada ponto de coleta seus níveis de alerta para esvaziamento e com base nisso otimizar as rotas reduzindo ainda mais os custos.

O sistema a ser proposto possibilita que as caixas coletoras contem com a capacidade de identificar seu volume armazenado de resíduos e comunicar esta informação a um serviço remoto. Embora fuja ao escopo deste trabalho, o desenvolvimento de uma plataforma completa para a tomada de decisão à partir das informações recebidas, a comunicação de dados proposta tornará possível que os fatores apontados sejam mitigados e a solução seja escalável.

A adoção de tecnologia IoT para esta proposta é muito factível porque permite que estes pontos de coleta tenham a capacidade de comunicar a informação sem a necessidade de interação humana, que é muito suscetível a falhas. Desta forma, confere integridade e confiabilidade tanto aos processos quanto aos dados produzidos, visto que o próprio coletor detecta e transmite as informações sem que o fator humano faça parte deste processo.

Por outro lado, a adoção de dispositivos IoT significa diretamente aumentar a quantidade de dispositivos eletrônicos em utilização, o que naturalmente resultará em maior quantidade de resíduos gerados. Entretanto, os dispositivos sugeridos neste trabalho não são bens descartáveis, pelo contrário, se trata de bens duráveis com uma vida útil de anos ou décadas e em quantidade pequena. Especialmente quando comparado ao volume de REEE descartados de forma incorreta, a quantidade de dispositivos utilizados, aliado ao benefício ambiental e econômico, justifica sua adoção.

Desta forma, a solução proposta organiza os coletores em células que concentram e repassam as informações ao serviço remoto. Para isso, cada coletor utiliza um microcontrolador para administrar o sensoramento dos resíduos e enviar a informação através da rede LoRa para o nó intermediário. Este nó intermediário, por sua vez, recebe a informação via rede LoRa e a repassa ao serviço remoto através da rede WiFi. O serviço recebe, então, a informação e tem a capacidade de aplicar os devidos tratamentos.

Os resultados observados foram satisfatórios. Isto porque a informação sensoreada foi trafegada desde o coletor até o serviço remoto, conforme esperado. Com isso, o sistema proposto se mostrou eficiente para o objetivo. Outro resultado favorável observado foi

o baixo consumo energético na operação dos coletores. Testes mais abrangentes não puderam ser feitos pelos motivos expostos, porém, esta característica de baixo consumo energético é um indício de que a solução proposta possa ser eficiente e aderente ao projeto em situações de uso real.

#### 1.4 ORGANIZAÇÃO DO TEXTO

O restante do texto está estruturado da seguinte forma:

- No Capítulo 2, são apresentados os conceitos básicos necessários para a compreensão deste trabalho.
- No Capítulo 3, é feita uma breve revisão dos trabalhos relacionados a este tema.
- No Capítulo 4, é apresentada a proposta deste trabalho e discutidos os temas sensoreamento, comunicação de dados, topologia e serviços remotos. Estes temas são necessários para a definição da arquitetura.
- No Capítulo 5, é proposto um experimento para validar a arquitetura proposta. Também é apresentada a montagem do experimento e discutida sua execução.
- No Capítulo 6, são apresentadas as conclusões deste trabalho.
- No Capítulo 7 são propostos trabalhos futuros que podem ser desenvolvidos a partir deste.

## 2 CONCEITOS BÁSICOS

### 2.1 INTERNET OF THINGS - IOT

Conforme descrito por (HUANG et al., 2014), IoT consiste em uma variedade de objetos que estão conectados a Internet por meio sem fio ou não. Os objetos em IoT podem sensorar o ambiente, transferir dados e comunicar entre si. Se tornam poderosas ferramentas para entender o mundo físico e responder prontamente a eventos e irregularidades. Já para (Xu; He; Li, 2014), IoT é definido como uma infraestrutura de rede dinâmica e global, com capacidade de se autoconfigurar, baseada em comunicação por protocolos padronizados e interoperáveis onde "coisas" físicas e virtuais tem identidades, atributos físicos, personalidades virtuais e integrados a Internet. Por sua vez, (Zhu et al., 2015) resume IoT à ideia básica de que tudo ao nosso redor pode ser conectado, sensorado e de forma cooperativa comunicar através da internet. Destas definições pode-se observar um consenso de que o conceito de IoT relaciona objetos à capacidade de sensorar o ambiente em que está imerso e comunicar esta informação através da internet.

Estes mesmos trabalhos indicam a arquitetura IoT organizada em camadas. Para (Xu; He; Li, 2014) a arquitetura consiste de quatro camadas, sejam elas sensoramento, rede, serviço e interface. A camada de sensoramento é responsável por perceber ou controlar o meio físico e gerar dados. A camada de rede é responsável por interconectar e transferir dados. A camada de serviço é responsável por disponibilizar e gerenciar os serviços que são oferecidos aos usuários. Por último a camada de interface é responsável por fornecer os métodos de interação com usuários ou outras aplicações. Já em (Zhu et al., 2015) são apontadas seis camadas na arquitetura IoT: identificação, sensoramento, comunicação, computação, serviços e semântica. A camada de identificação, como o nome diz, identifica os dispositivos e serviços. A camada de sensoramento coleta informações dos objetos e as armazena. A comunicação estabelece interligação entre os objetos e a internet e transfere os dados por protocolos bem definidos. A camada de computação trata do *hardware* responsável por processar as informações no dispositivo. A camada de serviço fornece os serviços necessários a identificação, agregação, tomada de decisão e reação. Por último, a camada de semântica é responsável por extrair valor das informações utilizando para isso técnicas de inteligência de dados. Novamente é possível observar semelhança nestas definições. Desta forma, os trabalhos concordam com as camadas de sensoramento, rede, comunicação e serviço.

## 2.2 INFRAESTRUTURA DE REDE

A infraestrutura de rede atualmente está numa posição bastante consolidada. O trabalho apresentado por (POPLI; JHA; JAIN, 2019) traz uma análise das diferentes tecnologias de infraestrutura disponíveis que atendem aos requisitos de um ambiente IoT. Dentre as abordadas por ele foram selecionadas para esta discussão as tecnologias ZigBee, Wi-Fi, BLE, LoRa e NB-IoT por serem bastante difundidas e diferentes entre si. A tabela 1 apresenta de forma bastante resumida um comparativo entre estas tecnologias.

Tabela 1 – Tecnologias de infraestrutura para comunicação de dados

	ZigBee	Wi-Fi	BLE	LoRa	NB-IoT
Alcance	75~100m	70~250m	100m	2~5km	10~15km
Taxa de dados	250kbps	600Mbps	1Mbps	50kbps	200kbps
Throughput da aplicação	250kbps	600Mbps	0,27Mbps	50kbps	150kbps
Latência	15ms	20ms	10ms	10ms	10s
Duração da bateria	alguns anos	-	-	> 10 anos	> 10 anos
Consumo de corrente (sleep)	12 $\mu$ A	30 $\mu$ A	9 $\mu$ A	1 $\mu$ A	5 $\mu$ A
Consumo de corrente (transmitindo)	52 mA	251 mA	32 mA	32 mA	120 mA
Custos	baixo	moderado	muito baixo	baixo	baixo

Cada uma delas conta com características próprias que, num dado ambiente IoT, submetidas a requisitos específicos, se mostram mais adequadas quando comparadas às outras tecnologias. Comparando, por exemplo, LoRa e NB-IoT, para projetos onde a latência não seja fator determinante, a tabela 1 sugere que talvez a tecnologia NB-IoT possa ser mais adequada. Porém não indica que caso seja necessário implantar esta infraestrutura,

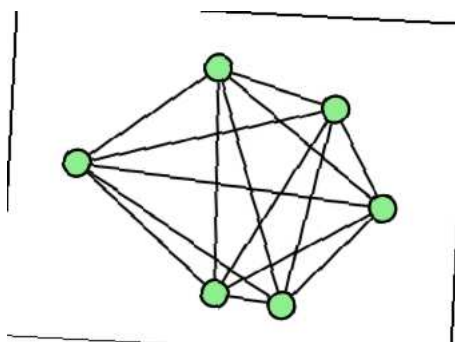
os seus custos podem ser extremamente elevados e inviabilizar o projeto. Isto porque se utiliza a rede LTE e suas licenças, além de custosas, podem ser reguladas pelos órgãos competentes. Com isto, a escolha por uma tecnologia de rede está diretamente relacionada com uma análise das necessidades do projeto.

### 2.3 TOPOLOGIAS FÍSICAS

A topologia indica de que forma os recursos estão dispostos e se comunicam, ou seja, trata tanto da organização física quanto lógica. A topologia física especifica como os dispositivos se comunicam entre si a partir de meios físicos, sejam eles cabeados ou não. Dentre as tecnologias existentes destaca-se aqui as topologias malha, estrela e árvore.

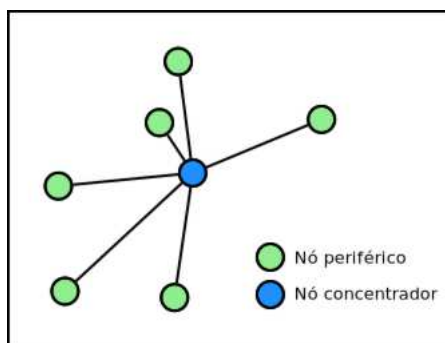
A topologia em malha, mostrada na figura 1, tem por característica que todos os nós participantes desta rede sejam capazes de se comunicar entre si diretamente. A organização em malha permite que haja redundância na comunicação entre os nós ao passo que torna a execução mais custosa. Isto porque aumenta o uso de rede aumentando por consequência o consumo energético.

Figura 1 – Topologia física em malha



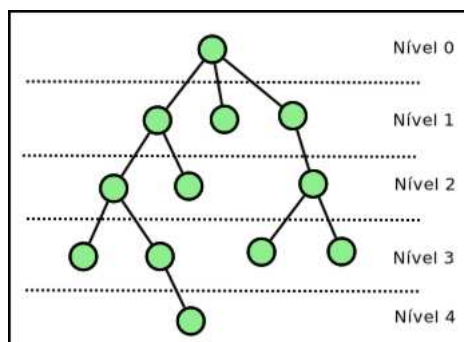
Observando agora a topologia em estrela, sua característica é a existência de um nó concentrador onde todos os outros nós participantes desta rede se conectam. Toda comunicação é feita através deste dispositivo central, ou seja, cada nó pertencente a esta estrela se comunica com qualquer outro nó, interno ou externo, enviando ou recebendo informações apenas por meio do nó central. Duas grandes desvantagens neste modelo são a necessidade deste dispositivo central dispor de um suprimento de energia maior, o que em muitos casos inviabiliza o uso de baterias, e a quantidade de nós que se conectam a ele, que em números elevados acarreta em falhas na comunicação e custos mais altos. Por outro lado, a vantagem deste modelo é justamente a baixa complexidade dos nós periféricos e, por consequência, seu baixo custo. Como eles apenas se comunicam com um único dispositivo e apenas quando precisam enviar ou receber informações, nunca para retransmitir, o dispositivo pode ser mais simples sob o aspecto eletrônico e o suprimento de energia pode ser feito por meio de baterias. A figura 2 mostra esta topologia.

