



Caio Henrique Penning

**PROTÓTIPO DE SOLUÇÃO EM INTERNET DAS COISAS PARA AUTOMAÇÃO
DA CAPTAÇÃO DE DADOS EM HIDRÔMETROS ANALÓGICOS**

Horizontina-RS

2020

Caio Henrique Penning

**PROTÓTIPO DE SOLUÇÃO EM INTERNET DAS COISAS PARA AUTOMAÇÃO
DA CAPTAÇÃO DE DADOS EM HIDRÔMETROS ANALÓGICOS**

Projeto do Trabalho Final de Curso apresentado como requisito parcial para a o Trabalho Final de Curso na Engenharia de Controle e Automação da Faculdade Horizontina, sob orientação do professor Alexandre Kunkel da Costa, Me.

Horizontina-RS

2020

FAHOR - FACULDADE HORIZONTALINA
CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO

A Comissão Examinadora, abaixo assinada, aprova o trabalho final de
curso

“PROTÓTIPO DE SOLUÇÃO EM INTERNET DAS COISAS PARA
AUTOMAÇÃO DA CAPTAÇÃO DE DADOS EM HIDRÔMETROS
ANALÓGICOS ”

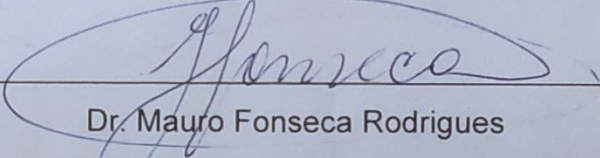
Elaborada por:

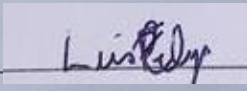
Caio Henrique Penning

Como requisito parcial para a obtenção do grau de Bacharel em
Engenharia de Controle e Automação

Aprovado em: 11/12/2020
Pela Comissão Examinadora

Me. Alexandre Kunkel da Costa
Presidente da Comissão Examinadora - Orientador


Dr. Mauro Fonseca Rodrigues
FAHOR – Faculdade Horizontalina


Dr. Luís Felipe Bianchi Carbonera

SATC - Associação Beneficente da Indústria Carbonífera de Santa Catarina

Horizontalina – RS

2020

Aos meus pais, por abdicarem de tanto para que eu pudesse continuar em busca da graduação, além de toda confiança colocada sobre mim. À minha família e amigos, por sua capacidade de me apoiar quando necessário para que eu me mantivesse firme no caminho.

Agradeço aos professores que me guiaram durante a trajetória da graduação. Aos amigos da equipe Sinuelo que possibilitaram tamanho crescimento técnico. Em especial ao meu orientado, Me. Alexandre Kunkel pelo suporte e conselhos sobre como continuar e para onde seguir, e a minha tia Sandra Mariana pelas correções e dicas para escrever o presente projeto.

“Você tem que ter uma atitude positiva e tirar o melhor da situação na qual se encontra.”.

(Stephen Hawking)

RESUMO

Internet das coisas e outros assuntos como indústria 4.0 estão cada vez mais em alta, o que auxilia o surgimento de novas tecnologias em âmbitos empresariais e pesquisas acadêmicas. Tendo essa ideia em mente, observou-se a seguinte situação problema: uma empresa de saneamento precisa colocar mão de obra nas ruas para realizar a leitura de hidrômetros analógicos, e assim computar a conta da residência. Para melhorar a eficiência operacional desse processo se coloca como objetivo desenvolver um protótipo para captação de dados a distância usando a tecnologia LoRa. Como proposta de pesquisa, busca-se desenvolver um dispositivo para coleta de dados no hidrômetro e outro hardware que receba e aloque os mesmo em um banco de dados on-line. Os testes foram realizados em ambiente simulado e de formas distintas, o primeiro operou com vazão constante enquanto o segundo a vazão era variável, o período de validação foi de cinco dias para cada método. O primeiro método resultou em 3,5% de variação entre o valor do hidrômetro e o valor coletado no banco de dados, já no segundo método essa variância foi de 2,5%. Por fim, espera-se que esse estudo inicial possa ser continuado e aperfeiçoado a ponto de contrapor todas adversidades, chegando a aplicação piloto nas empresas que prestam serviço de saneamento, sobretudo por possibilitar a redução dos custos operacionais e facilitar a supervisão da rede em tempo real.

Palavras-chave: Internet of Things, Automação, Hidrômetros e LoRa.

LISTA DE FIGURAS

Figura 1 - Lógica de Internet das Coisas.....	20
Figura 2 - Funcionamento do sensor de efeito hall	21
Figura 3 - Sensor YF – S201	22
Figura 4 - Hidrômetro monojato.....	23
Figura 5 – Hidrômetro Unijato	23
Figura 6 - Conexões Raspberry PI 3	25
Figura 7 - Attiny85	26
Figura 8 - Camadas da Rede LoRa.....	26
Figura 9 - Faixa de frequência LoRa por Região.....	27
Figura 10 - Estrutura rede LoRaWAN	29
Figura 11 - Classes de dispositivos LoRa	30
Figura 12 - Módulo LoRa E32	31
Figura 13 - Fluxograma da metodologia.....	36
Figura 14 - Fluxograma do protótipo	37
Figura 15 - Fluxograma do método 1	38
Figura 16 - Fluxograma do método 2	39
Figura 17 - Attiny85 modelo Digispark	42
Figura 18 - Modelagem do nó	42
Figura 19 - Esquemático do Nó.....	43
Figura 20 - Modelo para PCB do Nó	43
Figura 21 - Modelagem do Receptor	44
Figura 22 - Invólucro do nó em CAD	45
Figura 23 - Protótipo do nó montado.....	46
Figura 24 - Protótipo do Receptor montado	47
Figura 25 - Montagem do sensor no hidrômetro	47
Figura 26 - Ambiente de testes do nó	48
Figura 27 - Fluxograma da programação do nó	49
Figura 28 - Fluxograma programação do receptor	50
Figura 29 - Visualização Google Cloud Plataform (GCP).....	52
Figura 30 - SSH do terminal.....	52
Figura 31 - Terminal de acesso ao phpMyAdmin	53
Figura 32 - Colunas da tabela no banco de dados.....	54

Figura 33 - Liberação da porta no GCP	55
Figura 34 - Erro de execução do código do receptor	56
Figura 35 - Gráfico das leituras do método 1	57
Figura 36 - Gráfico das leituras do método 2	58

LISTA DE TABELAS

Tabela 1 - Estudos e projetos para referência em LoRa e IoT	18
Tabela 2 - Comparativo entre modelos de Raspberry	24
Tabela 3 - Informações modulo E32	32
Tabela 4 - Padrão para coleta de dados	39
Tabela 5 - Componentes comprados	45
Tabela 6 - Padrão para coleta de dados completo	58
Tabela 7 – Componentes detalhados.....	66

LISTA DE ABREVIATURAS E/OU SIGLAS

ABS – Anti-Lock Braking System

BD – Banco de Dados

BW – Bandwidth

CCS – Chirp Spread Spectrum

CF – Carrier Frequency

CR – Coding Rate

EEPROM - Electrically Erasable Programmable Read-Only Memory

FEC – Forward Error Correction

GHz - GigaHertz

IoT – Internet of Things

ISM – Industrial Scientific Medical

KB – Kilobyte

L - Litro

LED - Light-Emitting Diode

LoRa – Long Range

LoRaWAN – Long Range Wide Area Network

LPWAN – Low Power Wide Area Network

MHz - MegaHertz

ODBC - Open Database Connectivity

PCB - Printed Circuit Board

PWM - Pulse Width Modulation

RAM - Random Access Memory

ROM - Read-Only Memory

RX - Receive

SEQUEL – Structured English Query Language

SF – Spreading Factor

SGBD - Sistema de Gerenciamento de Banco de Dados

SSH - Secure Shell

SQL - Structured Query Language

TX - Transmit

V - Volts

Wi-Fi – Wireless Fidelity

SUMÁRIO

1 INTRODUÇÃO	13
1.1 TEMA	14
1.2 DELIMITAÇÃO DO TEMA	14
1.3 PROBLEMA DE PESQUISA	14
1.4 HIPÓTESES	15
1.5 JUSTIFICATIVA	15
1.6 OBJETIVOS	16
1.6.1 Objetivo Geral	16
1.6.2 Objetivos Específicos	17
2 REVISÃO DA LITERATURA	18
2.1 PROJETOS SEMELHANTES PARA COLETA E TRANSMISSÃO DE DADOS ..	18
2.2 IOT	19
2.3 SENSOR DE TAXA DE VAZÃO	20
2.4 HIDRÔMETRO MULTIJATO MAGNÉTICO	22
2.5 CONTROLADORES	23
2.5.1 Raspberry PI B	23
2.5.2 Microcontrolador ATTINY85	25
2.6 REDE LORA	26
2.6.1 Camada Física LoRa	27
2.6.2 Protocolo LoRaWAN	29
2.6.3 Modulo LoRa E32	31
2.7 BANCO DE DADOS	32
2.7.1 SQL	33
2.7.2 SGBD	33
3 METODOLOGIA	34
3.1 PLANEJAMENTO DO PROJETO	35
3.1.1 Pesquisa teórica	36
3.1.2 Prototipagem	37
3.1.3 Validações	38
3.2 MATERIAIS E EQUIPAMENTOS	40
4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS	41
4.1 MODELAGENS	41
4.1.1 Modelagem do nó	41
4.1.2 Modelagem do receptor	44
4.1.3 Modelagem do invólucro em 3D	44
4.2 MATERIAIS COMPRADOS	45
4.3 MONTAGEM DO PROTÓTIPO	46
4.3.1 Montagem do nó	46
4.3.2 Montagem do Receptor	46
4.3.3 Montagem do sensor de vazão	47
4.3.4 Montagem do ambiente de teste	48
4.4 PROGRAMAÇÃO	49
4.4.1 Programação do attiny	49
4.4.2 Programação do Raspberry Pi	50
4.5 CONFIGURAÇÃO DO BANCO DE DADOS	51
4.5.1 GCP	51
4.5.2 Instalação do Banco de Dados	52
4.5.3 Configuração do Banco de Dados	53

4.5.4 Liberação da Porta 3306	54
4.6 TESTES DO PROTÓTIPO	55
4.6.1 Ajustes iniciais	55
4.6.2 Resultados do Testes	56
CONCLUSÃO	60
SUGESTÃO DE MELHORIA E TRABALHOS FUTUROS	61
REFERÊNCIAS.....	62
APÊNDICE A – CIRCUITO ELÉTRICO DO PROJETO	65
APÊNDICE B – DESENHO DETALHADO DO INVÓLUCRO	66
APÊNDICE C – CÓDIGO DO NÓ	69
APÊNDICE D – CÓDIGO DO RECEPTOR.....	71

1 INTRODUÇÃO

O avanço tecnológico apresenta inúmeras formas para facilitar e até mesmo reduzir alguns empregos, que durante muito tempo, foram realizados pelo ser humano, por exemplo trabalhos manuais desenvolvidos pelos caixas de bancos e os técnicos de laboratórios fotográficos. Nessa perspectiva de evolução, o projeto apresentado aqui tem como enfoque um sistema para facilitar o gerenciamento da rede hidráulica de cidades e reduzir o trabalho manual na captação de dados para geração das faturas, do mesmo modo aumentar as funções de níveis técnico e superior.

Seguindo nessa linha de raciocínio, o intuito deste trabalho é desenvolver um protótipo eletroeletrônico funcional para captar os dados que referentes ao consumo hídrico de uma residência e enviá-los, com auxílio da tecnologia de rede de longas distâncias LoRa (*long range*), para uma central. Esta central, que poderia estar inserida dentro de um Centro de Controle Operacional/Comercial, é responsável pela interação entre a tecnologia LoRa e a rede de internet. Este protótipo é conectado a um servidor, cujo objetivo é realizar o tratamento e endereçamento do dado. Por fim, por meio dessa medição remota é possível gerar a fatura de um consumidor, seja residencial, comercial, industrial ou público. Além disso, com essas informações podem-se obter picos de consumo e gerenciar outras informações de valia para as companhias de saneamento.

Para o desenvolvimento do protótipo, o foco dá-se em itens de mercado, tanto sensores como tecnologias embarcados, que tragam níveis de robustez e qualidade, possibilitando a criação de um sistema confiável, e possivelmente mais acessível. Além do pressuposto, coloca-se que diante de uma possível implementação do sistema a redução dos gastos com vínculos empregatícios será de enorme valia para empresas concessionárias de saneamento.

Em questões tecnológicas supõe-se que o uso da rede LoRa junto com seu protocolo LoRaWAN pode trazer maior segurança para o sistema, pois é uma tecnologia razoavelmente nova, que necessita de *software* e *hardware* específicos para poder acessá-la. Como a aplicação final do projeto é voltado às empresas. Por isso, faz-se importante salientar que o sistema deverá ter uma rede segura e a prova de adulterações.

Durante o andamento deste projeto foi apresentado os aspectos de pesquisa e a escolha de componentes chave, custos dos materiais, métodos usados para desenvolvimento e manufatura do protótipo, aplicação do modelo em testes de funcionamento e precisão, além dos resultados obtidos, englobando os prós e os contras encontrados.

1.1 TEMA

Protótipo de sistema eletroeletrônico para captação de dados de hidrômetros analógicos, com foco em gerenciamento de consumo, e transmissão de dados através de rede LoRa.

1.2 DELIMITAÇÃO DO TEMA

Este estudo delimita-se por matérias pré-estabelecidos, com enfoque na funcionalidade do conjunto como protótipo, para enviar o dado referente ao consumo de água, de um hidrômetro da marca LAO, através da tecnologia LoRa. O dado terá como única finalidade aqui, aferir precisão e exatidão do sistema, visto que o protótipo não atingirá perfil de produto finalizado e não será testado sobre circunstâncias reais.

1.3 PROBLEMA DE PESQUISA

Caracteriza-se o problema da seguinte forma, um funcionário de uma empresa fornecedora da rede hídrica deve se dirigir a todas as casas de um município "X" e coletar todos os valores apresentados em seus hidrômetros. Após isso, ele deve levar esses dados para uma central em que será computado e irá gerar a fatura do consumo de água do referido mês. Em seguida, um terceiro recebe essa relação e volta a cidade para fazer a entrega para os residentes, isso ocorrerá todos os meses do ano da mesma forma.

Sendo assim, são muitas pessoas envolvidas para a captação de um único dado e geração de uma fatura que ainda adota o meio físico. De modo crítico, existem muitas funções desnecessárias sendo exercidas e gastos atrelados ao serviço dentro das companhias de saneamento em decorrência disso. A implementação de tecnologia como meio para agilizar esse processo dispõe de hidrômetros digitais integrados a rede, os quais apresentam um custo elevado e necessitam de mão de obra especializada para a manutenção, uma vez que, a maioria das companhias

trocam esse dispositivo em média a cada 4 anos. Dessa forma, o dispêndio monetário ultrapassaria o valor do problema apresentada anteriormente.

Tendo em vista que são grandes os empecilhos para a melhoria de eficiência e redução de custo na coleta de dados de redes hídricas indaga-se da seguinte forma: seria possível desenvolver um sistema capaz de realizar a leitura, de forma precisa, em um hidrômetro analógico e transmiti-lo através de algum tipo de rede sem fio utilizando tecnologias de mercado?

1.4 HIPÓTESES

Consideradas as preposições a serem confirmadas mediante testes posteriores, independentemente de serem reais ou falsas, as hipóteses são postas para alinhar o que se quer obter ao final do estudo e comparar com o real resultado obtido, para assim concluir se as ideias iniciais do projeto foram efetivas. Com referência ao tema e aos problemas já relatados, apresenta-se as seguintes hipóteses no desenvolvimento do protótipo:

Hipótese 1: o protótipo será funcional e apresentará boa precisão e confiabilidade na coleta dos dados do hidrômetro analógico, assim como não evidenciará perda de dados durante a transmissão via LoRa?

1.5 JUSTIFICATIVA

Muitas pessoas desempenham atividades profissionais repetitivas e prejudiciais ergonomicamente, que normalmente não desenvolvem habilidades e acabam em frustração. Com a ampliação de tecnologias desenvolveu-se meios para que o ser humano seja realocado e máquinas realizem suas atividades, de maneira completa ou colaborativa. A união da tecnologia ao meio de trabalho, apesar de gerar custos, tem um enorme valor agregado tanto em qualidade quanto em rendimento.

Sendo assim, ao analisar o funcionamento da companhia local de fornecimento de água observa-se o seguinte cenário: todos os meses um grupo de funcionários se dirige até as residências da cidade para coletar os valores informados nos hidrômetros, estes são levados até uma central da empresa, onde um sistema gerencia e transforma os dados em faturas, que por sua vez são encaminhadas de volta para os residentes daquela cidade. Dessa forma, a empresa pode economizar uma boa parte de seu orçamento se empregar tecnologia a essa atividade.

Como método para simplificar e otimizar tal processo a automação é uma possível solução. Os custos dessa aplicação, devido ao barateamento de tecnologias, seria acessível e aumentaria questões de eficiência. Sensores, controladores e módulos para redes de transmissão tornaram-se itens de mercado e podem ser encontrados por valores acessíveis, além de possuírem uma boa garantia de funcionamento de suas fabricantes.

O conjunto das tecnologias citadas acima, forma uma rede de IoT (*Internet of Things*) ou internet das coisas que de acordo com Santos et. al. (2016), primeiro viabiliza controlar remotamente dispositivos, e por segundo esse dispositivo pode ser acessado e controlado por outro semelhante, essa habilidade ocasiona inúmeras possibilidades em áreas acadêmicas ou industriais, todavia existem riscos técnicos e sociais que acompanham tais benefícios.

Assim, para justificar o barateamento do sistema final de forma mais clara, o sensor deve usar método evasivo a fim de reduzir danos e custos com manutenção. Da mesma forma a escolha da tecnologia LoRa se fundamenta por ter como benefícios o baixo custo para posicionamento, além de baixo consumo energético, enquanto redes como ZigBee e WiFi tem alcance muito menor e custos de implementação e consumo mais altos. Em comparação com outros métodos de maior alcance, como GSM e LTE, a complexidade para instalação e o custo direcionado ao usuário final não se encaixam com a proposta.

Nas redes de transmissão LoRa uma *network* é capaz de controlar “n” nodes espalhados dentro de sua área, apesar do tamanho dos pacotes serem limitados ainda assim permite grande número de aplicações. Um protótipo inicial será elaborado ao longo desse projeto com intuito de eliminar a necessidade do funcionário da companhia de saneamento e possibilitar ao mesmo desenvolvimento técnico ou operar em outra função enquanto a tecnologia de automação substitui o trabalho de coleta manual.

1.6 OBJETIVOS

1.6.1 Objetivo Geral

Como objetivo geral do presente projeto estipula-se criar um sistema eletroeletrônico dotado de comunicação LoRa, sendo este confiável e de fácil aplicação nas redes hídricas, com finalidade de centralizar dados e gerar medições

de consumo de água, além de informações passíveis de análise operacional. Além disso, aborda-se a questão da redução de custos para empresas desse ramo com funcionários, onde necessita de profissionais para realizar a coleta desses dados em todas as residências de um bairro, cidade ou até mesmo região.

1.6.2 Objetivos Específicos

Tendo em vista que para se alcançar o resultado esperado e atingir o objetivo geral outros estágios devem ser estabelecidos, para isso seguem os objetivos específicos que devem ser atingidos durante o processo de desenvolvimento e validação do presente projeto.

- I. Definir componentes a serem empregados no protótipo e levar em conta requisitos do projeto;
- II. Fabricar o protótipo de maneira robusta, apropriado para ser colocado em teste;
- III. Desenvolver o código a ser empregado no protótipo;
- IV. Realizar testes do protótipo em ambiente real ou simulado, a fim de aferir a sua exatidão.

2 REVISÃO DA LITERATURA

Neste capítulo serão abordadas questões técnicas de todas as tecnologias utilizados para desenvolvimento do protótipo. Será possível entender questões como o que é internet das coisas e a comunicação LoRa, conhecer os *hardwares* utilizados, desde o sensor de vazão, os controladores e módulos de transmissão LoRa, além de suas informações de fábrica. Se expõe ainda nesta partição sobre banco de dados, seu funcionamento e sua linguagem de programação.

2.1 PROJETOS SEMELHANTES PARA COLETA E TRANSMISSÃO DE DADOS

Durante a revisão literária serão colocados temas relevantes para confecção do projeto. Para entusiastas em conhecer mais sobre as tecnologias e aplicação destas em diversas áreas, principalmente na coleta e transmissão de dados, serão mencionados artigos e estudos, na Tabela 1, como complemento para referência.

Tabela 1 - Estudos e projetos para referência em LoRa e IoT

Título	Autores	Instituição de Ensino/Publicação	Descrição
IMPLEMENTAÇÃO DE TELEMETRIA EM UM VEÍCULO BAJA SAE ATRAVÉS DA TECNOLOGIA LORA	Ânderson Canssi	FAHOR - Faculdade de Horizontina	Tem por objetivo implementar uma telemetria de dados com a tecnologia LoRa em um veículo BAJA, voltado para parte de validação da recepção de dados a distancias variadas.
Uma aplicação da tecnologia LoRa em um ambiente hospitalar	Jonatas Magno Tavares Ribeiro	UFSC - Universidade Federal de Santa Catarina	Aplicação da transmissão via LoRa no monitoramento remoto da temperatura perante ambiente hospitalar, para melhorar índices de qualidade de serviço
Desenvolvimento de um Sistema de Aquisição de Dados em Rede Híbrida de Comunicação de uma Planta Hidráulica focado em LoRa	Emmanuel Leite de Medeiros	UFPB - Universidade Federal da Paraíba	Se trata da instrumentação de uma planta hidráulica e gerenciamento remoto através de rede LoRa. Ainda apresenta dados do comportamento da transmissão nesse caso.

A Review on Internet of Things (IoT)	M.U. Farooq, Muhammad Waseem, Sadia Mazhar, Anjum Khairi & Talha Kamal	Sir Syed University of Engineering & Technology, Pakistan	Retrata uma breve visão sobre IoT e se refere a partes técnica da ideia como arquitetura, novas tecnologias e aplicações.
Low Power Wide Area Network Analysis: Can LoRa Scale?	Orestis Georgiou & Usman Raza	IEEE WIRELESS COMMUNICATIONS LETTERS	Avaliação dos níveis de escalabilidade de um "gateway" LoRa em questão a número de dispositivos e análise de interferência com sequência de espalhamento iguais.

Fonte: Autor (2020).

2.2 IOT

Um relógio que monitora os batimentos cardíacos de um usuário com tendências a infarto é capaz de contatar a emergência se os valores saírem dos parâmetros. Uma assistente doméstica que as manhãs liga sua cafeteira 10 minutos antes do seu despertador e lembra que há probabilidade de chuva naquele dia e que deve levar um guarda-chuva. Ou ainda uma linha de produção dotada de parafusadeiras inteligentes, capazes de dar o torque preciso e de fazer correções se necessário, além disso essas podem contatar a manutenção em caso de mau funcionamento. Todos esses exemplos demonstram a ampla utilização para as *Internet of Things*(IoT) e como isso pode ser vantajoso em inúmeros aspectos.

De acordo com McEwen et. al. (2014), o que há de comum entre os exemplos citados acima e que definem a IoT é o uso da internet para enviar, receber ou comunicar dados. Outro ponto é que os dispositivos citados não são computadores, *tablets* ou *smartphones*, mas sim objetos, coisas (Figura 1). Então, a ideia de IoT sugere que, em vez de poucos dispositivos de computação poderosos (*notebook, tablet, smartphone*), você tenha um grande número de periféricos menos poderosos (relógio, espelho, cafeteira, tênis). Algo aproximado desse conceito utilizado anteriormente era a "computação ubíqua", que descrevia uma onipresença no dia a dia, isso reflete em muitas possibilidades de objetos dotados com tecnologias de computação.

Figura 1 - Lógica de Internet das Coisas



Fonte: Autor (2020).

Seguindo o mesmo conceito Morais et. al. (2018), complementa que a ideia de IoT se baseia na intersecção do mundo real com o mundo virtual (internet), resultando em constante interação entre humanos e objetos inteligentes. Com funcionalidades que variam entre rastreamento, avaliação inteligente, monitoramento, rastreamento e troca de informações a todo instante, esse conceito tendência serviços surpreendentes. O tema tem se tornado relevante para inúmeros segmentos como indústria, acadêmico e até mesmo governamental, pois essa tecnologia permite aprimorar questões sócias, profissionais e econômicas.

2.3 SENSOR DE TAXA DE VAZÃO

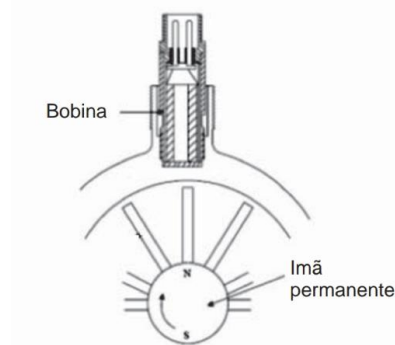
Segundo Soisson et. (2002), o uso de instrumentação em processos é indispensável para que exista um padrão de qualidade, tanto para suas etapas quanto para o produto final. Esses instrumentos são utilizados para controlar variáveis em processos e sistemas, variando sua precisão e *range* dependendo da necessidade, possibilitando assim atingir parâmetro determinado ou obter um produto com especificações mais precisas.

Diretamente falando sobre a instrumentação da vazão de líquidos ou gases se ressalta a importância, principalmente para aplicações industriais. Alguns dispositivos e equipamentos dependem taxas de vazão específicas, outros precisam ser monitorados para que não haja problema pelo excesso ou falta de um líquido e ainda há aquelas circunstâncias onde o consumível tem custo elevado, se fazendo necessário a instrumentação para fins de planejamento. (DUNN, 2013)

A vazão volumétrica quantifica a medição de substâncias, uma das técnicas utilizadas para isso é o medidor do tipo turbina com bobina de indutância (Figura 2), que consiste em uma turbina acoplada a um rotor e um conjunto de laminas que se

desloca ao ser submetido ao fluxo do fluido. A velocidade angular gerada a partir disso é diretamente proporcional à vazão volumétrica. Essa velocidade é captada por um sensor de efeito *hall* colocado do lado de fora do medidor, ou seja, ele opera através de um campo magnético que se move através da turbina. (FRANCHI, 2015)

Figura 2 - Funcionamento do sensor de efeito hall



Fonte: FRANCHI (2015).

O sensor de Fluxo de água modelo “YF-S201” (Figura 3), tem funcionamento do tipo turbina, como descrito acima, nesse modelo o campo magnético gera pulsos por modulação por largura de pulso (PWM) de aproximadamente 2,25mm, estes que podem ser enviados e interpretados por diversos modelos de microcontroladores. O dispositivo tem conexão de 1/2” e diâmetro interno de 0,78”, opera em tensão de 5 V até 24 V e corrente máxima de 15 mA(5 V). A faixa de fluxo permite até 30 L/min com pressão máxima de 2,0 Mpa, o mesmo tem uma referência de 450 pulsos por litro e apresenta exatidão de 10%.(EASYTRONICS)

Figura 3 - Sensor YF – S201



Fonte: Easytronics (2020).

2.4 HIDRÔMETRO MULTIJATO MAGNÉTICO

Um medidor de volume de água é destinado a medir continuamente, exibir e memorar o volume de água que passou pelo transdutor de medição. Além do transdutor, um medidor inclui um dispositivo calculador e um dispositivo indicador, o calculador pode ter ainda um dispositivo de correção ou ajuste. Estes dispositivos podem estar arranjados em um ou múltiplos invólucros de proteção. (Portaria nº 246 de 17 de outubro de 2000).

Baseado em Silva et.(xxxx), um hidrômetro taquimétrico ou de velocidade é acionado pela movimentação da água no seu interior, que aciona as hélices internas do dispositivo e transmite a informação obtida ao relógio, onde pode se visualizar a grandeza medida, esse modelo pode ser dividido em dois tipos, monojato ou multijato. Os do tipo monojato funciona por um único jato aplicado de forma tangencial na hélice interna para provocar a rotação de acordo com a Figura 4.

Figura 4 - Hidrômetro monojato



Fonte: AltoQi (2020).

Os hidrômetros taquimétricos multijatos têm como única diferença no seu funcionamento a aplicação de vários jatos tangenciais a hélice, tal como demonstrado na Figura 5 De forma simplificada, os medidores unijatos apresentam dimensões e alcance de vazão menores, além do preço de mercado mais acessível.