Figura 2 – Topologia física em estrela



Por último, a topologia em árvore vista na figura 3 tem como característica a organização dos dispositivos de forma hierárquica. Assim como ocorre na topologia estrela, os dispositivos periféricos se comunicam apenas com seus dispositivos centrais que, por sua vez, se comunicam apenas com seus periféricos ou seus centrais. De forma prática, o que difere a topologia em árvore da topologia em estrela é a necessidade de tratar a comunicação sob um aspecto hierárquico. Diferente do roteamento, onde decisões lógicas podem ser tomadas de forma hierárquica para obter um caminho até o destino, na topologia física em árvore as informações necessariamente seguem o caminho definido pela árvore. Isto é interessante, por exemplo, nos casos onde a fusão de dados se aplica. Considerando um número qualquer de dispositivos periféricos que sensoreiam e comunicam determinada informação, o dispositivo que está no nível acima da hierarquia recebe estes dados, aplica a fusão e repassa a informação fundida ao nível acima. Por sua vez, o nível acima pode aplicar o mesmo processo ou qualquer outro que for adequado. Desta forma, é fácil perceber que a topologia em árvore pode apresentar as mesmas vantagens e desvantagens observadas na topologia em estrela.

Figura 3 – Topologia física em árvore



### 3 TRABALHOS RELACIONADOS

Alguns trabalhos foram publicados sugerindo uma arquitetura para a coleta e o gerenciamento dos resíduos. Em (Shyam; Manvi; Bharti, 2017) é proposta uma solução para a coleta inteligente do resíduo comum com base no volume de resíduos depositados nas lixeiras. Este volume é detectado por um sensor ultrassônico e a informação é enviada por meio de protocolo IEEE 802.11 para uma base de dados que, por sua vez, alimenta um sistema de tomada de decisão que calcula a rota otimizada através de um algoritmo *shortest path spanning tree* (SPST). O trabalho também propõe a implementação de um algoritmo de predição de níveis de volume ocupado tomando como base para esta tarefa a série histórica de cada lixeira. Porém, a proposta não apresenta soluções para otimizar o consumo energético demandado pelo protocolo IEEE 802.11 assim como o consumo de processamento pelo microcontrolador. A otimização energética é importante porque, além de ser uma postura ambientalmente responsável, mantém o sistema em funcionamento por um período de tempo maior antes de requerer intervenção para substituição de baterias. Este objetivo poderia ser obtido pela adoção de protocolos de comunicação energeticamente mais eficientes e ativando o microcontrolador, assim como o rádio de transmissão, apenas quando necessário.

A proposta apresentada por (Longhi et al., 2012) também considera o volume de resíduos sólidos depositado na lixeira e envia a informação através da rede de comunicação para um sistema central. As diferenças estão nos protocolos adotados. As lixeiras se comunicam por meio do padrão 802.15.4 com nós intermediários que, por sua vez, atuam como *bridges* trafegando a informação pela rede GSM até os servidores. O uso de nós intermediários é uma proposta bastante interessante porque agrupa em LoWPANs o nó intermediário e as lixeiras próximas permitindo assim que o consumo energético seja reduzido. Já a comunicação por GSM entre o nó intermediário e os servidores aproveita a infraestrutura de telecomunicação de longa distância já existente facilitando assim a implantação e reduzindo custos. Importante apontar que em locais onde a infraestrutura de telecomunicações não opere de forma satisfatória outras tecnologias, como por exemplo LoRa, podem ser mais indicadas em comparação ao GSM.

Uma outra solução semelhante as anteriores foi apresentada por (SOUSA; COSTA, ). Nesta proposta foram adicionados, além do sensor ultrassônico, células de carga, chave magnética, sensor de temperatura e sensor de pressão. Estes últimos foram incluídos para tentar diminuir os erros de leitura do sensor ultrassônico, que é suscetível a alterações em função da temperatura e da umidade no ambiente sensoreado. Uma característica muito interessante que não foi percebida nos outros trabalhos é o uso de células de carga. Esta informação pode ser bastante útil para os algoritmos de logística porque os veículos utilizados para coletar os resíduos tem limitação de volume e de peso. Por isso, resíduos

que ocupam pouco volume podem ter peso elevado e esgotar a capacidade do veículo, logo, algoritmos que não cruzam esse tipo de informação estão sujeitos a este tipo de problema. Para a infraestrutura de comunicação usa protocolo IEEE 802.11 e, conforme já apontado, não é um protocolo eficiente sob o aspecto energético. Para o envio das mensagens entre os coletores e o serviço de dados se utiliza o protocolo MQTT. Este protocolo tem sido bastante usado em projetos IoT por ser bem estruturado, de fácil implementação e requer baixa largura de banda, porém, mantém o canal aberto com o servidor que se conecta e isto pode acarretar em uso desnecessário de energia. Por último, foi adotado um sensor magnético para perceber a abertura da tampa e na ocorrência deste evento as informações sensoreadas são enviadas. Infelizmente as mensagens são enviadas de forma periódica independente do evento de abertura da tampa. Logo, o que poderia ser uma otimização para baixar o consumo energético acaba agindo da forma oposta ao sensorar e enviar dados desnecessários.

O trabalho apresentado por (KANG et al., 2020) também sugere sensor ultrassônico e comunicação via IEEE 802.11, porém difere dos anteriores ao focar nos REEE e implementar um sistema de notificação para guiar os usuários aos locais de coleta dos resíduos e notificar as empresas de logística da necessidade de coletar determinado local. Diferente das anteriores, são implementadas técnicas de redução do consumo de energia colocando o microcontrolador em modo *sleep*. O microcontrolador permanece neste estado até que o evento da leitura de um QRcode faz com que o servidor remoto envie o sinal de *wake up* e, após a caixa verificar o nível de resíduo, envia a informação aos servidores remotos e volta ao modo *sleep*.

Todos os trabalhos analisados oferecem propostas interessantes e, de forma geral, todos apresentam alguma solução específica para um tipo de resíduo, ou não otimiza o consumo energético e a comunicação. Por se tratar de uma caixa coletora que se comunica com um serviço remoto, o suprimento de energia é um fator relevante uma vez que são colocadas em locais de acesso comunitário e não se pode garantir acesso ininterrupto a fontes de energia. Além disso, as práticas sustentáveis apontam para a redução do consumo energético sempre que possível. Outro fator relevante é o protocolo de comunicação escolhido. Apesar da facilidade de implementação e escalabilidade do protocolo IEEE 802.11, este é um protocolo suscetível a interferências, tem alcance limitado e apresenta consumo energético elevado. Desta forma, as propostas deixam lacunas a serem melhoradas por trabalhos futuros.

## 4 PROPOSTA

Dentro do contexto apresentado, este trabalho tem por objetivo o desenvolvimento de um sistema IoT voltada para a coleta de Resíduos de Equipamentos Eletro-Eletrônicos (REEE) que proporcione meios para implantar a comunicação de dados dos coletores com um sistema remoto, a otimização das rotas de coleta dos resíduos, a bonificação para estimular o descarte correto pela comunidade e baixo consumo energético. Para que este objetivo seja alcançado é preciso que as caixas coletoras tenham capacidade de sensoriamento para detectar o volume de REEE que foi depositado em cada uma delas. Com a informação de volume ocupado obtida, é necessário então comunicar ao serviço remoto e, portanto, uma infraestrutura de rede será necessária. Nesta infraestrutura as caixas coletoras serão organizadas em células que comunicarão com nós intermediários operando em formato de *sink node*, onde as informações são comutadas da rede coletora para o serviço remoto na Internet.

Foge ao escopo deste trabalho o desenvolvimento de um sistema de bonificação com base no REEE depositado nos coletores, assim como sua identificação automatizada. Entretanto, independente do sistema que venha a ser desenvolvido futuramente com esta finalidade, ele será inevitavelmente alimentado pela informação oriunda da caixa coletora, através de visão computacional, ou pelo usuário que deposita o REEE via preenchimento de um formulário. Desta forma, a própria infraestrutura de comunicação existente na caixa atenderia a este requisito. Também não é escopo deste trabalho o desenvolvimento da roteirização para otimizar a coleta do REEE depositado na caixa coletora. Isto porque a informação de volume armazenado em cada caixa coletora associado com sua localização já fornece dados suficientes para a execução de um algoritmo de rotas a ser adotado. Futuramente, técnicas de inteligência artificial podem ser incorporadas para estimar a probabilidade de alteração do volume armazenado em cada caixa coletora e, com isso, refinar o algoritmo de otimização de rotas.

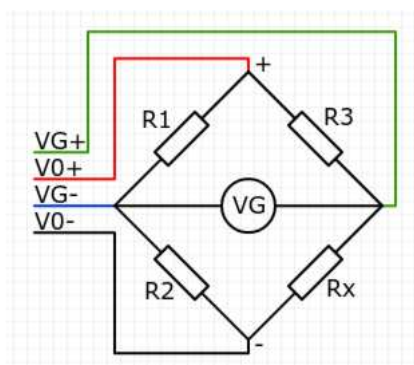
Nas seções seguintes cada um dos requisitos descritos acima será abordado de forma detalhada.

### 4.1 SENSOREAMENTO

De acordo com o cenário em que está inserida esta proposta, os usuários do sistema depositam os REEE nos coletores para que sejam reciclados e, quando atinge o volume especificado, a equipe de logística realiza o transporte destes resíduos para destinação às recicladoras. Conforme mostrado, esta dinâmica requer o conhecimento do volume e da carga armazenada no coletor para que seja identificada a necessidade de esvaziamento do coletor e calculada a melhor rota a ser seguida.

Inicialmente foram identificadas três formas diferentes de sensorar o volume armazenado na caixa coletora, sendo a primeira por peso, a segunda por ultrassom e a terceira por feixe luminoso. A informação do peso fornecida por células de carga, embora seja bastante útil para a logística dos resíduos, não identifica por si só qual o volume ocupado. Neste contexto, é ideal que as células de carga trabalhem em conjunto com sensores como ultrassom e feixe luminoso. Estas células trabalham a partir da sua deformação em função da carga submetida a elas. Isto significa que ao adicionar carga ocorre uma deformação que modifica a resistência elétrica do sensor, aumentando à medida que se deforma e diminuindo à medida que retorna ao estado normal. Por medir a carga através da variação da resistência elétrica são organizadas em um circuito eletrônico conhecido como Ponte de Wheatstone que, conhecendo os resistores que compõem o circuito, afere de forma precisa qual é a resistência da célula de carga e permite que seja calculado o peso. A figura 4 exemplifica este conceito onde R1, R2 e R3 são resistores com valores fixos e Rx é a resistência variável.