Figura 5 – Hidrômetro Unijato



Fonte: AltoQi (2020).

2.5 CONTROLADORES


2.5.1 Raspberry PI B

O *Raspberry Pi* tem a potência de um computador, mas a praticidade e o tamanho de um Arduino, além de seu custo muito acessível. Este é capaz de ser usado para aprender a programar, criar robôs e inúmeros outros projetos. O minicomputador permite tudo que já é esperado de um computador tradicional, porém nele você pode colocar seu próprio sistema operacional e conectar cabos e circuitos diretamente aos pinos da placa. Essa placa é feita para ensinar a programar em

linguagens como *Python* e *Scratch*, além da maioria das linguagens possuir um sistema operacional oficial. (BARNES, 2018)

Vários modelos do dispositivo foram lançados desde o precursor modelo B, cada um com melhorias e especificações diferentes, afim de atender aplicações mais específicas. A família do *Raspberry PI Zero*, por exemplo, é uma versão menor que abdica de algumas funções para ser compacto e consumir menos energia. O maior exemplo sem dúvida é o *Raspberry PI3* modelo B, sendo ele o mais popular entre todos modelos da linha devido a sua gama de aplicações em relação ao custo, veja o comparativo de alguns modelos na Tabela 2.

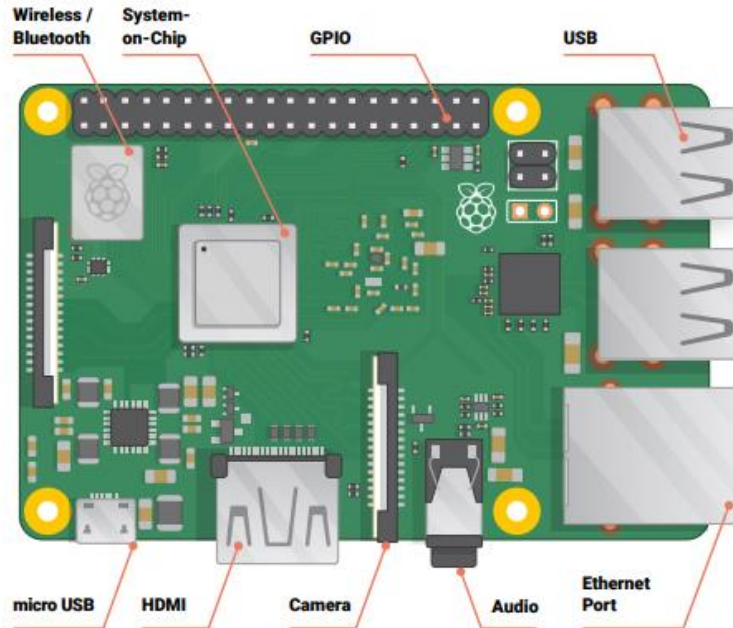
Tabela 2 - Comparativo entre modelos de Raspberry

	Raspberry Pi 3 Model B	Raspberry Pi 2 Model B	Model B+	Model A+	Model A	CMDK
Processor Chipset	Broadcom BCM2837 64Bit ARMv7 Quad Core Processor powered Single Board Computer running at 1250MHz	Broadcom BCM2836 32bit ARMv7 Quad Core Processor powered Single Board Computer running at 900MHz	Broadcom BCM2835 32bit ARMv6 SoC full HD multimedia applications processor	Broadcom BCM2835 32bit ARMv6 SoC full HD multimedia applications processor	Broadcom BCM2835 32bit ARMv6 SoC full HD multimedia applications processor	Broadcom BCM2835 32bit ARMv6 SoC full HD multimedia applications processor
GPU	Videocore IV	Videocore IV	Videocore IV	Videocore IV	Videocore IV	Videocore IV
Processor Speed	QUAD Core @1250 MHz	QUAD Core @900 MHz	Single Core @700 MHz	Single Core @700 MHz	Single Core @700 MHz	Single Core @700 MHz
RAM	1GB SDRAM @ 400 MHz	1GB SDRAM @ 400 MHz	512 MB SDRAM @ 400 MHz	256 MB SDRAM @ 400 MHz	256 MB SDRAM @ 400 MHz	512 MB SDRAM @ 400 MHz
Storage	MicroSD	MicroSD	MicroSD	MicroSD	SDCard	4GB eMMC
USB 2.0	4x USB Ports	4x USB Ports	4x USB Ports	1x USB Port	1x USB Port	1x USB Port
Power Draw / voltage	2.5A @ 5V	1.8A @ 5V	1.8A @ 5V	1.8A @ 5V	1.2A @ 5V	1.8A @ 5V
GPIO	40 pin	40 pin	40 pin	40 pin	26 pin	120 pin
Ethernet Port	Yes	Yes	Yes	No	No	No
Wi-Fi	Built in	No	No	No	No	No
Bluetooth LE	Built in	No	No	No	No	No

Fonte: Raspberry PI 3 Datasheet (2020).

No que diz respeito ao *Raspberry PI3* modelo B, ele possui entradas Ethernet, HDMI, Composite RCA, Audio 3.5mm, 4x USB 2.0, 40 pinos IO, 15 pinos para conexão com câmera serial MIPI e mais 15 para conexão com display, além de uma entrada para Micro SD. Seu *hardware* é dotado processador de 1.2GHz Quad-Core Arm Cortex-A53(64 bits), GPU VideoCore IV de 400MHz, módulos *Wireless* e *Bluetooth* internos além de 1GB de memória LPDDR2, veja Figura 6 para entender a disposição dos componentes em uma placa de 85x56mm. (Datasheet Rapsberry)

Figura 6 - Conexões Raspberry PI 3



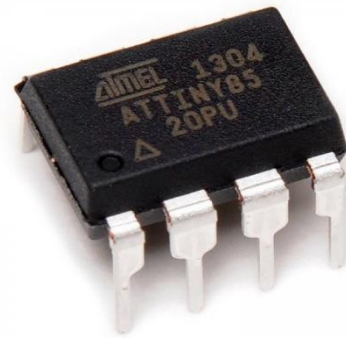
Fonte: Raspberry Pi Beginner's Guide (2020).

2.5.2 Microcontrolador ATTINY85

Segundo Delgado et. (2017), um microcontrolador é um sistema dentro de um circuito integrado que compreende ao processador, memórias (RAM e ROM) e periféricos, constituindo um computador completo e compacto. Diversos sistemas são controlados somente pelos microcontroladores, onde estes são ligados a interruptores, LEDs, interfaces de controle de motores e outros atuadores.

Mais especificamente o microcontrolador ATtiny85 (Figura 7), desenvolvido e fornecido pela empresa Atmel, ótimo para projetos pequenos e com baixo consumo de energia, dotado apenas de 8 pinos físicos, mas com uma boa variedade de funcionalidades por não ter só pinos digitais, mas também analógicos e suporte a PWM. Se tratando de hardware o pequeno ATtiny85 possui memória *Flash* de 8KB e EEPROM de 512B, além do *clock*, se podendo optar pelo interno de até 8MHz ou ainda a utilização de um externo de até 20 MHz. (ATMEL)

Figura 7 - Attiny85



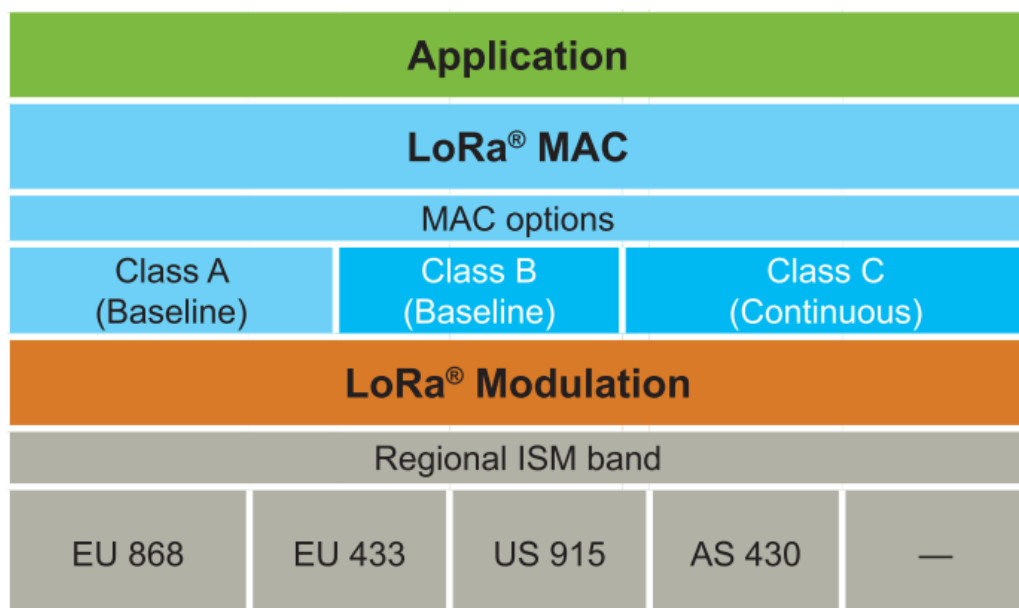
Fonte: Baú da eletrônica (2020)

2.6 REDE LORA

De acordo com Augustin et. al. (2016), LoRa compreende a “*Long Range*”, isso é um sistema de comunicação “*wireless*” que atinge grandes distâncias e que foi desenvolvido pela empresa LoRa Alliance para solucionar redes LPWAN. O sistema busca alto desempenho energético como algo importante e tem duas camadas distintas que compreendem ao seu funcionamento, uma física de modulação por rádio chamada “*Chirp Spread Spectrum*” (CSS) e outra baseada no protocolo MAC (LoRaWAN) (Figura 8).

Figura xx: Estrutura física LoRa

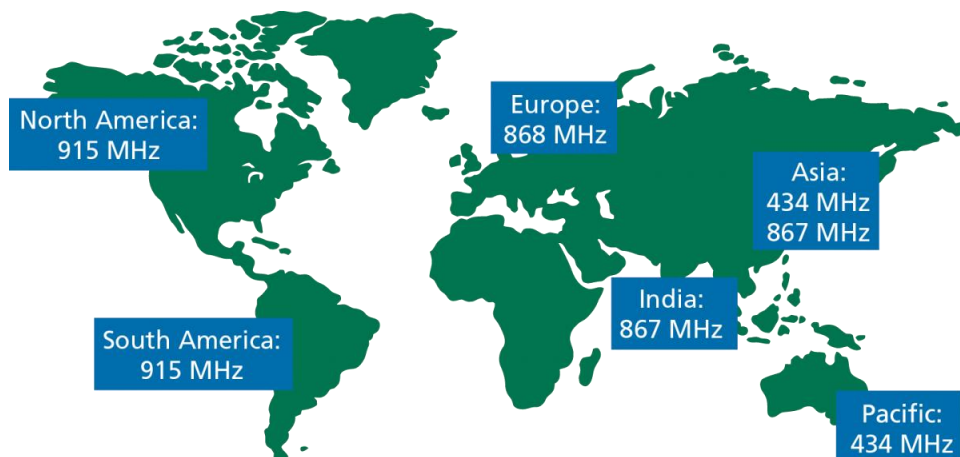
Figura 8 - Camadas da Rede LoRa



Fonte: LORA ALLIANCE (2015).

Na Figura 9, observar-se que os sistemas LoRa operam em frequências menores que giga-hertz, onde as bandas são pré-estabelecidas de acordo com o local do globo. É importante salientar que a frequência utilizada é a ISM (*Industrial Scientific Medical*), ou seja, não se necessita obter licenciamento ou pagar taxas para utilização.

Figura 9 - Faixa de frequência LoRa por Região



Fonte: LoRa Based Solutions (2020)

O LoRa ao ser utilizado como rede, forma uma LPWAN (*Low Power Wide Area Network*). A topologia adotada pelo LoRa pode ser de dois tipos, de estrela, onde existe uma central que recebe dados de vários nós, ou de ponto a ponto, normalmente quando se tem somente dois dispositivos trocando dados. Pelo fato de que não há endereçamento de rede em uma rede LoRa *default* ao colocar mais de dois dispositivos eles iram trocar informações entre si. (BERTOLETI 2019)

2.6.1 Camada Física LoRa

Camada física, segundo Bor et. al. (2016), se baseia nas especificações do CSS integrado com o “*Forward Error Correction*” (FEC). Esse tipo de sistema usa uma banda larga para lidar com interferências e compensação de frequência para o cristal. O LoRa pode decodificar transmissões abaixo do nível de ruído, e por isso pode se comunicar a grandes distâncias. As principais propriedades a serem citadas nessa camada são: alta robustez, múltiplas rotas, resistência ao efeito “*doppler*”, grande alcance e baixa potência.

Ao explicar a camada física do LoRa deve-se salientar os quatro parâmetros de modulação presentes: frequência de operação, fator de espalhamento, largura de banda e taxa de código.

2.6.1.1 Frequência de Operação

Frequência de Operação ou “*Carrier Frequency*” (CF) representa a frequência central de transmissão usada em uma banda. Pode ser programada entre 137 MHz até 1020 MHz, em espaços de 61 Hz.

2.6.1.2 Fator de Espalhamento

Também chamado por “*Spreading Factor*” (SF) o fator de espalhamento é a relação entre frequência proposta e a frequência do chip. Um alto fator é capaz de aumentar a relação sinal/ruído, afetando por consequência a sensibilidade e alcance, porém, o consumo de energia também é aumentado, devido a necessidade de mais tempo para transmissão dos pacotes. Comunicações de rádio com fator de espalhamento diferentes são ortogonais entre si, portanto é possível separar redes usando diferentes SF.

2.6.1.3 Largura de Banda

Largura de banda, também chamada de “*bandwidth*” (BW) é o alcance da frequência durante a transmissão da banda. Quanto maior a BW for maior será a taxa de transmissão, porém a sensibilidade diminuirá devido a maior quantidade de ruídos integrados ao sinal. Com um BW menor os status serão inversamente proporcionais, mas em soma disso o sistema requerer um cristal mais preciso.

2.6.1.4 Taxa de Código

Taxa de código ou “*Coding Rate*” (CR) é a taxa de FEC usada pelos modems LoRa, que oferecem proteção contra picos e interferências. Quanto maior a taxa de código, maior a proteção, em consequência o consumo de energia também aumenta. Frequências de rádio com diferentes CR e parâmetros anteriores iguais ainda conseguem manter comunicação.

2.6.2 Protocolo LoRaWAN

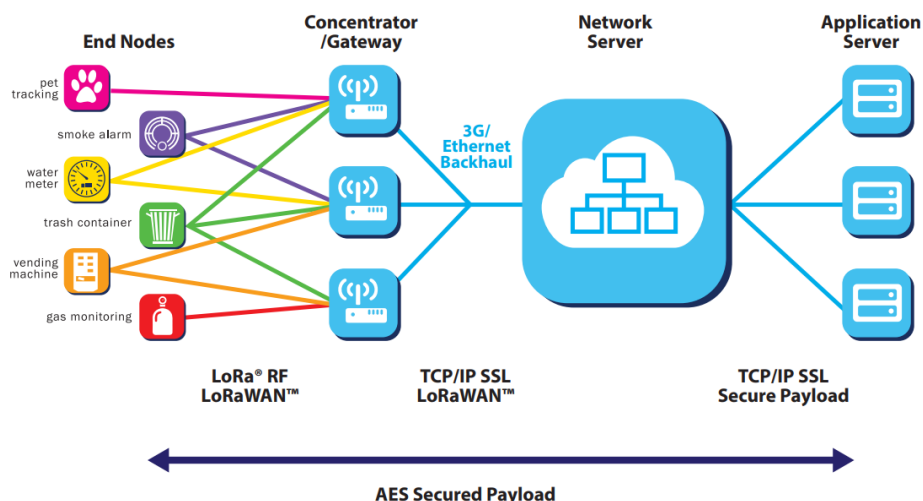
Para gerenciar redes e garantir organização, escalabilidade, dimensionamento e eficiência, tanto energética quanto de cobertura, em IoT a rede LoRa precisa seguir algum protocolo que dite o comportamento dos dados, por isso muitas plataformas tem tentado na LPWAN meios para unir o grande alcance das redes de telefonia e a alta eficiência de energia das redes RSSF. Vários padrões foram desenvolvidos, mas o de maior impacto no mercado é o LoRaWAN. (Teixeira et. al., 2017)

O protocolo LoRaWAN foi projetado com finalidade de conexão sem fio entre módulos em redes regionais, nacionais ou globais, visando principalmente suprir aspectos da IoT como comunicação direcional, segurança na transmissão, mobilidade e localização.

2.6.2.1 Arquitetura LoRaWAN

Como mostrado na Figura 10, em uma rede LoRaWAN os dispositivos não estão associados a um único *gateway*. Normalmente, os dados são enviados pelos *end nodes* e recebidos por vários *gateways* que enviam o pacote para o servidor através de algum *backhaul* (celular, *Ethernet*, *Wi-Fi* ou Satélite). O *network server* deve fazer a análise e gerenciamento de redundância nos dados, validações de segurança, reconhecer rotas mais rápidas e definir qual gateway permitirá isso e só então permitir que estas informações sejam usadas pelas aplicações finais. (LORA ALLIANCE, 2015)

Figura 10 - Estrutura rede LoRaWAN



Fonte: LORA ALLIANCE (2015)

2.6.2.2 Classes de Dispositivo

LoRaWAN é dividido em três classes de dispositivos para transmissão, sendo essas a Classe A, Classe B e Classe C, cada qual para diferentes aplicações e diferentes requisitos. As classes de dispositivo entrelaçam latência de comunicação de downlink em relação a vida útil de bateria, como mostrado na Figura 11. Em casos de controle ou de alguns atuadores a latência de comunicação do downlink é muito importante. (LORA ALLIANCE, 2015)

Figura 11 - Classes de dispositivos LoRa



Fonte: Adaptado de LORA ALLIANCE (2015).

Classe A (Comunicação bidirecional): Sensores e atuadores de classe A permitem comunicação Bidirecional onde cada transmissão de *uplink* é seguida por duas janelas curtas de recepção *downlink*. O slot de transmissão é programado baseado na necessidade do próprio dispositivo final com uma pequena variação baseada em base aleatória de tempo. Esse tipo de operação da classe A apresenta o consumo de energia mais baixo para aplicações que precisam de apenas transmissão *downlink* a partir do servidor pouco depois do dispositivo final enviar transmissão *uplink*. As comunicações de *downlink* do servidor em qualquer outro momento terão que esperar até o próximo *uplink* agendado.

Classe B (Comunicação bidirecional com recebimento agendado): Além das janelas de recebimento aleatórios dos dispositivos de classe A, os de classe B abrem

janelas de recebimento em momentos agendados. Para que o dispositivo final abra a janela de recepção no tempo programado, ele recebe um sinal sincronizado com o tempo do *gateway*. Isso permite que o servidor saiba quando o dispositivo final está “ouvindo”.

Classe C (Comunicação bidirecional com máximo de slots de recebimento): Dispositivos finais de classe C ficam continuamente com a janela de recebimento de dados aberto, somente é fechada quando o mesmo está transmitindo.

2.6.3 Módulo LoRa E32

O modelo E32-TTL-100, como visto na Figura 12, é um módulo *transiever* com da tecnologia LoRa “*spread spectrum*” baseado no SX1278 da SEMTECH. Com potencial de transmissão a longas distâncias o E32 traz vantagens como estabilidade de potência e melhor performance no quesito anti-interferências. Além disso, o módulo pode ser configurado pelas portas “M0” e “M1” nos modos normal, “*Wake up*”, “*Power Saving*” e “*Sleep*”.(ATMEL 2020)

Figura 12 - Módulo LoRa E32



Fonte: ATMEL (2020).

O E32 pode ser utilizado para inúmeras finalidades, seja como no presente projeto para leitura automática de medidores até automação de edifícios residenciais, sistemas de segurança, sistemas de irrigação remota, entre tantas outras. Em adicional esse modelo possui um FEC que torna os códigos mais eficientes, no caso de uma interferência súbita ele pode corrigir automaticamente o pacote afetado, mas

como já citado, quanto maior a taxa de código desse FEC maior será o consumo energético. As demais características podem ser visualizadas na Tabela 3.

Tabela 3 - Informações modulo E32

No.	Parameter item	Parameter details	Description
1	IC	SX1278	SEMTECH
2	Size	21* 36mm	Without SMA
3	Weight	6.7g	With SMA
4	Frequency Band	433MHz	Frequency range : 410~441MHz, Channel: 32
5	PCB	4-layer	Impedance-matching, lead-free
6	Connector	1 * 7 * 2.54mm	Plug-in
7	Supply voltage	2.3 ~ 5.5V DC	Note: the voltage higher than 5.5V is forbidden
8	Operation Range	3000m	Clear and open area, 20dBm, antenna gain: 5dBi , height: 2m , air data rate: 2.4kbps
9	Transmitting power	20dBm	4 optional level (20、17、14、10dBm)
10	Air data rate	2.4kbps	6 optional level (0.3、1.2、2.4、4.8、9.6、19.2kbps)
11	Standby current	3.0uA	M1=1, M0=1 (Mode 3)
12	Transmitting current	120mA@20dBm	The proposed power supply current is not less than 300mA
13	Receiving current	14mA	Mode 0 or 1
14	Communication interface	UART	8N1、8E1、8O1 , Eight kinds of UART baud Rate, from 1200 to 115200 bps (Default: 9600)
15	Driving mode	UART	Can be configured to push-pull/high pull, open-drain
16	Transmitting length	512 bytes buffer	58 bytes per package
17	Receiving length	512 bytes buffer	58 bytes per package
18	Address	65536	Easy for network, broadcast and fixed transmission
19	WOR	Available	Minimum average power consumption is about 30uA (applicable for battery powered applications)
20	RSSI	No support	Built-in intelligent processing
21	Sensitivity	-138dbm@0.3kbps	Sensitivity has nothing to with baud rate or delay time
22	Antenna type	SMA-K	50Ω impedance
23	Operating temperature	-40 ~ +85°C	Industrial-grade
24	Operating temperature	10% ~ 90%	No condensation
25	Storage temperature	-40 ~ +125°C	Industrial-grade

Fonte: Atmel (2020).

2.7 BANCO DE DADOS

Para Alves (2014), primeiro é interessante entender o conceito de informação e dado. Informação é um fato ou conhecimento do mundo real e dado é a

representação da informação. Um banco de dado (BD) é um aglomerado de dados organizados com objetivo definido e não uma matriz desestruturada sem fim estabelecido, por isso ao se construir um banco os dados devem ser arranjados de acordo com necessidade de manipulação e em certos casos de expansão.

2.7.1 SQL

SQL é uma linguagem de programação, considerada por ser a maior influência do sucesso de banco de dados em âmbito comercial. O nome SQL deriva de *Structure Query Language* ou “Linguagem Estruturada de Consulta”, que inicialmente era denominada SEQUEL (*Structured English QUERY Language*) pelos criadores da *IBM Research*, onde era visto como sistema experimental para conexão com bancos relacionados. (ELMASRI, 2005)

De acordo com DAMAS et. (2014), a linguagem SQL é específica para banco de dados e possui comandos que permitem consultar, definir e atualizar dados como Oracle e SQL Server. Se pode criar ainda visões (*views*) nos bancos, a fim de especificar questões de acesso, restringir os dados apresentados, garantir mais segurança e manter a integridade do banco de dados. Além de tudo o SQL é uma linguagem de estrutura simples, fácil de ser aprendido e implementado.

2.7.2 SGBD

Sistema gerenciador de banco de dados ou SGBD é um conjunto de *softwares* que possibilitam manter um banco de dados. Esse surgiu na década de 70 com intuito de simplificar interações com bancos de dados. Os primeiros sistemas tinham custo elevado e complexos para uso, requerendo até mesmo profissionais especializados para um SGBD específico. Na década de 80 surgiu por sua vez o SGBD relacional, e com o barateamento de *hardware* e *softwares* para essa aplicação o novo estilo de aplicação dominou o mercado, este que atualmente é usado na maioria dos desenvolvimentos em BD. (HEUSER, 2009)

Portanto, o SGBD é um sistema de *software* de propósito geral que simplifica aplicações de usuário, como definição, construção, manipulação e compartilhamento de um banco dados. A definição de refere as especificações dos tipos de dados, as estruturas e as restrições a serem seguidas naquele banco. Construção tem como base apropriar o armazenamento seguindo o controle do SGBD. (ELMASRI, 2005)

Seguindo Silberschatz et. al. (2020), a manipulação tem como funções, a pesquisa em banco para recuperar algum dado específico, atualizar o BD e gerar relatórios das atividades do mesmo. A aplicação chamada de compartilhamento permite múltiplos acessos, sendo por usuários ou softwares, ao banco de forma concorrente. Ainda falando sobre SGBD não se pode deixar de fora a proteção, que diz respeito ao sistema contra implicações externas e mal funcionamento em hardware ou software, e a manutenção, que devido a vida útil de um banco de dados poder chegar a muitos anos, o SGBD deve ter a capacidade de evolução para as necessidades que se alteram durante todo esse período, e assim manter o mesmo funcionando.

2.7.2.1 ODBC

ODBC se refere a *Open Database Connectivity* ou em português Conectividade a Banco de Dados Aberta que é um padrão de conexão SGBD que permite a conexão entre o cliente e o SGBD, desde que os dois possuam o *software* requerido instalado. Sendo assim, o cliente pode se conectar em vários SGBDs e enviar solicitações de consulta pelo ODBC API (*application programming interface*), estas que são processadas no servidor e enviadas de volta para o cliente, que as usa de acordo com a necessidade. (ELMASRI, 2005)

2.7.2.2 MySQL

O MySQL é um sistema de gerenciamento de banco de dados SQL *open source* muito popular atualmente, desenvolvido e distribuído pela MySQL AB. Sendo assim, o MySQL é uma plataforma SGBD, ele permite a interação cliente/servidor, onde o servidor SQL multitarefa suporta diferentes tipos de acesso, diversos softwares, bibliotecas, ferramentas para gerenciamento e interfaces de programação. O sistema apresenta como oferta velocidade, confiabilidade e uso simplificado permitindo alta adaptabilidade para conexão com bancos na Internet. (ORACLE)

3 METODOLOGIA

A metodologia aplicada foi a de pesquisa explicativa, esta tem como enfoque descobrir e identificar como é o funcionamento de um objeto posto a estudo, é o método que se alinha mais com a realidade e a prática porque explica o funcionamento

e a razão dos elementos. Assim este possibilita bons resultados científicos a troco de maior complexidade, da forma que cometer erros é um risco mais propenso e que pode afetar o resultado obtido. (GIL, 2017)

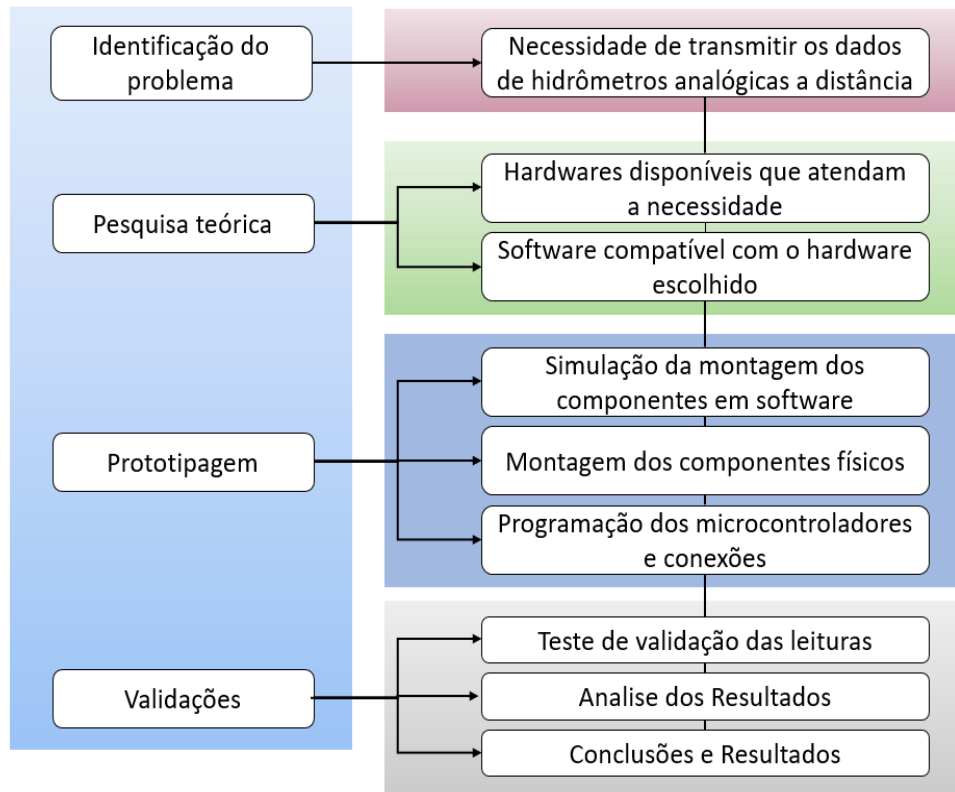
O delineamento adotado é conhecido como pesquisa experimental e de acordo com Gil (2017), consiste na escolha de um objeto com as respectivas variáveis que o afetam, então realizar uma experimentação controlada com finalidade de observar os efeitos produzidos a fim de gerar resultados. Esta pesquisa dispõe de propriedades que a caracterizam e devem ser seguidas durante o seu desenvolvimento, sendo elas manipulação, controle e distribuição aleatória.