Figura 4 – Ponte de Wheatstone

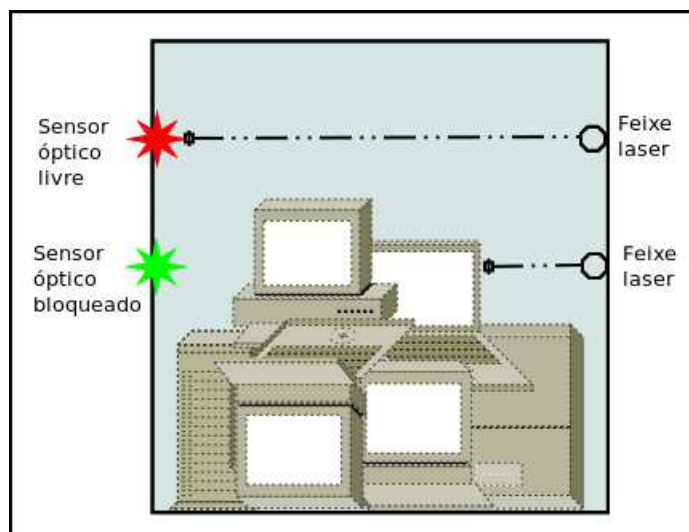


Para que se conheça o volume, o sensoramento através de ultrassom seria indicado. Com este tipo de sensor o volume ocupado no coletor pode ser obtido por meio do tempo em que as ondas sonoras emitidas pelo sensor demoram para refletir nos resíduos e serem percebidas. Entretanto, um ponto negativo para o sensor ultrassom é o fato de ser extremamente suscetível a interferências como temperatura e umidade do ambiente sensorado. Além disso, o REEE tem superfície e tamanho muito variável, o que pode causar a reflexão irregular das ondas sonoras e interferir diretamente na leitura do sensor. Estas características sugerem que também sejam incorporados outros sensores para medir estas condições e calibrar de forma automatizada a aferição de volume. Outro fator que visa reduzir a variação no sensoramento via ultrassom é a adoção de um filtro de média móvel considerando um número razoável de amostras coletadas. Em termos práticos são realizadas N leituras pelo sensor e calculada a média entre os valores lidos.

Uma alternativa ao sensor ultrassônico é a detecção do volume armazenado em visada direta de um feixe luminoso contra um sensor, conforme mostrado pelo croqui representado

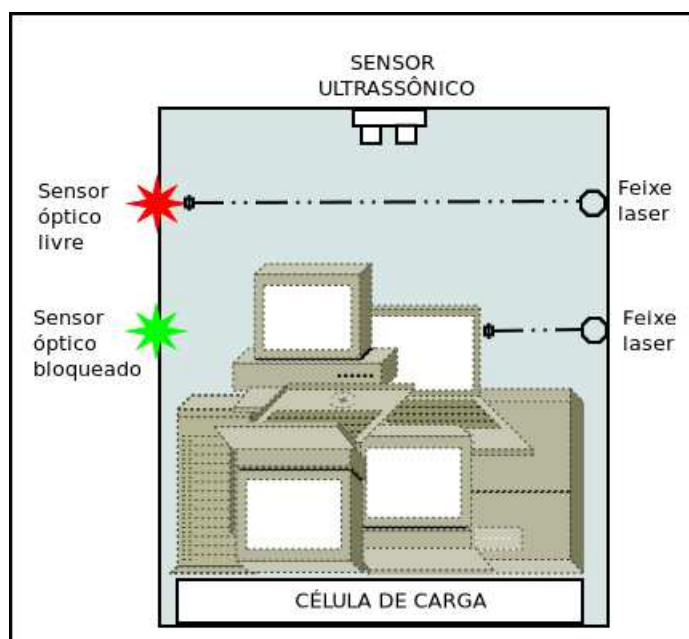
na figura 5. Na ocasião da interrupção deste feixe, é então considerado que o volume sensoreado foi atingido.

Figura 5 – Croqui do sensor óptico



Em comparação com o sensor ultrassônico, o feixe luminoso tende a ser mais preciso na detecção de que determinado volume foi atingido. Porém, uma vez que o volume esteja abaixo ou acima do ponto sensoreado, nenhuma outra informação pode ser aferida sem o auxílio de outros sensores. Por esta razão, é idealmente proposto neste trabalho a adoção da combinação entre a célula de carga, o sensor ultrassônico e o feixe luminoso medindo três níveis de volume. A figura 6 exemplifica esta solução.

Figura 6 – Croqui do coletor com os sensores



## 4.2 COMUNICAÇÃO DE DADOS

Imprescindível ao projeto é a comunicação das informações sensoreadas a um sistema remoto. Pelas características expostas até o presente momento não faria sentido que os dados sensoreados sejam processados nos coletores uma vez que a tomada de decisão requer o conhecimento do estado de toda a rede de coletores distribuídos nos diferentes endereços físicos. Sendo assim, se faz necessária a adoção de uma infraestrutura de rede para que estas informações sejam tratadas de forma útil.

Quando se fala da implantação de uma infraestrutura de rede para comunicação de dados é preciso compreender, além dos requisitos para alcançar o objetivo, as limitações a que se está submetido. Pelo contexto desenhado no projeto, os coletores estão dispostos em diversos locais que contam com características amplamente distintas. Estes locais variam desde estabelecimentos comerciais ou administrativos, com boa infraestrutura energética e telecomunicação, até locais públicos como parques com baixa ou nenhuma infraestrutura energética e, além disso, contam apenas com cobertura de telefonia móvel para a telecomunicação, quando existente.

Observando apenas a localização geográfica dos coletores pode-se dividir o problema em 2 casos distintos: 1) os coletores estão dispostos a distâncias longas entre si, isto é, distâncias maiores do que 100 metros, e 2) os coletores estão dispostos a distâncias curtas entre si, ou seja, até 100 metros. No caso 1 as tecnologias LoRa e NB-IoT são capazes que fornecer cobertura sem a necessidade de repetidores de sinal enquanto ZigBee, Wi-Fi e BLE requerem repetidores para que o sinal de rádio seja recebido na outra ponta da comunicação. Uma alternativa para solucionar esta necessidade é o próprio coletor agir como repetidor propagando sinais emitidos por outros coletores dentro de sua área de alcance. Porém esta abordagem aumenta consideravelmente o consumo energético e, quando inserido num contexto onde o suprimento de energia seja limitado ou a substituição de baterias seja um problema, esta solução não se mostra a melhor opção. Para este trabalho será considerado o caso 1 onde os coletores não trocam informações entre si e dispõem de limitado suprimento de energia. Portanto, as tecnologias Lora e NB-IoT são as mais indicadas no que diz respeito ao alcance.

Para analisar a taxa de transferência e a latência é necessário entender a natureza de sua aplicação. Por se tratar de coletores que monitoram seu volume ocupado para que a coleta seja otimizada não há necessidade de que a informação seja trafegada de forma instantânea, podendo assim ser recebida com um atraso de alguns segundos ou até mesmo minutos. Além disso, a quantidade de dados trafegados é pequena visto que se resumem a volume ocupado e identificador único do coletor. Sendo assim, qualquer uma das 5 tecnologias observadas atende bem a este critério.

Sob o aspecto do consumo energético é possível observar uma grande diferença entre as tecnologias. Importante ressaltar que a tabela 1 apresenta um valor de referência mas

o valor preciso do consumo energético depende de cada projeto. Por limitações financeiras o presente trabalho não dispõe de todas as tecnologias para aferir de maneira precisa o consumo energético de cada conjunto e, por isto, serão considerados os valores de referência para esta discussão. Dito isto e tendo em vista o cenário em que os coletores dispõem de pouco suprimento energético, é traçado o objetivo de maximizar o tempo de vida da bateria. Para que seja alcançado é necessário que, de forma prioritária, o consumo de energia seja mínimo. Abstraindo o circuito eletrônico do conjunto, que seria o mesmo independente da tecnologia, e considerando apenas os valores de referência dos rádios transmissores descritos na tabela 1, observa-se que o LoRa possui o menor consumo de corrente tanto em transmissão quanto em modo *sleep*. Cabe apontar que a busca na redução do consumo energético é encorajada pela literatura quando se fala em *Green Computing* e *Green IoT*. Estes termos visam, conforme descritos nos trabalhos de (MURUGESAN, 2008), (Arshad et al., 2017), (ALBREEM et al., 2017) e (Shaikh; Zeadally; Exposito, 2017) entre outros, a melhoria do impacto socioambiental através da redução da emissão de gases e eficiência energética. Estes fatos indicam que a tecnologia LoRa é a mais indicada para o objetivo de maximizar o tempo de vida da bateria minimizando o consumo de energia.

Por último serão analisados os custos de implantação. Considerando o rádio transmissor, o BLE apresenta os menores custos seguido por LoRa, NB-IoT e ZigBee. Entretanto o BLE e o ZigBee são tecnologias de curto alcance. Por isto, no contexto em que os coletores não trocam informações entre si e estão dispostos a distâncias superiores a 100 metros, são necessárias mais unidades transmissoras para que as informações sejam propagadas. Desvantagem esta que as tecnologias LoRa e NB-IoT, por serem de longo alcance, não apresentam. Importante salientar que o NB-IoT utiliza como base a plataforma LTE e, por isto, embora o transmissor tenha baixo custo, os receptores e repetidores não o tem. Sendo assim, em locais onde a plataforma LTE não está disponível a sua implantação torna o projeto extremamente oneroso enquanto em locais onde exista esta infraestrutura sua utilização é uma alternativa bastante interessante.

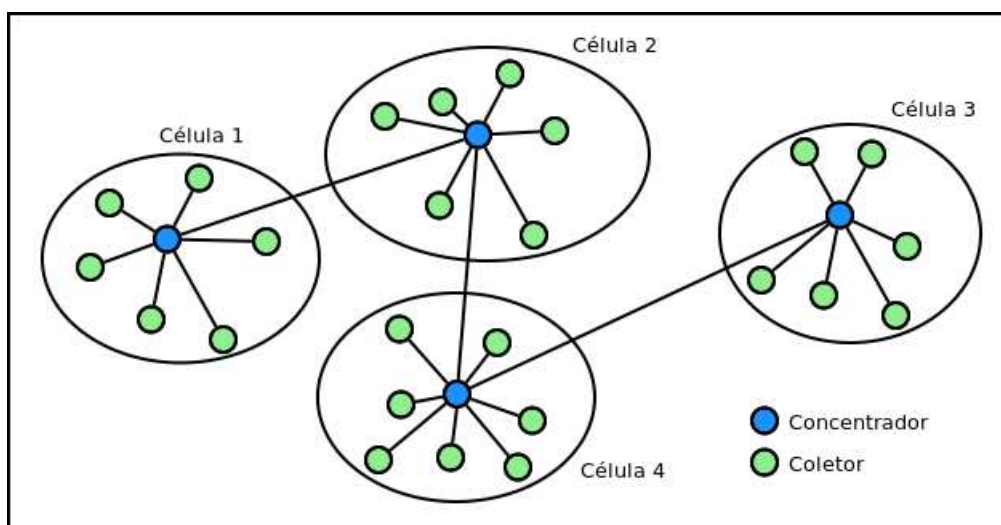
Pelos argumentos expostos podemos observar que todas as tecnologias discutidas aqui tem suas vantagens em relação a outras. Porém, quando submetidas ao contexto da coleta de resíduos eletrônicos onde cada coletor está distante do outro, não há necessidade de trocar informações entre si e a infraestrutura energética e de comunicação é deficitária a escolha da tecnologia a ser adotada se reduz a duas das opções analisadas. Tendo em vista que os custos de implantação de uma rede LTE inviabilizam este trabalho, a escolha pela tecnologia de comunicação LoRa se mostra a mais adequada pelas suas características.

### 4.3 TOPOLOGIA

Como o protocolo de comunicação adotado é o LoRa, e este implementa uma rede sem fio via rádio, a escolha da topologia física pode ser resumida a optar entre as topologias malha, estrela e árvore mostradas na seção 2.3. Tendo em vista a premissa de que os coletores não trocam informações entre si, com os conceitos definidos no capítulo 2 adotar a topologia física em malha não se mostra a opção mais eficiente visto que esta topologia tende a uma complexidade de hardware maior e conseqüentemente um consumo energético mais elevado. A topologia em estrela, por ter um nó concentrador permite que os nós periféricos sejam mais simples e se comuniquem apenas com o concentrador. Por este motivo pode ser uma opção bastante interessante neste sistema. Por último, a topologia em árvore, dependendo do seu tamanho, pode apresentar as mesmas vantagens e desvantagens observadas na topologia em estrela.

Após a análise destas três diferentes topologias físicas é possível concluir que no presente contexto da coleta de resíduos a topologia física mais adequada é em estrela. Isto porque a topologia em malha acrescentaria um custo desnecessário ao projeto e, embora a topologia em árvore seja muito próxima da estrela, a informação transmitida não é tratada de forma hierárquica. De forma prática, todos os coletores de resíduos estão no mesmo nível hierárquico e a organização dos dispositivos nas folhas da árvore ou nas pontas da estrela acaba por ser meramente em função do alcance do rádio ou da quantidade de nós em cada célula. A figura 7 exemplifica a organização em estrela das diferentes células formadas pelos coletores e os concentradores. Convém frisar que o dispositivo concentrador em cada célula estrela se comunica com outros meios através de pontes ou comutadores para fazer com que a informação seja transmitida aos serviços mas isto em nada representa uma topologia em árvore pois define apenas de forma lógica o caminho até o destino.

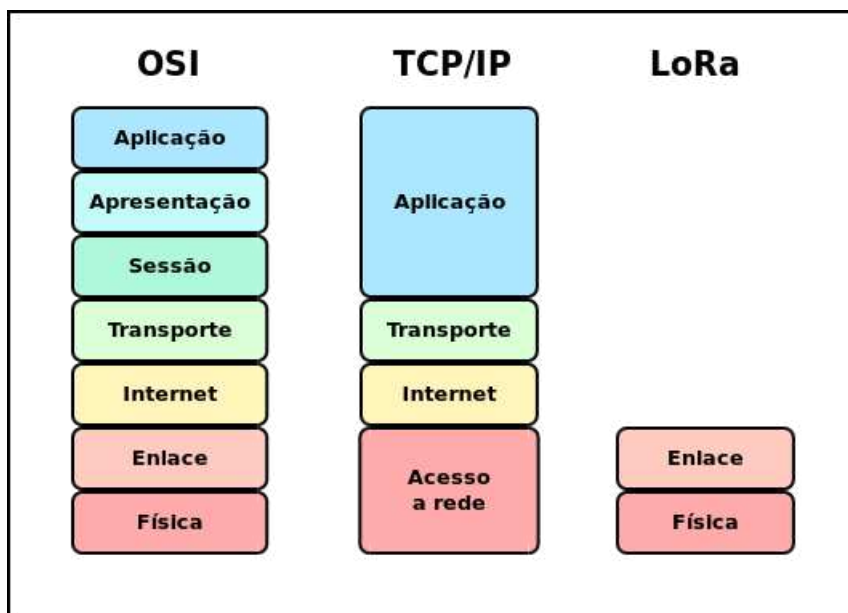
Figura 7 – Topologia proposta neste trabalho



Com a topologia física definida será analisada agora a topologia lógica. Significa que serão observados os aspectos que permitem à rede desempenhar o seu papel de entregar informações aos destinos. Especificamente no contexto dos resíduos, permitir que cada coletor informe aos servidores seu nível ocupado.

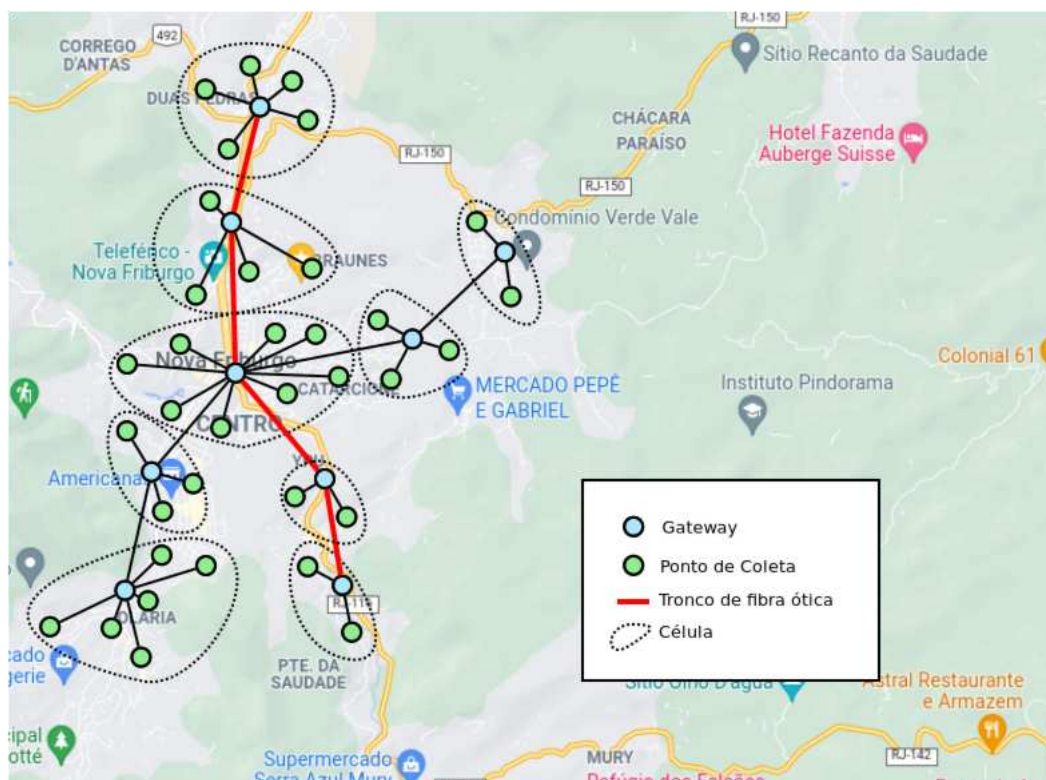
A tecnologia de comunicação LoRa, que foi escolhida para este trabalho e discutida na seção 4.2, define a pilha de protocolos apenas para as camadas física e enlace, conforme mostra a figura 8. Isto permite, além da própria característica desacoplada da pilha de protocolos, que uma infraestrutura de comunicação pré-existente possa ser usada. Com isto, elimina-se a necessidade de implantação de toda uma infraestrutura LoRa desde os coletores até os serviços e, por consequência, tende a tornar a implantação da rede menos onerosa.

Figura 8 – Comparativo entre as pilhas OSI, TCP/IP e LoRa



A cidade de Nova Friburgo, local onde este trabalho está inserido no estado do Rio de Janeiro, conta com um tronco em fibra óptica projetado e implantado exclusivamente para fornecer a infraestrutura necessária aos projetos no âmbito de cidades inteligentes adotados no município. Para que seja possível integrar este trabalho nesta infraestrutura pré-existente, o nó concentrador de cada célula deve ter a capacidade de comutar entre a rede LoRa e a rede de fibra óptica. Num contexto mais amplo, independente da tecnologia de infraestrutura utilizada, o dispositivo precisa ter a capacidade de comutar entre a rede LoRa e qualquer outra por onde seja necessário trafegar a informação. A figura 9 mostra os *gateways* atuando como concentradores nas células ou atuando como extensões para as células posicionadas em locais distantes do tronco de fibra. Desta forma o *gateway* implementa a camada de transporte para que todos os coletores contem com capacidade de comunicação com os serviços remotos através da topologia descrita na figura 9.

Figura 9 – Representação da implantação na cidade de Nova Friburgo



#### 4.4 SERVIÇOS REMOTOS

Cada coletor precisa comunicar seu volume ocupado para que esta informação seja processada e sirva de suporte para a tomada de decisão. Esta comunicação entre o coletor e o gateway é implementada pela infraestrutura LoRa. Entretanto, a informação precisa trafegar do gateway para o serviço e isto é feito através de outros protocolos, diferente da pilha LoRa. Como o conceito de API define uma forma de comunicação entre sistemas distintos por meio de mensagens bem definidas, a integração entre o gateway LoRa e o serviço remoto que receberá os dados de volume será feita através de uma API. Esta API fornecerá apenas a chamada de informe de volume ocupado onde será necessário informar a identificação do coletor e o valor aferido pelos sensores. A API por sua vez não precisa fornecer nenhum dado como resposta, apenas entregará a informação ao serviço para que este possa adicionar o registro temporal da informação, calcular o volume ocupado neste coletor e armazenar a informação final.

Com a API definida é necessário agora especificar qual protocolo será utilizado para a comunicação. Em razão da transmissão dos dados à partir do gateway ocorrer por meio da Internet comum, será utilizada a pilha TCP/IP para que seja mantida a compatibilidade com os sistemas e dispositivos já existentes. Neste cenário tem-se como candidatos os protocolos HTTP, CoAP, MQTT e gRPC por serem bastante difundidos, simples e estáveis.

O protocolo HTTP permite utilizar o padrão REST, que é um formato amplamente

utilizado pelos sistemas disponíveis na Internet. As vantagens de se utilizar este protocolo começam com a maioria dos navegadores de Internet, senão todos, sendo capazes de interpretá-lo, facilitando assim a integração com uma gama maior de sistemas. Além disso, trabalha no modelo requisição/resposta permitindo aos coletores que a chamada à API seja feita apenas quando necessária. A estrutura das mensagens é formada por uma *request-line*, *header* e *body*. A *request-line* informa, além da versão do protocolo, o método caso seja uma requisição ou o código de resposta caso seja uma resposta. Como desvantagem observa-se um maior volume de dados trafegados quando se utiliza este protocolo em comparação a outros desenvolvidos especificamente para o ambiente IoT.

Já o protocolo CoAP por sua vez apresenta uma solução de arquitetura orientada a objetos baseada no modelo HTTP REST, porém mais compacta. Como este protocolo foi projetado para dispositivos com capacidade limitada de processamento ou infraestrutura de rede com taxa de perda ou latência mais elevada, sua adoção no ambiente IoT é bastante justificável. Seguindo esta linha de baixa capacidade o CoAP foi projetado para utilizar o protocolo UDP como protocolo de transporte e com isso a responsabilidade de controlar a conexão para garantir a confiabilidade foi deslocada para a camada de aplicação implementada pelo CoAP.

Uma outra alternativa é o protocolo MQTT. Este protocolo tem como base a baixa quantidade de dados trafegados e o baixo requisito de processamento. Com isso, é muito eficiente em ambientes onde a infraestrutura de rede apresenta taxa de perda ou latência mais elevada. Outra característica interessante é sua arquitetura publicação/assinatura, onde as mensagens são publicadas como um tópico num serviço, chamado de *broker*, e os clientes recebem através do *broker* e conteúdo do tópico assinado. Isto força a existir um serviço a mais na arquitetura e a aplicação se comporta como cliente requisitando os dados ao *broker*. Por último, o MQTT trabalha sobre a pilha TCP, o que garante confiabilidade sem a necessidade de fazer tratamento na camada de aplicação, e permite seu funcionamento de forma assíncrona. Uma desvantagem do MQTT é a necessidade do cliente estar sempre conectado ao serviço. Esta característica força que o *gateway* LoRa trabalhe de forma ininterrupta mantendo sempre ativa a conexão ao serviço.