De forma resumida, a manipulação é retratada pela necessidade de o pesquisador alterar alguma variável no objeto estudado, o controle diz que o pesquisador deve ter um ou mais controles durante a fase de experimentação e a distribuição aleatória indica que os elementos escolhidos tanto para a fase de manipulação quanto para de controle. Também devem ser escolhidos de forma aleatória. Casos que não apresentam a aleatoriedade podem ser indicados como pré-experimentais as quais trazem o peso da falta de objetos desconhecidos, nesse caso o importante é apresentar e esclarecer os resultados obtidos.

3.1 PLANEJAMENTO DO PROJETO

Para elaboração do presente projeto, o escopo foi definido de acordo com a Figura 13, onde se identificou o problema a ser trabalhado, o qual leva a necessidade de pesquisa teórica para concepção de ideias para uma possível solução, a prototipagem do sistema e validação do conjunto final em operação. Com exceção do problema, que já foi explicado anteriormente, os demais pontos serão retratados a seguir com intuito de esclarecer os métodos adotados em cada fase.

Figura 13 - Fluxograma da metodologia



Fonte: Autor (2020).

3.1.1 Pesquisa teórica

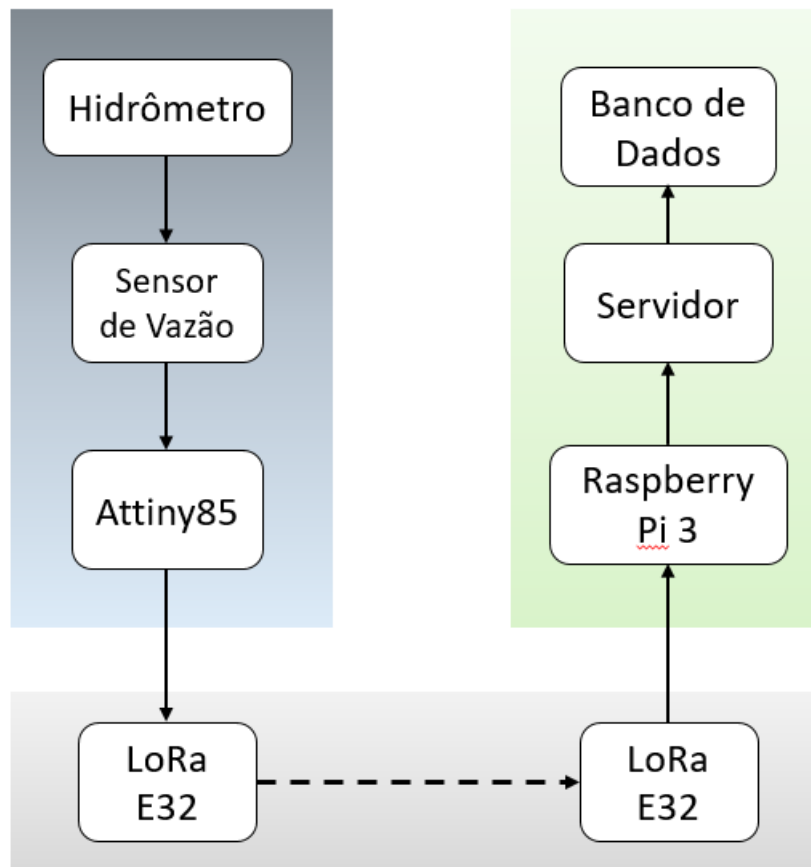
Ao ter o problema esclarecido, foi necessário a pesquisa de tecnologias capazes de atender as necessidades e limitações de um hidrômetro analógico. Em questão de *hardware* o estudo teve enfoque em componentes de tamanho pequeno para o circuito que corresponde ao nó, além disso, o custo não deveria ser elevado, deveria ser facilmente encontrado no mercado e capazes de interagir com outros tipos de tecnologias, já para o *gateway* o foco era encontrar um controlador potente e capaz de conexão direta com servidores.

Para a pesquisa de *softwares* o foco foi em tecnologias usuais de acordo com cada *hardware*, onde e quando houver mais de uma opção, optou-se pela com maior material de referência disponível. Para questões de manufatura do protótipo foi pesquisado métodos simples para montagem dos circuitos e que garantissem resistência, confiabilidade e proteção para o mesmo.

3.1.2 Prototipagem

O prototipo foi desenvolvido para seguir a ideia do fluxograma na Figura 14, onde se idealizou a coleta de dados no hidrometro através de um sensor de vazão. Já, a leitura e tratamento inicial do dado é realizado pelo microcontrolador Attiny85, este que também controla o modulo LoRa E32 e possibilita a conexão remota com o outro módulo ligado ao Raspberry Pi 3. Por fim, este microprocessador é responsável pela decodificação dos dados, conexão e envio das informações para o servidor.

Figura 14 - Fluxograma do protótipo



Fonte: Autor (2020).

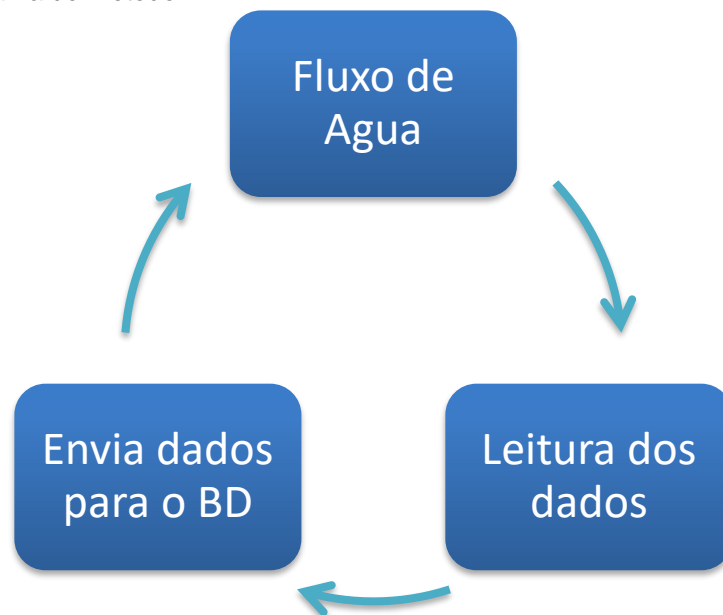
Para montagem do *node* foi optado pela criação de uma placa de circuito impressa para facilitar a montagem dos componentes, a interligação da parte negativa e minimizar espaço físico a ser utilizado. O mesmo possui um invólucro feito de material plástico para proteção do sistema contra impactos, poeira e água. O *gateway* foi prototipado de forma mais simples, pois não foi colocado fora de ambiente protegido, por isso foi utilizados alguns *jumpers* e uma *protoboard*.

3.1.3 Validações

Para experimentação e validação da precisão foi criado um ambiente simulado com hidrômetro e instrumentação acoplados em operação. Foram feitas baterias de testes, usando dois métodos, de forma que ambos os dados foram aferidos diariamente para validar a precisão, assim também com o registro de correções e ajustes a serem considerados nos resultados.

O primeiro método de validação teve como objetivo ver como o protótipo reage a trabalho intenso, como na Figura 15 o sistema ficou com corrente de água 24 horas por dia, durante cinco dias, os dados no banco foram aferidos e comparados com a grandeza apresentada no leitor analógico do hidrômetro, o sistema só foi parado no momento da validação diária e se houve necessidade de ajuste no código ou conserto no sistema.

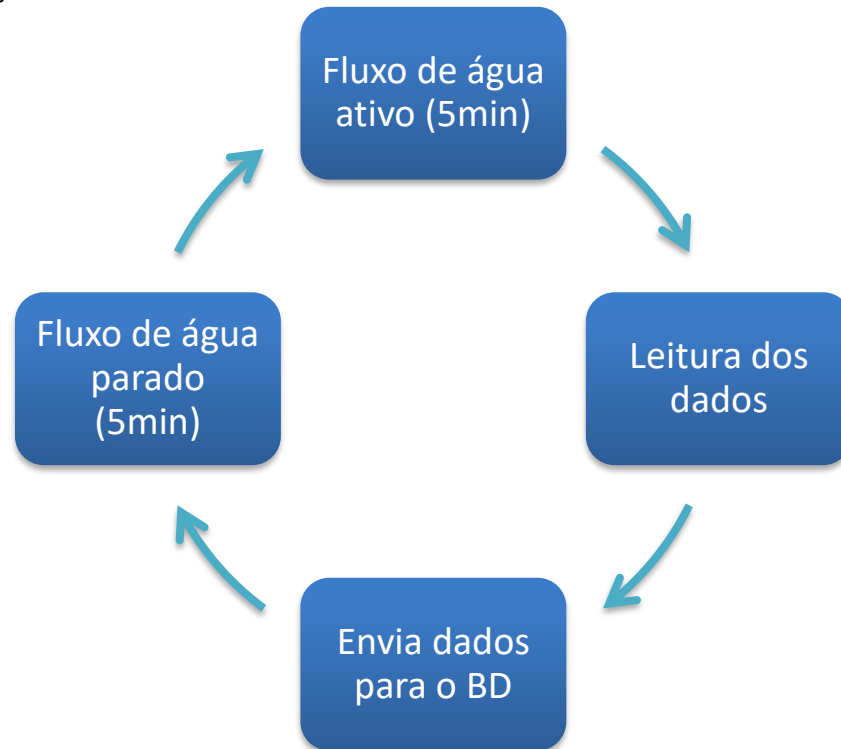
Figura 15 - Fluxograma do método 1



Fonte: Autor (2020).

O segundo método era muito similar ao primeiro, mas este com enfoque em validar o funcionamento do dispositivo aplicado a trabalho variável. Como demonstrado na Figura 16, o sistema hidráulico fica atuante pelo período de 5 minutos e desativado pelo mesmo tempo, esse ciclo se repete pelo mesmo intervalo que o método anterior e os dados também são aferidos uma vez ao dia, porém somente quando a vazão está parada.

Figura 16 - Fluxograma do método 2



Fonte: Autor (2020).

Para fins de padronização os dados foram anotados em uma matriz apresentada na Tabela 4. Os dois métodos foram aplicados pelo mesmo período de 5 dias, mas com horário variado, os dados de leitura a serem completados estão na grandeza de litros. Onde a referência para análise dos dados é leitura do hidrômetro, que se considera calibrado, por nunca ter sido utilizado.

Tabela 4 - Padrão para coleta de dados

Dia	Método 1 - Vazão continua			Método 2 - Vazão Oscilante		
	Hora	Leitura Hidrômetro	Leitura Instrumentação	Hora	Leitura Hidrômetro	Leitura Instrumentação
1						
2						
3						
4						
5						

Fonte: Autor (2020).

3.2 MATERIAIS E EQUIPAMENTOS

Nesta seção são listados os principais materiais e equipamentos utilizados no desenvolvimento do protótipo e durante a fase de testes deste projeto científico.

- a) Módulos LoRa – E32;
- b) Microcontrolador Attyni85;
- c) Sensor de Vazão – YF-S201;
- d) Fontes de alimentação 5V;
- e) Softwares de programação em linguagem C e Python;
- f) Software de Eletroeletrônica e Simulação;
- g) Placa *Raspberry Pi 3*;
- h) Material de laboratório;
- i) Banco de dados SQL;

4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Nesta seção são apresentados os resultados do planejamento e dos métodos discutidos anteriormente, expondo aspectos de desenvolvimento e validação do protótipo até os resultados obtidos através da submissão a testes.

4.1 MODELAGENS

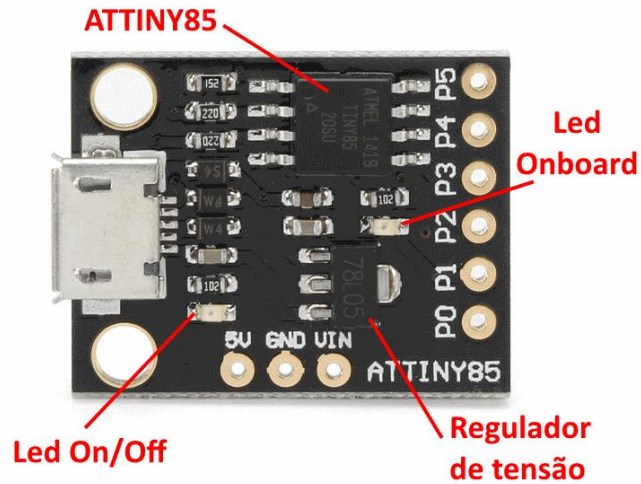
Essa seção aborda todos aspectos do projeto que necessitaram de desenho ou montagem em ambiente virtual para facilitar o entendimento e a concepção do protótipo.

Foram desenvolvidos dois circuitos distintos para o projeto, por ser um sistema de transmissão de dados sem fio, mostra-se necessário um dispositivo que coleta os dados e funcione como um transmissor e um outro capaz de receber esse sinal, o receptor. Para facilitar o entendimento chamaremos um circuito de “Nó”, que fara a coleta de dados, e o outro de “Receptor” que será responsável pelo recebimento e interação com o banco de dados, o esquema elétrico dos circuitos pode ser encontrado no APÊNDICE A.