O último a ser considerado é o gRPC. Este é middleware que implementa a abstração de programação de chamada remota de procedimento (RPC), de código aberto, projetada pelo Google, para obter uma comunicação de alta velocidade entre microsserviços que embute na sua implementação o formato de mensagem *protobuf* (*buffers* de protocolo). O formato *protobuf* é altamente compacto e eficiente na serialização de dados estruturados. Com isso, a quantidade de dados transmitidos é menor do que o habitual quando comparado ao Rest por exemplo, e muito simples de ser codificado no cliente e decodificado no serviço. Outra característica favorável é o protocolo utilizar a pilha TCP. Desta forma, o controle e confiabilidade da conexão ficam na camada de transporte e a camada de aplicação pode se preocupar apenas com os dados. A desvantagem em utilizar gRPC

está no fato de que, como este protocolo é baseado no protobuf, é necessário compilar o protobuf para a linguagem e o dispositivo onde ele será utilizado. No caso em tela, os serviços precisam compilar o protobuf para a mesma linguagem deles de modo que os códigos gerados sejam incorporados ao código do serviço. E pelo lado do cliente, o protobuf precisa ser compilado para o microcontrolador que será utilizado. Uma vez vencida esta barreira, o gRPC se mostra um forte candidato a ser escolhido pelo seu desempenho e simplicidade.

Com base no exposto sobre os quatro protocolos avaliados é correto dizer que todos desempenhem o papel de protocolo de aplicação para a API. Idealmente, o gRPC se mostra o mais promissor dentre os quatro considerando a facilidade de integração do código gerado e o nível de compactação dos dados. Porém, isto só é verdade se for vencida a barreira de compilação do protobuf para o hardware e a linguagem onde será utilizado.

## 5 EXPERIMENTO

Partindo da proposta discutida no capítulo 4, propõe-se o experimento descrito a seguir com o objetivo de validar a proposta apresentada e mostrar sua aplicação. Uma caixa em madeira é construída para acomodar os resíduos e fazer a montagem dos sensores bem como do rádio. Este rádio tem a função de sensorear e transmitir estas informações para o *gateway*. Outro rádio é colocado distante da caixa para atuar como *gateway* e comutar as informações da caixa para o serviço remoto. Um serviço remoto extremamente simples recebe estas informações geradas pela caixa e as exibe, confirmando assim o correto funcionamento da arquitetura. Com este experimento se avalia o sensoreamento, a obtenção e comunicação da informação sensoreada através de uma rede, a sua topologia da rede e a integração com um serviço remoto. Não é objetivo deste trabalho, tão pouco deste experimento, o desenvolvimento de um sistema que forneça o serviço de tratamento destas informações.

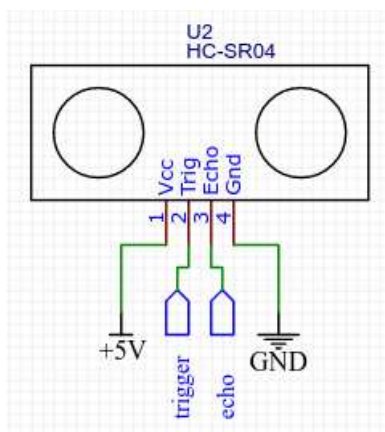
### 5.1 MONTAGEM DO EXPERIMENTO

Com o experimento definido e sabendo que o protocolo de comunicação proposto na seção 4.2 é o LoRa, foram adquiridas duas unidades da placa de prototipagem ESP32-LoRa OLED. Esta placa conta com um display OLED que facilita a visualização de informações, com o rádio LoRa integrado, além dos rádios WiFi e BLE e com o microcontrolador ESP-32. Com esta placa também é satisfeita a necessidade de implementar um *gateway* para a comutação da rede LoRa com a Internet comum, conforme a seção 4.3 descreve. Apesar de ter sido discutido na seção 4.2 que o WiFi não era o protocolo mais indicado para este trabalho, o uso de outras tecnologias de rede como ethernet ou fibra óptica é inviável para este experimento proposto em função do seu custo e complexidade. Como alterar a tecnologia de rede significa alterar a pilha de protocolos e esta é justamente a finalidade do desacoplamento das camadas que compõem esta pilha, adotar o WiFi pela limitação de custo não traz nenhum prejuízo ao experimento.

Uma vez definida a placa de prototipagem e as tecnologias envolvidas, olha-se agora para o sensoreamento. A seção 4.1 indica que o ideal é a combinação de diferentes tipos de sensores para complementar a informação e garantir maior precisão. Infelizmente a limitação financeira novamente não permitiu a aquisição das células de carga e, além disso, a falta de maiores conhecimentos em eletrônica por este aluno gerou um erro na montagem do circuito e alguns componentes do sensor ótico foram inutilizados. Por isso apenas o sensor ultrassônico foi adotado no experimento. Sua montagem é muito simples e segue conforme a figura 10.

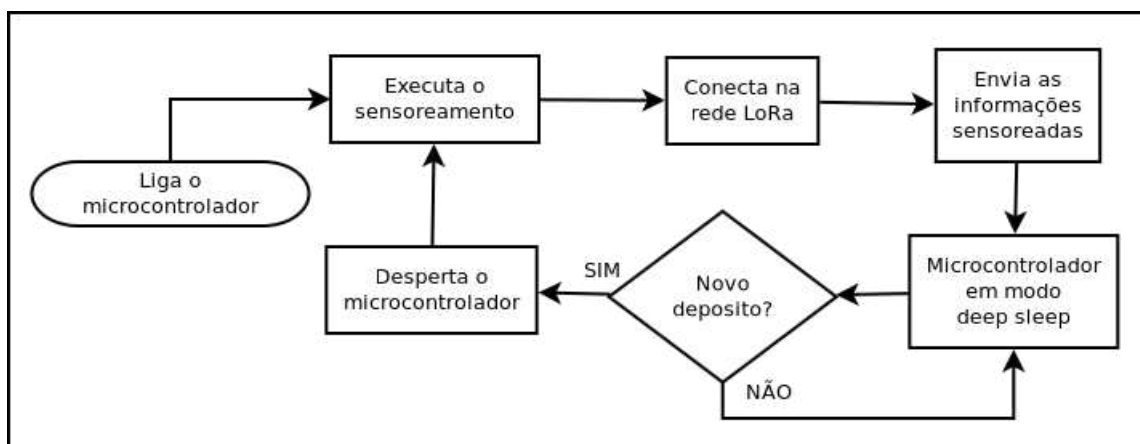
Outra vantagem do microcontrolador ESP-32 é sua capacidade de economia de ener-

Figura 10 – Diagrama esquemático do sensor ultrassônico



gia. Ele dispõe de modos de operação chamados de *deep sleep* onde apenas ficam ativas as funções do co-processador ULP (*Ultra Low Power*). Neste modo, o processador principal se desliga junto com os rádios e, conseqüentemente, o consumo energético diminui de forma drástica. Existem algumas formas para restabelecer o funcionamento do microcontrolador após ele ser colocado em modo *deep sleep*. A adotada neste trabalho é a *external wakeup*, onde a transição de nível lógico aplicada no GPIO que foi configurado via código é percebida pelo co-processador ULP. Por sua vez o co-processador restabelece o funcionamento normal do microcontrolador. A figura 11 mostra o fluxograma do comportamento do nó coletor.

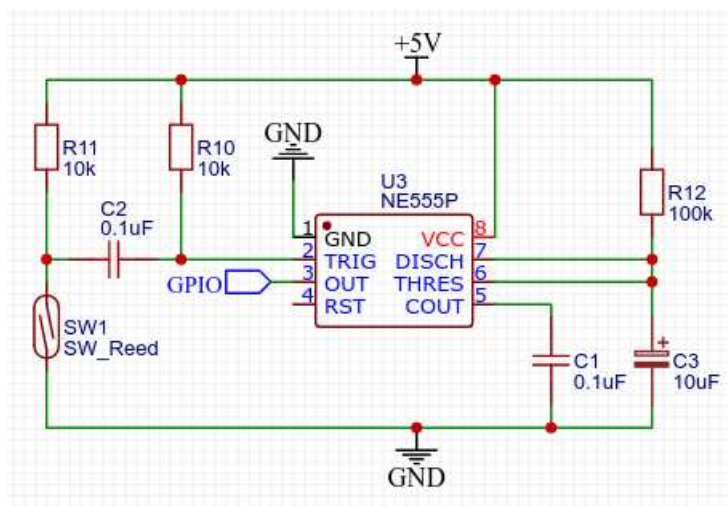
Figura 11 – Fluxograma de funcionamento do nó coletor



Como mostrado no fluxograma da figura 11, cada vez que um novo resíduo é depositado no coletor o microcontrolador é despertado. Este comportamento é implementado com uma chave magnética acoplada na porta do coletor. Este experimento não dispõe de recursos para a confecção do coletor nestes padrões, por isso, foi utilizado um ímã que é manualmente afastado e aproximado do sensor magnético simulando a abertura e o fechamento da porta.

O circuito eletrônico mostrado na figura 12 foi desenvolvido com a finalidade de controlar o sensoreamento da porta do coletor e emitir o sinal eletrônico responsável por despertar o microcontrolador. Este circuito utiliza o componente NE555 em uma configuração mono-estável fazendo com que quando ocorre a transição de sinal de entrada para nível lógico alto, um sinal de saída seja emitido em um pulso único com duração aproximada de 1s. As transições do sinal de entrada deste circuito são feitas pela abertura e fechamento da porta do coletor. Quando aberta o sinal de entrada no circuito transiciona para nível lógico baixo e quando fechada este sinal de entrada transiciona para nível lógico alto. Com isto um pulso único é emitido por este circuito e provoca uma transição para nível lógico alto no GPIO, que desperta o microcontrolador. Após este pulso a saída do circuito permanece em nível lógico baixo até que a porta do coletor seja novamente fechada.

Figura 12 – Diagrama esquemático do sensor de abertura e fechamento da porta do coletor

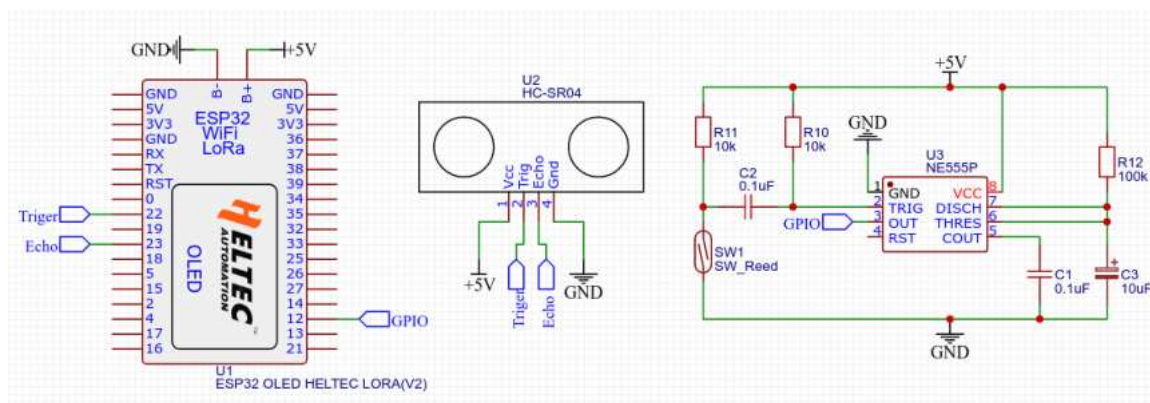


O diagrama esquemático completo do nó coletor é mostrado na figura 13. Este diagrama contém a montagem do sensor ultrassônico com a chave magnética e a placa de prototipagem ESP-32 LoRa. Por último, o código utilizado para o microcontrolador segue o fluxograma descrito na figura 11 e pode ser visto no anexo 1. Neste código apenas os dados de identificação do coletor e volume sensoreado são enviados para o *gateway*.