4.1.1 Modelagem do nó

Primeiramente abordaremos o nó, que nesse projeto é o conjunto com maior quantidade de componentes eletrônicos conectados. O controlador do nó é o Attiny85, contudo a versão utilizada é conhecida como “*Digispark*”, apresentada na Figura 17, este modelo foi escolhido por facilitar a prototipagem, possuir regulador de tensão e aceitar programação através IDE do Arduino. Apesar de serem variantes diferentes da mesma placa na simulação e no protótipo as portas e conexões são iguais, a única variância é o método de conexão para programação, onde um tem o conector macho do USB e o outro necessita de um cabo.

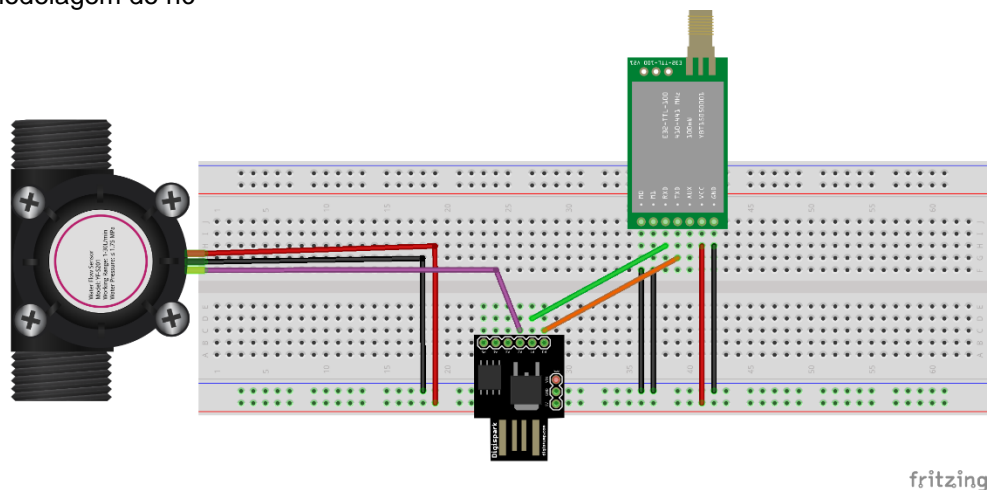
Figura 17 - Attiny85 modelo Digispark



Fonte: Arduino e Cia (2020).

Para modelagem inicial foi utilizado o software “Fritzing”, que possui o *digispark*, o módulo LoRa E32 e o sensor de vazão YF-S201 como bibliotecas adicionais facilmente encontradas na internet. As ligações são apresentadas na Figura 18, onde o pino 0 e 1 do *Digispark* são ligados ao TX e RX do módulo LoRa para comunicação serial. O pino 2 do *Digispark* foi escolhido como entrada do sensor de vazão por permitir a utilização da função “Interrupt” para contagem de pulsos.

Figura 18 - Modelagem do nó

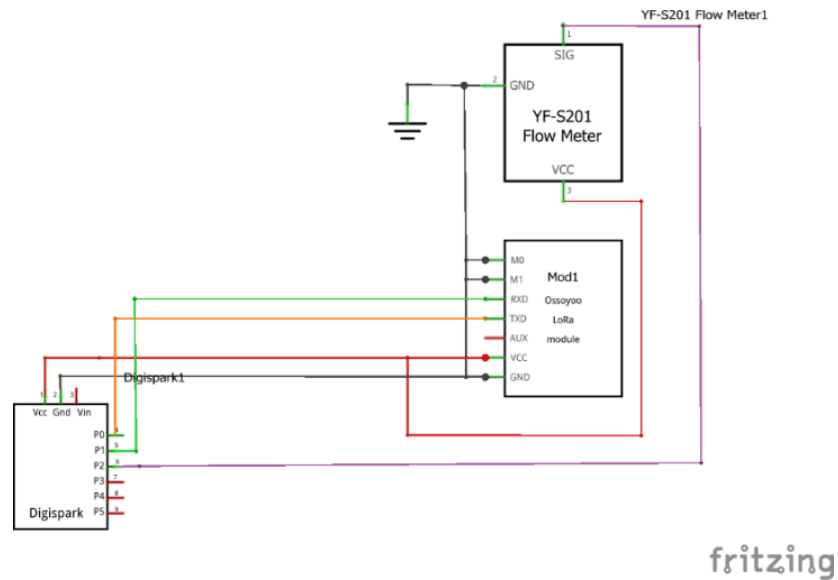


Fonte: Autor 2020.

As conexões M0 e M1 do LoRa E32 foram conectados diretamente ao negativo, para que o sistema opere em seu modo normal, similar a classe C, isso porque os testes foram de envio constante e não se tinha foco em consumo energético. A

alimentação é feita através de uma fonte externa de 5V, a mesma tensão é utilizada em todos componentes do sistema e o negativo é comum.

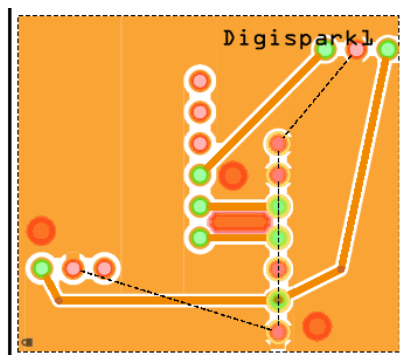
Figura 19 - Esquemático do Nó



Fonte: Autor (2020).

Ainda utilizando o software Fritzing, foi desenhado o modelo esquemático do nó como se pode ver na Figura 19, onde fica mais claro as ligações e as relações, por exemplo, dos números das portas do attiny85 com a nomenclatura adotada pela placa *digispark* e as demais ligações no modulo LoRa. Através do esquemático se pode gerar o modelo para impressão da PCB como na Figura 20, com finalidade de compactar o circuito. Esta que possui apenas uma face com cobre e o negativo é o comum na base.

Figura 20 - Modelo para PCB do Nó

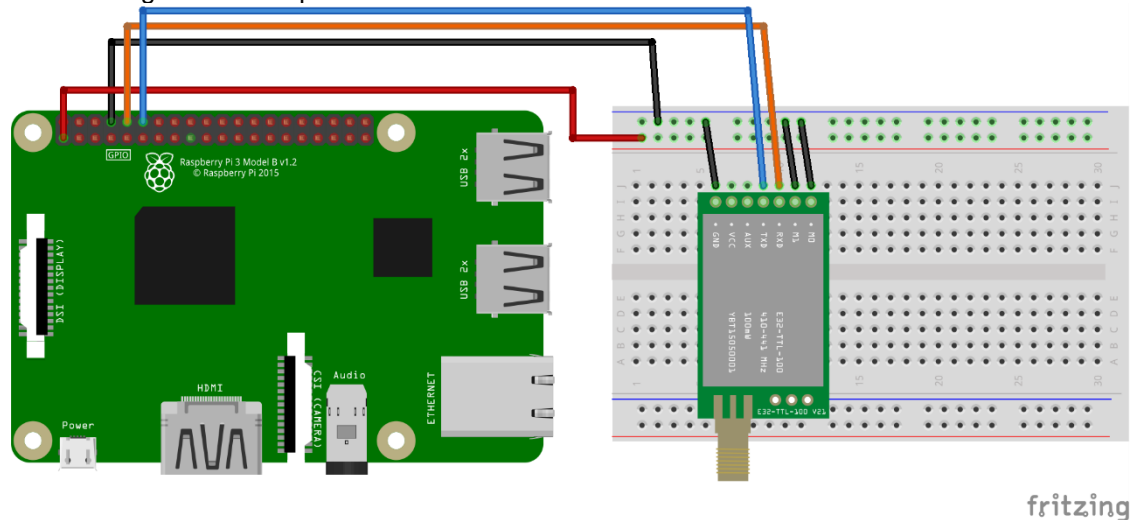


Fonte: Autor (2020).

4.1.2 Modelagem do receptor

A modelagem do receptor compreende as ligações do *Raspberry Pi 3 B* e do módulo LoRa E32 apresentados na Figura 21. É importante ressaltar que as disposições das portas de todos modelos de *Raspberrys Pi* são iguais, portanto para simulação qualquer uma poderia ter sido escolhida.

Figura 21 - Modelagem do Receptor



Fonte: Autor 2020.

As conexões TX e RX do Raspberry foram conectados ao RX e TX do LoRa E32 respectivamente, para que assim haja comunicação entre os dispositivos e recebimento dos dados. A alimentação para o módulo vem do *Raspberry*, sendo 3.3V positivo e o GND, adianta-se que alimentação externa será por meio de uma fonte 5V conectada a entrada “power” do *Raspberry Pi*. Da mesma forma que o nó, o M0 e M1 foram ligados ao negativo para recepção constante.

4.1.3 Modelagem do invólucro em 3D

Para proteção do nó foi desenvolvido um invólucro no *software* de CAD “Solid Works” apresentado na Figura 22. O conceito utilizado é similar a uma caixa de passagem de fiação elétrica, porem com as aberturas específicas para os tipos de conexões necessárias. A tampa tem uma espécie de encaixe para facilitar a vedação e as partes que ficarem abertas podem ser facilmente fechadas com silicone ou similar. Os desenhos detalhados das partes aqui citadas encontram-se no APÊNDICE B.

Figura 22 - Invólucro do nó em CAD



Fonte: Autor (2020).

4.2 MATERIAIS COMPRADOS

Após desenvolvimento em *software* do circuito, foi realizado a compra dos materiais para execução do projeto. Na Tabela 5 é apresentado o que foi adquirido e o seus custos respectivamente. Todos componentes são facilmente encontrados em lojas de eletrônica e principalmente e-commerce.

Tabela 5 - Componentes comprados

Componente	Qtd.	Custo Unitário	Custo total
Digispark	1	R\$ 18,90	R\$ 18,90
LoRa E32	2	R\$ 90,00	R\$ 180,00
Raspberry PI 3	1	R\$ 369,00	R\$ 369,00
YF-S201	1	R\$ 54,91	R\$ 54,91
Placa de cobre 5x5	1	R\$ 2,76	R\$ 2,76
Total			R\$ 625,57

Fonte: Autor (2020).

4.3 MONTAGEM DO PROTÓTIPO

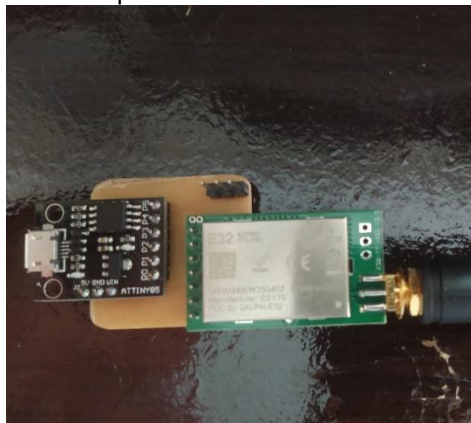
Para montagem do protótipo foram usados métodos e informações já descritas anteriormente. Nessa seção será descrito como foi feita a manufatura e montagens dos conjuntos.

4.3.1 Montagem do nó

Primeiramente, para montagem do nó, foi impresso a PCB apresentada na seção 4.1.1 em placa de cobre face simples por método caseiro. Para isso o desenho foi impresso em papel fotográfico e aplicado sobre a placa de cobre com ferro de passar roupa em temperatura alta. Após soltar a tinta na placa, ela foi submetida a percloroeto de ferro para corrosão das trilhas por alguns minutos, a placa foi lavada e se realizou as furações para inserção dos componentes. Por questão de homologação se verificou a continuidade das trilhas e do aterramento com multímetro.

O modulo Lora E32 juntamente com o *digispark* foram soldados diretamente a placa, para o sensor de vazão foi alocado pinos para conexão. De acordo com a figura 23(A) pode se ver a parte superior do conjunto enquanto na figura 23(B) é apresentado a parte inferior do mesmo.

Figura 23 - Protótipo do nó montado



(A)



(B)

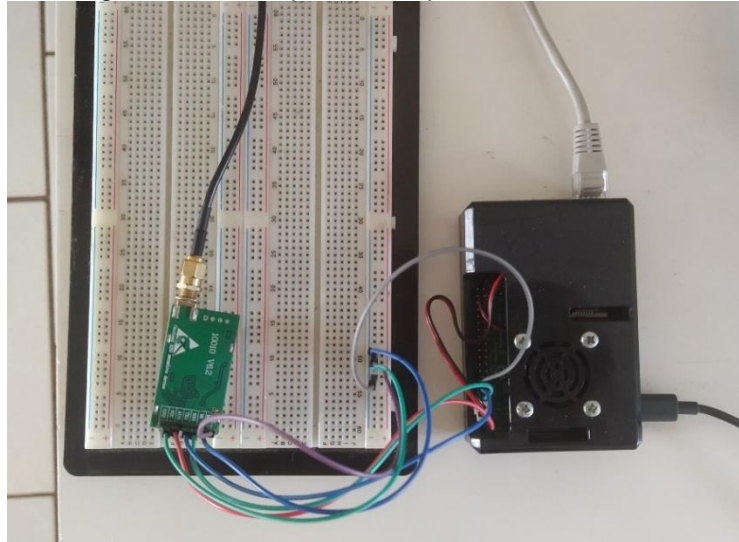
Fonte: Autor (2020).

4.3.2 Montagem do Receptor

Para montagem do receptor optou-se por simplificar e utilizar *jumpers* e uma *protoboard* além do *Raspberry PI 3* e do modulo LoRa E32. A montagem pode ser vista na Figura 24, onde se observar que o *Raspberry* foi colocado dentro de um case

contendo um *cooler*, isso se dá, pois, o minicomputador ficaria ligado por muito tempo e se constatou certa oscilação de desempenho atrelada a isso.