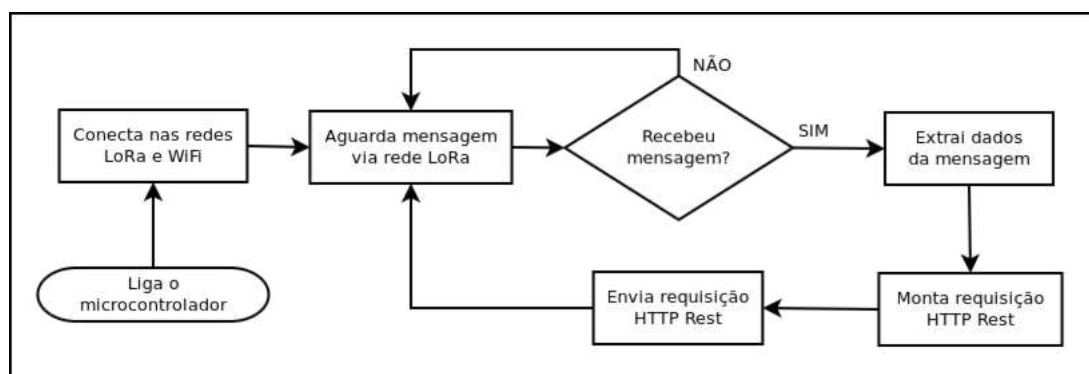
O *gateway* LoRa é um pouco mais simples comparado ao nó coletor. O *gateway* também faz uso da placa de prototipagem Esp-32 LoRa mas, diferente do nó coletor, o *gateway* não precisa de sensores e por isso não tem componentes eletrônicos adicionais. Sua complexidade está na recepção das informações via rede LoRa, transcrição da mensagem e envio para o serviço remoto.

Por motivo de simplicidade e facilidade de implementação o protocolo escolhido para a camada de aplicação neste trabalho é o HTTP Rest. Como discutido na seção 4.4, apesar deste protocolo ser o mais custoso sob o aspecto do uso de rede e do processamento, seu uso é extremamente simples sendo este o principal motivo, senão o único, para sua

Figura 13 – Diagrama esquemático do nó coletor



escolha. Desta forma, a figura 14 mostra o fluxograma do comportamento do *gateway*. O código utilizado no microcontrolador do *gateway* segue este fluxograma e pode ser visto no apêndice 2.

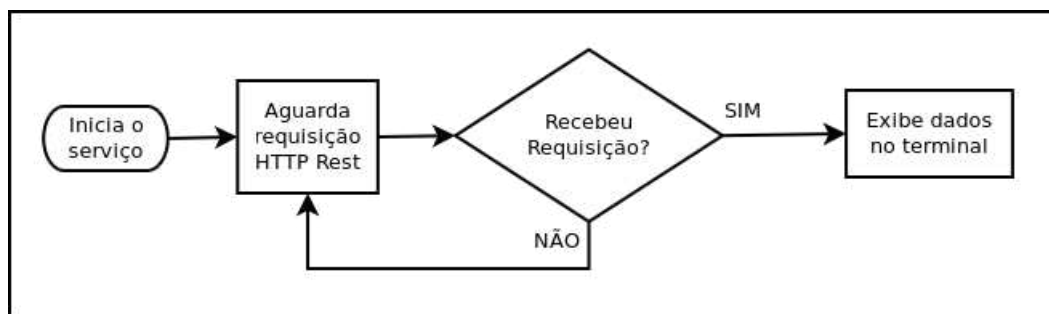
Figura 14 – Fluxograma de funcionamento do *gateway*

Por último na etapa de montagem do experimento tem-se o serviço remoto. Como não é escopo deste trabalho o desenvolvimento de um sistema para o tratamento das informações dos coletores mas apenas a definição de uma arquitetura que permita sua integração a um sistema remoto, foi desenvolvido um serviço que apenas recebe a requisição HTTP Rest e exibe as informações no terminal. Com isso, é validado o sistema proposto. A figura 15 mostra o fluxograma do funcionamento do serviço remoto proposto e seu código pode ser visto no apêndice 3.

## 5.2 EXECUÇÃO DO EXPERIMENTO

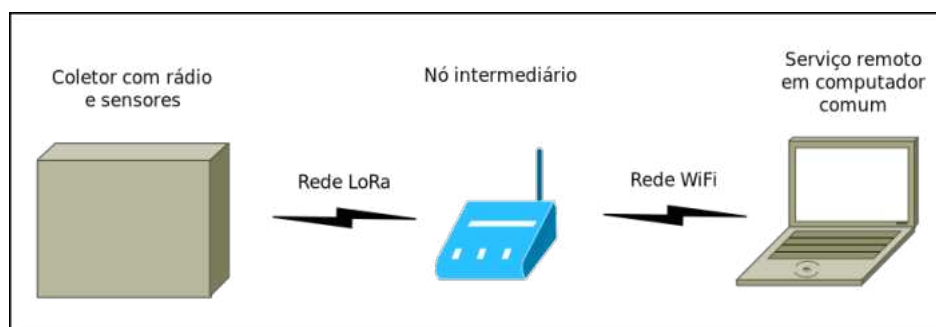
O experimento foi executado utilizando os elementos construídos na seção 5.1. Desta forma, o serviço remoto foi executado em um computador pessoal conectado a uma rede WiFi criada apenas para esta finalidade. Nesta mesma rede WiFi também foi conectado o *gateway* LoRa, que recebe os dados do coletor e os envia ao serviço. O coletor, por sua vez, envia os dados via rede LoRa até o *gateway*. A caixa coletora tem uma altura interna

Figura 15 – Fluxograma de funcionamento do serviço remoto



de 49 cm e a medição realizada pelo sensor ultrassônico, feita com ela vazia, indicou 46,34cm de distância. Um valor muito próximo da realidade considerando a posição e o tamanho do sensor, que é em torno de 2cm. Esta informação foi corretamente recebida pelo *gateway* e em seguida pelo serviço remoto. A figura 16 ilustra esta configuração.

Figura 16 – Configuração do experimento



Para simular descartes de eletrônicos um volume de aproximadamente 12cm de altura foi inserido no coletor. Um ímã foi afastado da chave magnética e reaproximado, despertando assim o microcontrolador do coletor que executou o sensoreamento aferindo uma distância de 34,77cm e transmitiu a mensagem conforme esperado. Como o valor inicial era de 46,34cm e o valor inserido de 12cm, o valor obtido de 34,77cm é pertinente e dentro do esperado. Fez-se então uma nova simulação de descartes inserindo outro volume de 12cm de altura. Após novamente afastar e aproximar o ímã, o coletor aferiu a distância de 22,80cm e transmitiu este dado até o serviço remoto de forma correta. Mais uma vez o volume aferido foi pertinente ao inserido, demonstrando com isso um grau de confiança aceitável. Um terceiro descarte foi simulado, desta vez com dimensões menores e uma altura de 6,5cm. O procedimento foi repetido e o serviço remoto exibiu o valor de 16,66cm. Mais uma vez os valores aferidos se mostram muito próximos da realidade.

Também foi avaliado no experimento a corrente elétrica consumida pelos microcontroladores e circuitos. A tabela 2 mostra que o conjunto utilizado na caixa coletora apresentou consumo máximo de 50mA durante seu funcionamento e 14mA no período em que permanece em modo *deep sleep*. Apesar de ter sido apontado na seção 4.2 um consumo de 1 $\mu$ A para o rádio LoRa neste modo de operação, os 14mA observados decor-

rem do circuito eletrônico do conjunto formado por placa de prototipagem e circuito dos sensores.

Modo	Consumo
operação	40mA a 50mA
<i>sleep</i>	14mA

Tabela 2 – Consumo de corrente por modo de operação

O tempo entre o microcontrolador despertar, realizar o sensoreamento, enviar a informação e entrar em *sleep* foi medido em torno de 7s. Este é o procedimento de maior consumo energético conforme mostra a tabela 2 e só ocorre quando a tampa do coletor é fechada. Significa dizer que durante todo o resto do tempo tem-se o consumo médio de 14mA com o conjunto em modo *deep sleep*. Se for considerado que durante 10% do tempo o coletor funciona com um consumo de 50mA e nos outros 90% do tempo com consumo de 14mA, então temos que  $consumo_{medio} = 10\% \times 50mA + 90\% \times 14mA = 17,6mA$ .

Para saber o tempo que uma bateria dura alimentando um dispositivo é preciso conhecer sua capacidade e a corrente drenada. A equação do tempo teórico de descarga é dado por  $tempo_{teorico} = \frac{capacidade(mAh)}{consumo(mA)}$ . Além disso, ocorre a dissipação de parte da energia armazenada por causas externas, como por exemplo temperatura ou degradação. Desta forma o tempo de duração estimado é  $tempo_{estimado} = tempo_{teorico} \times (1 - taxa_{dissipada})$  e, com isso, pode ser definida a tabela 3 a seguir. Esta tabela representa algumas durações estimadas com base numa taxa de dissipação arbitrária.

Capacidade (mAh)	Tempo teórico (horas)	Taxa dissipada	Tempo estimado (horas)	Tempo estimado (dias)
2000mAh	113,6	10%	102,3	4,3
		20%	90,9	3,8
5000mAh	284,1	10%	255,7	10,6
		20%	227,3	9,5
10000mAh	568,2	10%	511,4	21,3
		20%	454,5	18,9

Tabela 3 – Tempos de duração da bateria com base numa corrente de 17,6mAh

Analisando os valores obtidos, percebe-se que embora o tempo de vida de 4 dias seja relativamente curto, este consumo médio representa que a tampa do coletor foi fechada aproximadamente 51 vezes no período de uma hora. Ou ainda, 1224 vezes em um dia, o que pode indicar que os resíduos depositados esgotam o volume do coletor em um único dia. Por outro lado, no melhor cenário descrito a bateria teve duração estimada de 21 dias.

Avaliando uma segunda hipótese onde o funcionamento seja descrito por 1% do tempo em operação e 99% do tempo em modo *deep sleep* tem-se então  $consumo_{medio} = 1\% \times$

$50mA + 99\% \times 14mA = 14,36mA$ . Este novo consumo médio significa que no período de uma hora a tampa do coletor foi fechada aproximadamente 5 vezes, ou ainda, 123,4 vezes em um dia, o que é bastante razoável. Com estes valores e mantendo os mesmos parâmetros arbitrados na tabela 4, são obtidas novas estimativas para o tempo de duração.

Capacidade (mAh)	Tempo teórico (horas)	Taxa dissipada	Tempo estimado (horas)	Tempo estimado (dias)
2000mAh	139,9	10%	125,9	5,2
		20%	111,9	4,7
5000mAh	349,6	10%	314,7	13,1
		20%	279,7	11,6
10000mAh	699,3	10%	629,4	26,2
		20%	559,4	23,3

Tabela 4 – Tempos de duração da bateria com base numa corrente de 14,3mA

Comparando as tabelas 3 e 4 e o que suas informações representam, observa-se que embora o consumo médio menor reflita num maior tempo estimado a diferença não é expressiva pois no menor intervalo a diferença está em 1 dia e no maior intervalo está em 5 dias. Resta evidente que embora os dados apresentados mostrem viabilidade da proposta esta avaliação precisa incluir o volume médio por descarte de modo que seja possível mensurar se o ciclo de substituição da bateria é aderente ao ciclo da logística de retirada dos resíduos. Este dado pode ser obtido com a execução do experimento em condições reais.

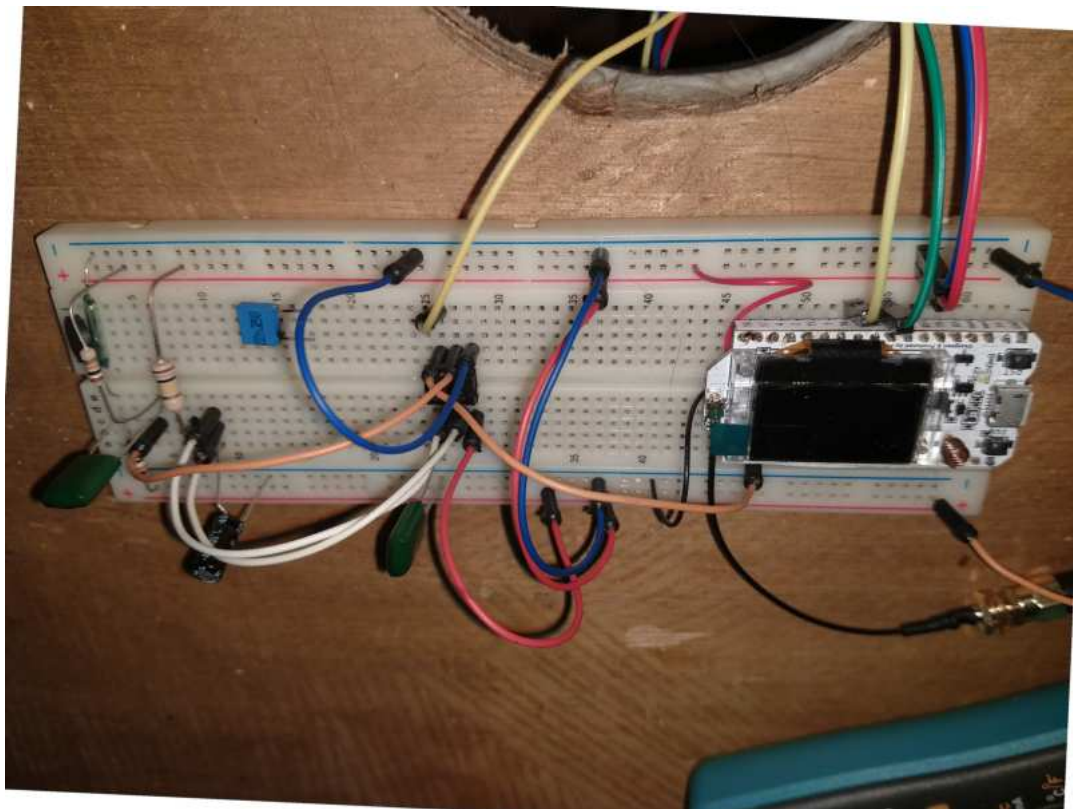
Também é interessante notar que uma pilha recarregável padrão AA, ou seja, uma pilha comum, conta com uma capacidade de 2000mAh ou até mais. Além deste, diversos tipos de baterias estão disponíveis no mercado e podem fornecer a capacidade necessária ao funcionamento do ponto de coleta. Portanto, um projeto adequado de uma bateria que atenda a esta arquitetura é bastante factível. Um outro fator favorável à arquitetura é a possibilidade de inclusão de fontes renováveis de energia, como por exemplo células fotoelétricas. Com isto a necessidade de manutenção poderia reduzir de forma drástica.

A sequência de imagens a seguir mostra o experimento montado e recebendo os dados provenientes da caixa coletora. Na figura 17 observa-se a montagem na *protoboard* do conjunto do nó coletor composto pelo microcontrolador programado para a função de coletor, o sensor sônico para detecção do volume, o sensor magnético para detecção do depósito e o restante dos componentes eletrônicos necessários ao projeto.

A figura 18 mostra a montagem do *gateway*. Apenas o microcontrolador programado como *gateway* é necessário para esta montagem e não requer nenhum outro componente eletrônico.

Por último a figura 19 mostra as informações recebidas exibidas no terminal pelo serviço remoto. Os valores de distância exibidos decrescem conforme esperado porque à

Figura 17 – Imagem da montagem na *protoboard* do conjunto do nó coletor composto pelo microcontrolador e sensores.



medida que os resíduos são depositados, a distância para o sensor diminui.

Figura 18 – Imagem da montagem do microcontrolador como *gateway*

Figura 19 – Imagem das informações exibidas no terminal pelo serviço remoto.

```
[luiz@Luiz-Bart projeto-final]$ ./main
method: POST
ID: 142279
Distance: 46.34
method: POST
ID: 142279
Distance: 34.77
method: POST
ID: 142279
Distance: 34.77
method: POST
ID: 142279
Distance: 22.8
```

## 6 CONCLUSÃO

Tal como queria ser demonstrado, a informação sensoreada foi transmitida de forma satisfatória desde a caixa coletora até o serviço remoto. Com isso foi confirmada a viabilidade da arquitetura proposta uma vez que a informação foi gerada, transmitida e interpretada corretamente. O método de sensoreamento também apresentou desempenho bastante satisfatório visto que o volume alterado foi sensoreado com um bom nível de precisão. Por último, o consumo de corrente do conjunto, embora precise de um estudo específico, mostra valores com bons padrões para a arquitetura proposta passível de melhorias.

Conforme apontado na seção 4.1, o sensor ultrassônico apresenta erros que são decorrentes da superfície que reflete a onda, seja em razão do material ou sua irregularidade, e até mesmo temperatura e umidade do ambiente interferem na sua precisão. Para aumentar o grau de precisão da caixa coletora é possível combinar outros tipos de sensores, como por exemplo, o feixe luminoso e as células de carga. Para isto basta apenas adequar o circuito eletrônico e os códigos dos microcontroladores e do serviço remoto além de, obviamente, adequar a construção da caixa coletora.

Apesar do experimento ter obtido avaliação positiva, é importante ressaltar que é necessário realizar um experimento em condições reais de uso. Todos os volumes utilizados são superfícies planas e não houve movimentação de forma que alterasse o armazenamento. Por isso, é correto afirmar que os diferentes resíduos depositados, em seus diversos tamanhos, formatos e materiais, pode provocar diferença na precisão do sensoreamento. Em razão da pandemia de COVID-19 um experimento avaliando estas hipóteses não pode ser realizado.

## 7 TRABALHOS FUTUROS

Com o intuito de aprimorar este trabalho, algumas sugestões são apontadas a seguir como possíveis trabalhos futuros. No que tange o projeto como um todo, ocorre a necessidade de implementar a proposta em um ambiente real. Assim, é possível validar o sensoramento com os variados resíduos que serão efetivamente depositados nos coletores.

Para o sistema remoto, sugere-se que seja realizado um trabalho de desenvolvimento de um sistema de tomada de decisão onde a informação recebida dos nós coletores seja tratada e ações sejam tomadas a partir destas informações. Por exemplo, emitir alertas aos operadores logísticos que determinados coletores atingiram o volume necessário para serem esvaziados.

Outra sugestão de melhoria ao sistema remoto é a otimização de rotas de coleta dos resíduos nas caixas coletoras visando reduzir custos com o transporte dos resíduos bem como reduzir a emissão de poluentes oriunda dos veículos empregados nesta tarefa. Para isto, não apenas os coletores cheios podem ser considerados nestas rotas como também os coletores próximos a encher e os coletores posicionados geograficamente perto. Dados complementares como o peso, além do volume, podem ser necessários e conseqüentemente outros sensores precisarão ser incluídos.

A adoção de visão computacional para identificação precisa do que é depositado é uma melhoria necessária ao projeto para aumentar a usabilidade e a adoção da comunidade. Isto porque facilitaria o processo de cadastro dos descartes, tornando mais simples atrativo e atrativo este processo. Esta melhoria levanta a discussão de qual decisão de projeto entre processamento local ou remoto para a visão computacional deve ser tomada. E conseqüentemente, poderá impactar na infraestrutura de comunicação e alterar a arquitetura proposta.

**REFERÊNCIAS**

- ALBREEM, M. et al. Green internet of things (iot): An overview. In: . [S.l.: s.n.], 2017.
- Arshad, R. et al. Green iot: An investigation on energy saving practices for 2020 and beyond. **IEEE Access**, v. 5, p. 15667–15681, 2017.
- HUANG, J. et al. A novel deployment scheme for green internet of things. **Internet of Things Journal, IEEE**, v. 1, p. 196–205, 04 2014.
- KANG, K. D. et al. Electronic waste collection systems using internet of things (iot): Household electronic waste management in malaysia. **Journal of Cleaner Production**, v. 252, p. 119801, 2020. ISSN 0959-6526. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0959652619346712>.
- Longhi, S. et al. Solid waste management architecture using wireless sensor network technology. In: **2012 5th International Conference on New Technologies, Mobility and Security (NTMS)**. [S.l.: s.n.], 2012. p. 1–5.
- MURUGESAN, S. Harnessing green it: Principles and practices. **IT Professional**, v. 10, p. 24 – 33, 02 2008.
- POPLI, S.; JHA, R. K.; JAIN, S. A survey on energy efficient narrowband internet of things (nbiot): Architecture, application and challenges. **IEEE Access**, v. 7, p. 16739–16776, 2019.
- Shaikh, F. K.; Zeadally, S.; Exposito, E. Enabling technologies for green internet of things. **IEEE Systems Journal**, v. 11, n. 2, p. 983–994, 2017.
- Shyam, G. K.; Manvi, S. S.; Bharti, P. Smart waste management using internet-of-things (iot). In: **2017 2nd International Conference on Computing and Communications Technologies (ICCT)**. [S.l.: s.n.], 2017. p. 199–203.
- SOUSA, P. V.; COSTA, M. Protótipo de lixeira inteligente no contexto das smart cities e da internet das coisas.
- Xu, L. D.; He, W.; Li, S. Internet of things in industries: A survey. **IEEE Transactions on Industrial Informatics**, v. 10, n. 4, p. 2233–2243, 2014.
- Zhu, C. et al. Green internet of things for smart world. **IEEE Access**, v. 3, p. 2151–2162, 2015.

## APÊNDICES

## APÊNDICE A – CÓDIGOS DOS MICROCONTROLADORES.

Os códigos utilizados no nó coletor e no *gateway* foram desenvolvidos em linguagem C/C++ utilizando a IDE do Arduino. A linguagem foi escolhida por ser a utilizada pela IDE que, por sua vez, foi selecionada pela sua praticidade e ampla cobertura de compilação do código para diversas placas de prototipagem. Está incluído nesta cobertura a placa Esp-32 LoRa OLED, que foi utilizada neste trabalho.

A seguir é apresentado o código do microcontrolador que desempenha a função de nó coletor dos resíduos.