Figura 24 - Protótipo do Receptor montado

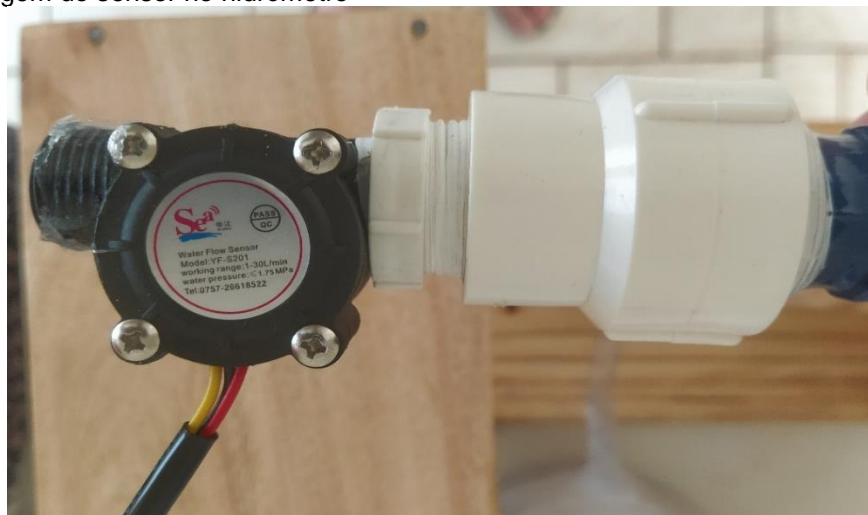


Fonte: Autor (2020).

4.3.3 Montagem do sensor de vazão

Apesar de simples é importante ressaltar a adaptação feita para conexão entre o hidrômetro que possui abertura de 1", e o sensor de vazão que possui abertura de $\frac{1}{2}$ ". Como pode ser visto na Figura 25 foram colocadas duas reduções na secção, uma de 1" para $\frac{3}{4}$ " e outra de $\frac{3}{4}$ " para $\frac{1}{2}$ ", é importante ressaltar que isso pode influenciar consideravelmente nos resultados e deve ser levado em conta durante as análises.

Figura 25 - Montagem do sensor no hidrômetro

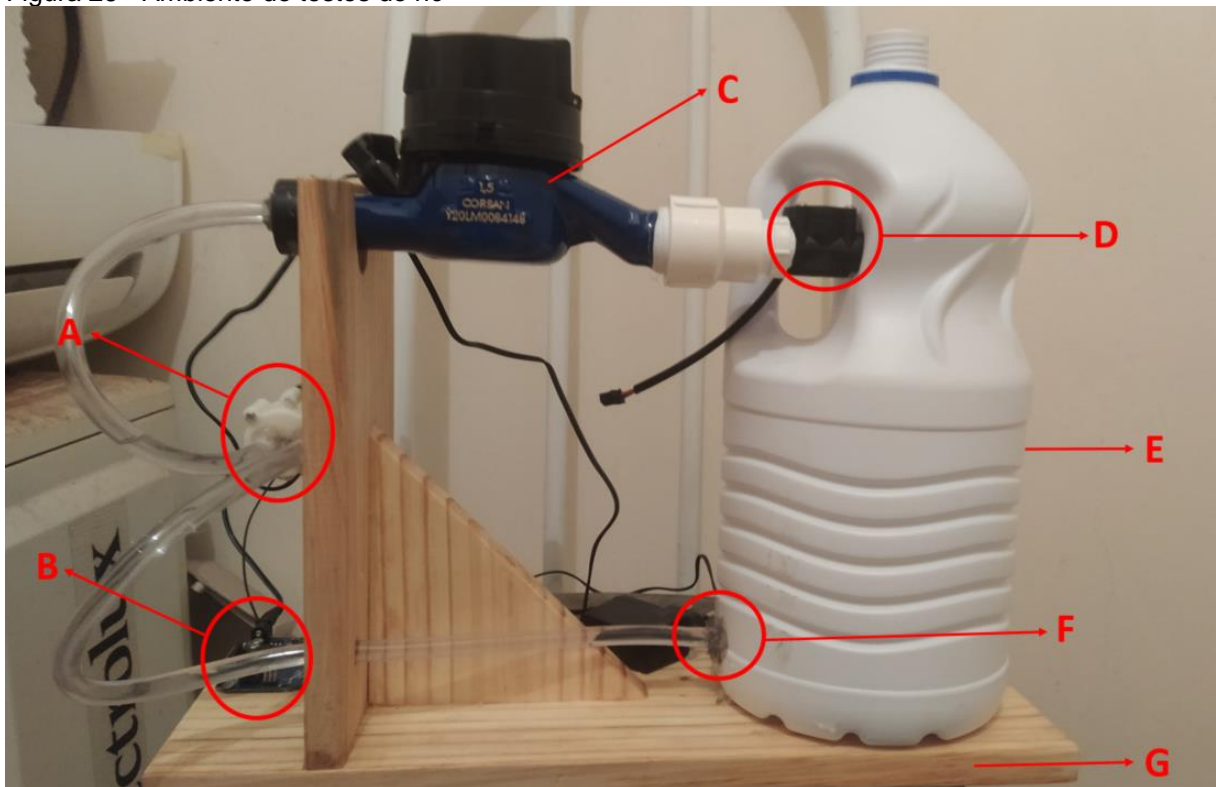


Fonte: Autor (2020).

4.3.4 Montagem do ambiente de teste

Conforme proposto, o sistema não seria montado em um hidrômetro conectado a companhia de água, porém se estabeleceu a necessidade da criação de um ambiente similar ao real para aplicação da metodologia. A Figura 26 apresenta o ambiente montado e sumariza cada componente utilizado, estes que serão mais detalhados na sequência.

Figura 26 - Ambiente de testes do nó



Fonte: Autor (2020).

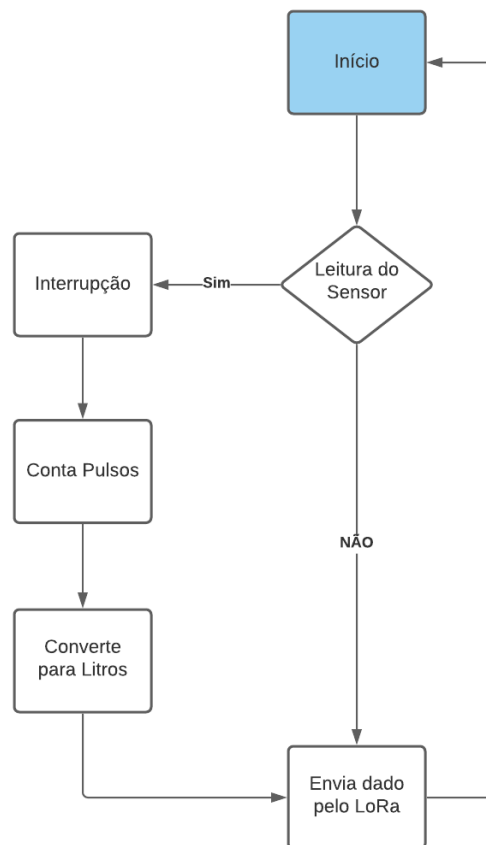
A composição do ambiente do teste se dá pela bomba de água (A), responsável por alimentar a vazão no hidrômetro, o arduino (B) que controla a bomba de água, o hidrômetro (C) que será usado para validação, o sensor de vazão (D) que coleta os dados, o reservatório da água (E), as mangueiras (F) conectas e vedadas para circulação da água e pôr fim a base em madeira (G) que sustenta os demais componentes.

4.4 PROGRAMAÇÃO

4.4.1 Programação do attiny

Para programação do attiny85 foi utilizado a IDE do Arduino e a linguagem C++ com pequenas modificações, um conjunto muito conhecido e utilizado para prototipagem de projetos. Seguindo o fluxograma da Figura 27 pode se entender a lógica aplicada no desenvolvimento do código e o seu funcionamento, o mesmo está na íntegra no APÊNDICE C.

Figura 27 - Fluxograma da programação do nó



Fonte: Autor (2020).

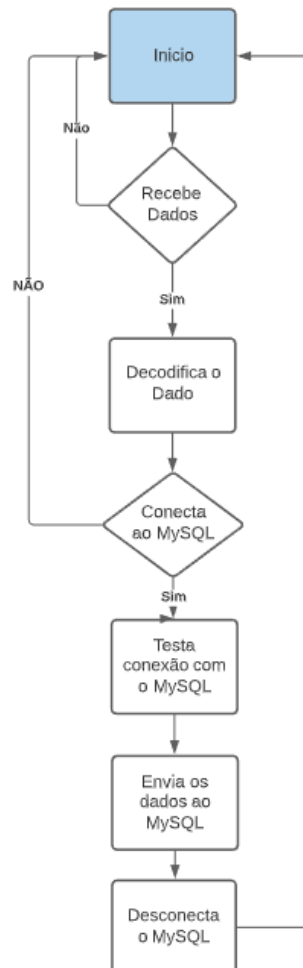
O código inicia quando o sistema é energizado, onde se o sensor de vazão realizar leitura de algum dado ele inicia uma interrupção de alguns segundos e conta os pulsos gerados nesse intervalo, após isso acontece a conversão dos pulsos para unidade de Litros e o mesmo é enviado através da rede LoRa e reinicia o código. Se o sensor não captar nenhuma informação ele envia o último dado coletado pela rede LoRa e reinicia o código.

4.4.2 Programação do Raspberry Pi

Abordando programação no *Raspberry Pi 3*, modelo B, é importante ressaltar que o sistema operacional escolhido foi o Raspbian, sistema gratuito baseado em Debian e otimizado para utilização no hardware do *Raspberry*. A partir dessa informação poderiam ser usadas várias linguagens diferentes para programação do código de recebimento da rede LoRa, porém a escolhida foi o Python 3, que apresenta grande acervo de bibliotecas e códigos de referência na internet, além de ser considerado uma linguagem de alto nível e mais fácil de ser lida.

Além do python foi utilizado a linguagem SQL, para executar uma *query* e enviar os dados recebidos para o banco de dados. Para programar foi utilizado o editor de texto do sistema operacional chamado “nano”, acessado e executado através do terminal do Debian. A lógica utilizada pode ser vista no fluxograma apresentado na Figura 28, já o código em suma pode ser encontrado no APÊNDICE D.

Figura 28 - Fluxograma programação do receptor



Seguindo o fluxograma o código se inicia e faz uma validação, se nenhum dado é recebido através do LoRa ele reinicia até que algum dado seja encontrado, se encontrado o mesmo é decodificado e o *Raspberry* tenta conectar ao banco de dados. Se o banco não for encontrado o código reinicia, se a conexão for estabelecida, há um teste nessa conexão, os dados decodificados então são enviados e a conexão com o banco é fechada, reiniciando o código para uma nova coleta de dados.

A questão de decodificação utilizada aqui é referente a divisão da *string* recebida do nó, que ao ser enviada é um texto corrido, contendo uma separação entre dados por dois pontos. Os dados precisam estar em variáveis diferentes para serem enviados ao banco de dados, para que sejam alocados corretamente nas colunas, para isso se utilizou da função “Split” do Python que divide baseado onde estão os dois pontos.

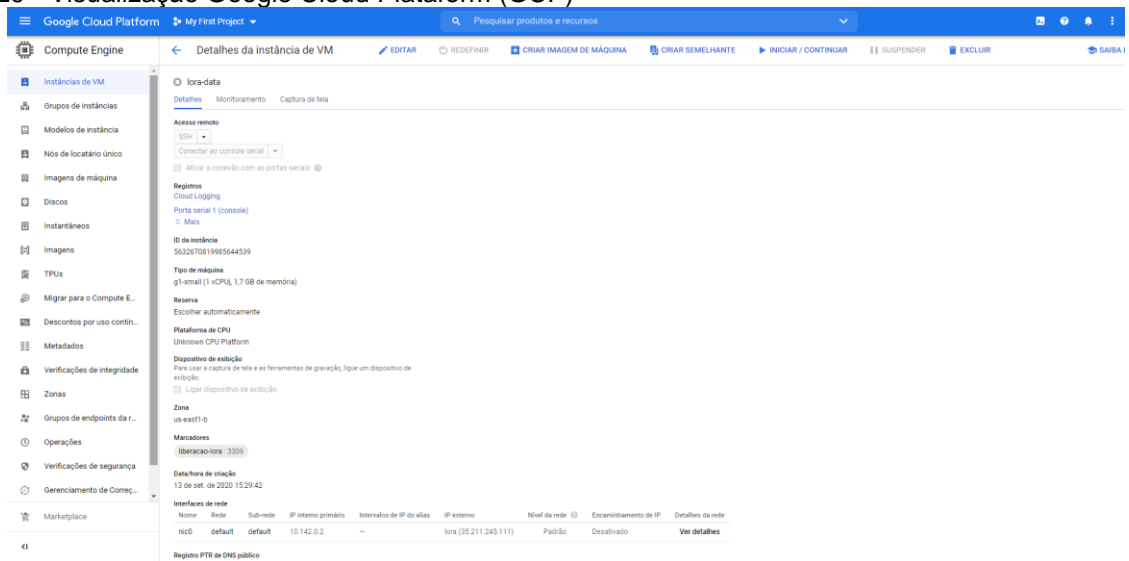
4.5 CONFIGURAÇÃO DO BANCO DE DADOS

No que tange a criação e uso do banco de dados, haviam duas possíveis escolhas. Uma opção se limita em criar um banco local e usar o armazenamento do próprio *Raspberry*, enquanto que a outra opção consiste em usar um servidor online dedicado, o que implica em algumas complexidades e custos. No entanto, para não correr o risco de atrapalhar o desempenho do computador optou-se pela segunda opção e apresentam-se os métodos e soluções usadas em seguida.

4.5.1 GCP

Para alocação do servidor foi utilizado o serviço “*Google Cloud Platform (GCP)*”, conforme ilustra a Figura 29, que permite a criação de uma máquina virtual entre outras aplicações. O mesmo dispõe gratuitamente um saldo de U\$300 por um período de 90 dias para utilização de servidores em múltiplas locações. Foi escolhido um servidor nomeado “g1-small”, contendo 1,7GB de memória RAM e localizado no “us-east-b”, ou seja, no leste dos Estados Unidos. Se escolheu este local pelo menor custo de hospedagem e porque a aplicação não necessita baixa latência.