Código 1 – Código do coletor para sensorar e enviar a informação ao gateway.

```

/*
  This is a simple example show the Heltec.LoRa sended data in OLED.

  The onboard OLED display is SSD1306 driver and I2C interface. In order
  to make the
  OLED correctly operation, you should output a high-low-high(1-0-1)
  signal by soft-
  ware to OLED's reset pin, the low-level signal at least 5ms.

  OLED pins to ESP32 GPIOs via this connectin:
  OLED_SDA -- GPIO4
  OLED_SCL -- GPIO15
  OLED_RST -- GPIO16

  by Aaron.Lee from HelTec AutoMation, ChengDu, China

  https://heltec.org

  this project also realess in GitHub:
  https://github.com/Heltec-Aaron-Lee/WiFi_Kit_series
*/

#include "heltec.h"
#include "images.h"

#define BAND      915E6 //you can set band here directly,e.g. 868E6,915E6

#define ID 142279

const int trigPin = 22;
const int echoPin = 23;
const int wakeupPin = 12;

```

```

const int nSamples = 50;

unsigned int counter = 0;
String rssi = "RSSI --";
String packSize = "--";
String packet ;

void logo()
{
  Heltec.display->clear();
  Heltec.display->drawXbm(0,5,logo_width,logo_height,logo_bits);
  Heltec.display->display();
}

float _senseVolume(){
  trigPulse();
  long pulse = pulseIn(echoPin, HIGH);
  float cm = pulse/58.82;
  return cm;
}

void trigPulse(){
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
}

float senseVolume(){
  float acc = 0;
  for(int i = 0; i < nSamples; i++){
    acc += _senseVolume();
    delay(100);
  }

  return acc / nSamples;
}

void setup()
{
  Heltec.begin(true /*DisplayEnable Enable*/, true /*Heltec.Heltec.
    Heltec.LoRa Disable*/, true /*Serial Enable*/, true /*PABOOST
    Enable*/, BAND /*long BAND*/);
}

```

```

Heltec.display->init();
Heltec.display->flipScreenVertically();
Heltec.display->setFont(ArialMT_Plain_10);
logo();
delay(1500);
Heltec.display->clear();

Heltec.display->drawString(0, 0, "Heltec.LoRa Initial success!");
Heltec.display->display();

pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin, INPUT); // Sets the echoPin as an Input
digitalWrite(trigPin, LOW);
delay(1000);

// esp_sleep_enable_timer_wakeup(60000000);
esp_sleep_enable_ext0_wakeup(GPIO_NUM_12, 1);
}

void loop()
{
  float distance = _senseVolume();

  Heltec.display->clear();
  Heltec.display->setTextAlignment(TEXT_ALIGN_LEFT);
  Heltec.display->setFont(ArialMT_Plain_10);

  Heltec.display->drawString(0, 0, "ID: " + String(ID));
  Heltec.display->drawString(0, 10, "Distance: " + String(distance));
  Heltec.display->display();

  // send packet
  LoRa.beginPacket();

  /*
  * LoRa.setTxPower(txPower, RFOUT_pin);
  * txPower -- 0 ~ 20
  * RFOUT_pin could be RF_PACONFIG_PASELECT_PABOOST or
  *   RF_PACONFIG_PASELECT_RFO
  *   - RF_PACONFIG_PASELECT_PABOOST -- LoRa single output via PABOOST,
  *     maximum output 20dBm
  *   - RF_PACONFIG_PASELECT_RFO      -- LoRa single output via RFO_HF /
  *     RFO_LF, maximum output 14dBm
  */
  LoRa.setTxPower(14, RF_PACONFIG_PASELECT_PABOOST);
  LoRa.print(String(ID));
  LoRa.print("|");

```

```

    LoRa.print(String(distance));
    LoRa.print("|");
    // LoRa.write(ID, sizeof(ID));
    // LoRa.write(distance, sizeof(distance));
    LoRa.endPacket();

    counter++;
    digitalWrite(LED, HIGH);    // turn the LED on (HIGH is the voltage
                               level)
    delay(1000);                // wait for a second
    digitalWrite(LED, LOW);    // turn the LED off by making the voltage
                               LOW
    delay(1000);                // wait for a second

    esp_deep_sleep_start();
}

```

A seguir é apresentado o código do microcontrolador que desempenha a função de gateway para receber as mensagens dos nós e enviar as informações ao serviço remoto.

Código 2 – Código do gateway para receber as mensagens dos coletores e repassar ao serviço remoto.

```

/*
   This is a simple example show the Heltec.LoRa recived data in OLED.

   The onboard OLED display is SSD1306 driver and I2C interface. In order
   to make the
   OLED correctly operation, you should output a high-low-high(1-0-1)
   signal by soft-
   ware to OLED's reset pin, the low-level signal at least 5ms.

   OLED pins to ESP32 GPIOs via this connecthin:
   OLED_SDA -- GPIO4
   OLED_SCL -- GPIO15
   OLED_RST -- GPIO16

   by Aaron.Lee from HelTec AutoMation, ChengDu, China

   www.heltec.cn

   this project also realess in GitHub:
   https://github.com/Heltec-Aaron-Lee/WiFi_Kit_series
*/
#include "heltec.h"
#include "images.h"
#include <WiFi.h>
#include <HTTPClient.h>

```

```

const char* ssid = "Seraph";
const char* password = "gtwwlsluiz";
const char* url = "http://192.168.0.105:8080";

#define BAND      915E6  //you can set band here directly, e.g. 868E6, 915E6
                        ,433E6
String rssi = "RSSI --";
String packSize = "--";
String packet;

int senderId;
float distance;

void logo(){
  Heltec.display->clear();
  Heltec.display->drawXbm(0,5,logo_width,logo_height,logo_bits);
  Heltec.display->display();
}

void WIFISetup(){
  WiFi.disconnect(true);
  delay(100);
  WiFi.mode(WIFI_STA);
  WiFi.setAutoConnect(true);
  WiFi.begin("Seraph", "gtwwlsluiz");
  delay(100);

  byte count = 0;
  while(WiFi.status() != WL_CONNECTED && count < 10){
    count ++;
    delay(500);
    Heltec.display -> drawString(0, 0, "Connecting...");
    Heltec.display -> display();
  }

  Heltec.display -> clear();
  if(WiFi.status() == WL_CONNECTED){
    Heltec.display -> drawString(0, 0, "Connecting... OK");
    Heltec.display -> display();
  }
  else{
    Heltec.display -> clear();
    Heltec.display -> drawString(0, 0, "Connecting... Failed");
    Heltec.display -> display();
  }
  Heltec.display -> drawString(0, 10, "WIFI Setup done");
}

```

```

    Heltec.display -> display();
    delay(5000);
}

void LoRaData(){
    Heltec.display->clear();
    Heltec.display->setTextAlignment(TEXT_ALIGN_LEFT);
    Heltec.display->setFont(ArialMT_Plain_10);
    Heltec.display->drawString(0 , 15 , "Received "+ packSize + " bytes");
    Heltec.display->drawStringMaxWidth(0 , 26 , 128, packet);
    Heltec.display->drawString(0, 0, rssi);
    Heltec.display->display();
}

void cbk(int packetSize) {
    packet = "";
    packSize = String(packetSize,DEC);
    for (int i = 0; i < packetSize; i++) { packet += (char) LoRa.read(); }
    rssi = "RSSI " + String(LoRa.packetRssi(), DEC) ;
    LoRaData();
}

void setup() {
    // Serial.begin(115200);

    //WIFI Kit series V1 not support Vext control
    Heltec.begin(true /*DisplayEnable Enable*/, true /*Heltec.Heltec.
        Heltec.LoRa Disable*/, true /*Serial Enable*/, true /*PABOOST
        Enable*/, BAND /*long BAND*/);

    Heltec.display->init();
    Heltec.display->flipScreenVertically();
    Heltec.display->setFont(ArialMT_Plain_10);
    logo();
    delay(1500);
    Heltec.display->clear();

    WIFISetup();

    Heltec.display->clear();
    Heltec.display->drawString(0, 0, "Heltec.LoRa Initial success!");
    Heltec.display->drawString(0, 10, "Wait for incoming data...");
    Heltec.display->display();
    delay(1000);
    // LoRa.onReceive(cbk);
    LoRa.receive();
}

```

```

void loop() {
  int packetSize = LoRa.parsePacket();
  if (packetSize) {
    int receivedRssi = LoRa.packetRssi();
    float receivedSnr = LoRa.packetSnr();
    while(LoRa.available()){
      String aux;
      aux = LoRa.readStringUntil('|');
      senderId = aux.toInt();
      aux = LoRa.readStringUntil('|');
      distance = aux.toFloat();
    }

    Heltec.display->clear();
    Heltec.display->setTextAlignment(TEXT_ALIGN_LEFT);
    Heltec.display->setFont(ArialMT_Plain_10);
    Heltec.display->drawString(0 , 0 , "RECEIVED PACKET");
    Heltec.display->drawString(0 , 10 , "RSSI: " + String(receivedRssi) +
      " SNR: " + String(receivedSnr) + " dB");
    Heltec.display->drawString(0 , 20 , "Sender ID: " + String(senderId))
      ;
    Heltec.display->drawString(0 , 30 , "Distance: " + String(distance));
    Heltec.display->display();

    Heltec.display->clear();
    if(WiFi.status() == WL_CONNECTED){
      WiFiClient client;

      if(!client.connect("192.168.0.105", 8080)){
        Heltec.display->drawString(0, 30, "Connection Failed!");
        Heltec.display->display();
        return;
      }

      String content = "{\"id\":\"" + String(senderId) + "\",\"distance\":\"" +
        String(distance) + "\"}";
      client.println("POST / HTTP/1.1");
      client.println("Host: 192.168.0.105");
      client.println("User-Agent: Arduino/1.0");
      client.println("Content-Length: "+String(content.length()));
      client.println();
      client.println(content);
      client.println();
    }
    else{
      Heltec.display->drawString(0, 10, "WiFi disconnected!");
    }
  }
}

```

```
        Heltec.display->display();  
    }  
}  
delay(1000);  
}
```

## APÊNDICE B – CÓDIGO DO SERVIÇO REMOTO.

O código a seguir descreve um serviço extremamente simples desenvolvido apenas para validar a arquitetura proposta. Este serviço, escrito em linguagem Go, recebe mensagens HTTP Rest e exibe as informações no terminal. A linguagem Go foi escolhida por mera conveniência do autor e não por apresentar vantagens em relação a outras linguagens similares.

Código 3 – Serviço HTTP Rest para receber as mensagens e exibir o conteúdo no terminal.

```
1 // HTTP REST API to receive the ID as an int32 and the Distance as a
   float32 float from the bins
2
3 package main
4
5 import (
6     "encoding/json"
7     "fmt"
8     "io/ioutil"
9     "log"
10    "net/http"
11 )
12
13 type Message struct {
14     ID          int32
15     Distance    float32
16     Message     string
17 }
18
19 func main() {
20     http.HandleFunc("/", handler)
21     log.Fatal(http.ListenAndServe(":8080", nil))
22 }
23
24 func handler(w http.ResponseWriter, r *http.Request) {
25     fmt.Println("method:", r.Method)
26     if r.Method == "POST" {
27         var m Message
28         body, err := ioutil.ReadAll(r.Body)
29         if err != nil {
30             fmt.Println(err)
31         }
32         err = json.Unmarshal(body, &m)
33         if err != nil {
34             fmt.Println(err)
```

```
35     }
36     fmt.Println("ID:", m.ID)
37     fmt.Println("Distance:", m.Distance)
38     fmt.Fprintf(w, "You sent the id %d and the Distance %f\n", m.ID, m.
        Distance)
39 }
40 }
```