Figura 29 - Visualização Google Cloud Platform (GCP)

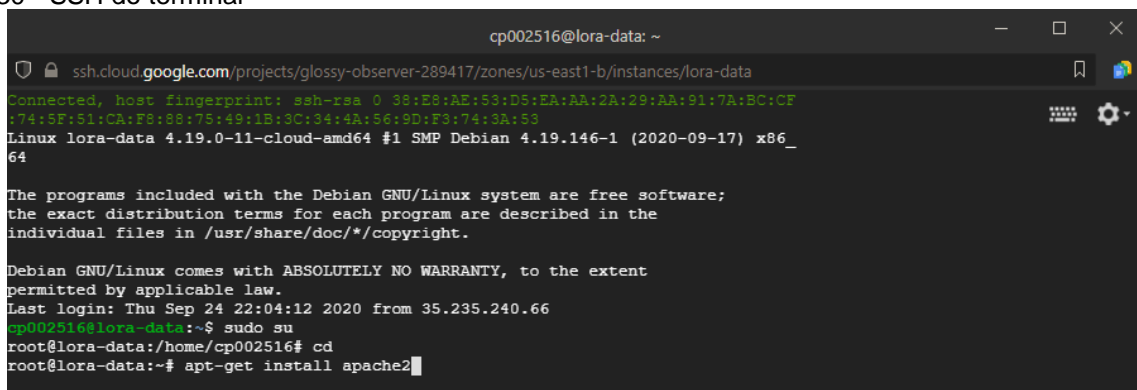


Fonte: Autor (2020).

4.5.2 Instalação do Banco de Dados

Como se utilizou de uma máquina virtual, com sistema operacional Debian, foi necessário fazer a instalação do servidor e de um SGBD através do terminal SSH do mesmo, Figura 30. O servidor HTTP instalado foi o Apache Server através do comando `apt-get install apache2` e iniciado através do comando `systemctl start apache2.service`, este que é o responsável pelos acessos a todos os recursos de um servidor *web*.

Figura 30 - SSH do terminal



Fonte: Autor (2020).

O SGBD escolhido foi o MySQL, que já foi citado durante o presente projeto, este que é a interface direta do usuário com o banco, tabelas e *queries* SQL para importação e exportação de dados. O conjunto de comando para instalação do SQL é mais complexo e depende de versões compatíveis, como o intuito do projeto não é

dar o passo a passo dessa instalação, vale se ressaltar o comando para instalação do PHP MySQL *“apt-get install php-mysql php-gd”*.

4.5.3 Configuração do Banco de Dados

Para começar a configurar o MySQL foi necessário a criação de um usuário e uma senha para acesso através do SSH do servidor. Para criação das credencias se usou o seguinte comando:

```
“CREATE USER 'USUARIO'@%' IDENTIFIED BY 'SENHA';”
```

Onde se substituiu o *Usuario* por “caio” e a *Senha* por “root”, as informações podem ser configuradas de acordo com o gosto e a necessidade do usuário. Além disso foi necessário liberar acesso de administrador para esse usuário através do comando:

```
“GRANT ALL PRIVILEGES ON * . * TO 'USUARIO'@%'”;
```

Para criação das tabelas o acesso dever ser feito através do PHPMyAdmin, que pode ser feito em qualquer navegador web, neste caso basta colocar o IP do servidor seguido por *“/phpmyadmin”*. Como mostrado na Figura 31, para acessar o SGBD basta colocar o usuário e a senha criados previamente nos campos “A” e “B” respectivamente, em seguida clicar em executar.

Figura 31 - Terminal de acesso ao phpMyAdmin

The image shows the phpMyAdmin login page. At the top, it says 'Bemvindo ao phpMyAdmin'. Below that is a language selection dropdown menu currently set to 'Português - Portuguese'. Underneath is a section titled 'Entrada' (Login) which contains two input fields. The first field is labeled 'Utilizador:' and contains the text 'caio'. The second field is labeled 'Palavra-passe:' and contains four asterisks '****'. Red rectangular boxes are drawn around both input fields. Red arrows point from the right side of the 'caio' box to the letter 'A', and from the right side of the '****' box to the letter 'B'. At the bottom right of the login form is a button labeled 'Executar'.

Fonte: Autor (2020).

Dentro do PHPMyAdmin foi criada a tabela para armazenamento dos dados de acordo com a Figura 32. Cada linha apresentada aqui corresponde a uma coluna e seus parâmetros. Os parâmetros importantes aqui são nome, tipo, tamanho de valores, predefinido e A_I (Auto Incremento).

Figura 32 - Colunas da tabela no banco de dados

Nome	Tipo	Tamanho/Valores*	Predefinido	Agrupamento (Collation)	Atributos	Nulo	A_I
ID	INT	11	Nenhum			<input type="checkbox"/>	<input checked="" type="checkbox"/>
DATA	DATETIME		CURRENT_TIME			<input type="checkbox"/>	<input type="checkbox"/>
ID_Hidrometro	INT	11	Nenhum			<input type="checkbox"/>	<input type="checkbox"/>
Leitura	DECIMAL	10,2	Nenhum			<input type="checkbox"/>	<input type="checkbox"/>

Fonte: Autor (2020).

Os campos com nome “ID” e “DATA” são informações do servidor, ou seja, não vem no conjunto enviado pelo nó. O “ID” é um valor inteiro que se auto incrementa a cada entrada, possibilitando controle do número de entradas totais no banco. Já a coluna “DATA” tem o formato “*datetime*” que equivale a data completa, e tem como predefinição “Current_Time”, que realiza o registro no momento, data e horário, em que o dado correspondente entra no banco.

As colunas nomeadas “ID_Hidrometro” e “Leitura” são os dados recebidos a partir do nó. Sendo assim, a primeira tem como função identificar o dispositivo, que seria importante se houvesse mais nós ligados, e que para uma aplicação real é fundamental. Em sequência “Leitura” tem como capacidade números decimais, com capacidade para dez números antes e dois após a virgula, essa coluna se refere a leitura do hidrômetro em litros propriamente.

4.5.4 Liberação da Porta 3306

Como padrão a conexão do MySQL utiliza a porta do servidor 3306, para isso foi necessário criar uma liberação. Foi criada uma regra vista na Figura 33 para liberação da porta, se ressalta que o acesso a porta está aberto para qualquer usuário, que pode causar problemas de segurança e que não deve ser usado esse método para aplicações reais.

Figura 33 - Liberação da porta no GCP

The screenshot shows the Google Cloud Platform console interface. The main content area displays the details of a VM instance. The 'SSH' tab is active, showing options to connect to the serial console. The instance details include:

- SSH:** Connect to serial console, Ativar a conexão com as portas seriais.
- Registros:** Cloud Logging, Porta serial 1 (console), Mais.
- ID da instância:** 5632670819985644539
- Tipo de máquina:** g1-small (1 vCPU, 1,7 GB de memória)
- Reserva:** Escolher automaticamente
- Plataforma de CPU:** Unknown CPU Platform
- Dispositivo de exibição:** Ligar dispositivo de exibição.
- Zona:** us-east1-b
- Marcadores:** liberacao-lora: 3305 (highlighted in red)
- Data/hora de criação:** 13 de set. de 2020 15:29:42
- Interfaces de rede:**

Nome	Rede	Sub-rede	IP interno primário	Intervalos de IP do alias	IP externo	Nível da rede	Encaminhamento de IP	Detalhes da rede
nic0	default	default	10.142.0.2	—	lora (35.211.245.111)	Padrão	Desativado	Ver detalhes
- Registro PTR de DNS público:**

Fonte: Autor (2020).

4.6 TESTES DO PROTÓTIPO

Esta seção tem como finalidade apresentar os testes no que tange a ajustes realizado e dados obtidos.

4.6.1 Ajustes iniciais

Com todos componentes montados e programados se fez um teste preliminar, afim de validar as leituras, e para isso se leu um intervalo de aproximadamente 500 leituras. O primeiro resultado não foi satisfatório, onde se encontrou aproximadamente 50L no mostrador analógico do hidrômetro e pouco mais de 24L nos dados coletados no banco de dados.

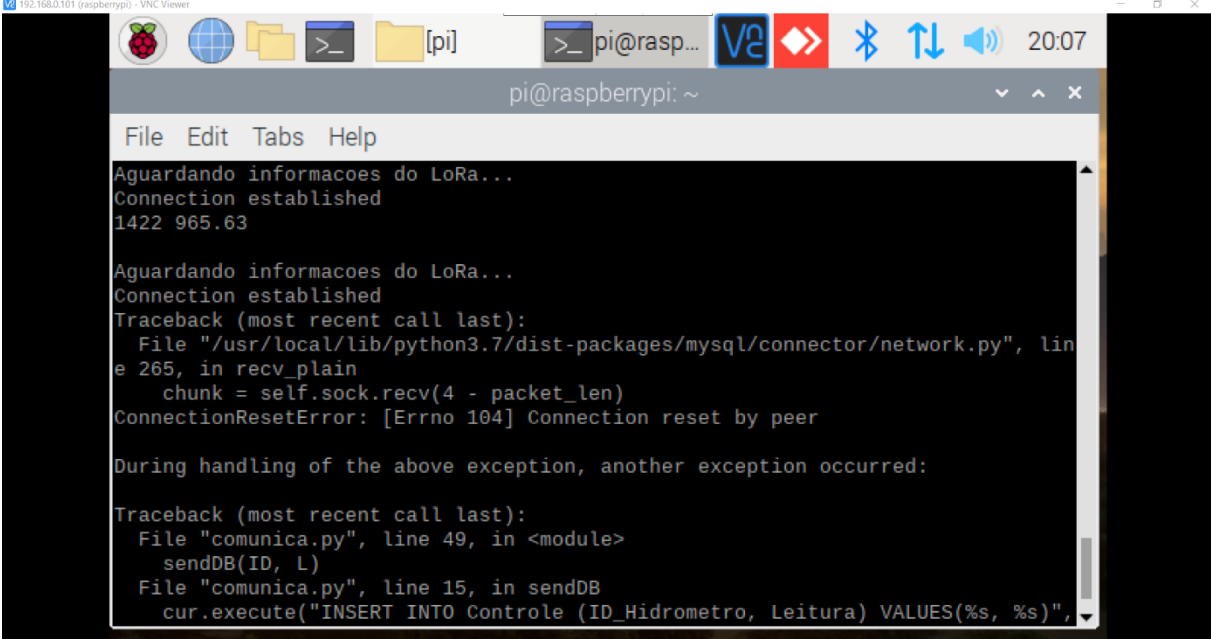
Após analisar o sistema se chegou à seguinte conclusão, a bomba que simula a passagem de água não gera vazão suficiente para encher a câmara do hidrômetro, este que por ser analógico interpreta uma rotação da sua turbina como se em capacidade máxima. Para solucionar o problema e possibilitar o seguimento do teste foi elevado o reservatório de água, onde o hidrômetro ficou com uma angulação em relação ao posicionamento anterior.

Dessa forma, o volume da câmara foi completo e o liquido se deslocou de forma uniforme, garantindo a leitura do hidrômetro. Em uma nova validação curta os valores

obtidos foram, no hidrômetro de 25L e nos dados coletados pelo sensor 25,1L. Sendo assim o sistema estava pronto para ser colocado a teste nos dois métodos citados.

Outro ajuste realizado foi devido a um erro encontrado durante a primeira coleta do primeiro método, que foi refeita após correção. O problema foi durante a execução do código de coleta no *Raspberry*, que devido a alguma interferência ou oscilação de energia parou sua execução. O erro pode ser visto no serial do minicomputador e está apresentado na Figura 34.

Figura 34 - Erro de execução do código do receptor



```

File Edit Tabs Help
Aguardando informacoes do LoRa...
Connection established
1422 965.63

Aguardando informacoes do LoRa...
Connection established
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/mysql/connector/network.py", line
e 265, in recv_plain
    chunk = self.sock.recv(4 - packet_len)
ConnectionResetError: [Errno 104] Connection reset by peer

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "comunica.py", line 49, in <module>
    sendDB(ID, L)
  File "comunica.py", line 15, in sendDB
    cur.execute("INSERT INTO Controle (ID_Hidrometro, Leitura) VALUES(%s, %s)",

```

Fonte: Autor (2020).

Foi colocado um “*delay*” de 2 segundos após o fechamento da conexão com o banco dados para evitar abarrotamento de informações. Possivelmente o problema não está ligado a esse aspecto, mas durante os dez dias seguintes de teste o código funcionou normalmente onde o problema não tornou a ocorrer.

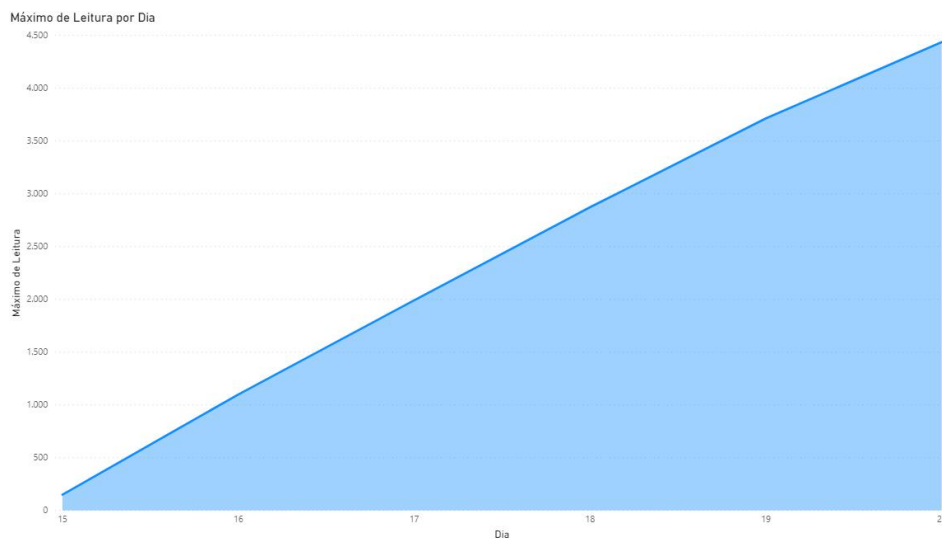
4.6.2 Resultados do Testes

Após aplicar a metodologia descrita e realizar a coleta dos dados se pode observar os resultados nessa seção. A apresentação se dá pela análise dos dois métodos e o funcionamento do sistema de forma geral e pôr fim a avaliação das grandezas coletas contra os valores reais do hidrômetro. A análise da perda de pacotes foi realizada pelo intervalo de tempo entre as transmissões, que era de

aproximadamente 10 segundos, perdas muito grandes resultaria em oscilações visíveis nos gráficos a seguir.

O primeiro método, com vazão contínua, registrou um total de 43908 leituras de entrada no período de 5 dias, iniciando em 0L e finalizando em 4435,45 L. O sistema ficou ligado por aproximadamente 118,47 horas com coletas a cada 10 segundos. Não foi necessário parar o teste fora dos horários de apuração.

Figura 35 - Gráfico das leituras do método 1

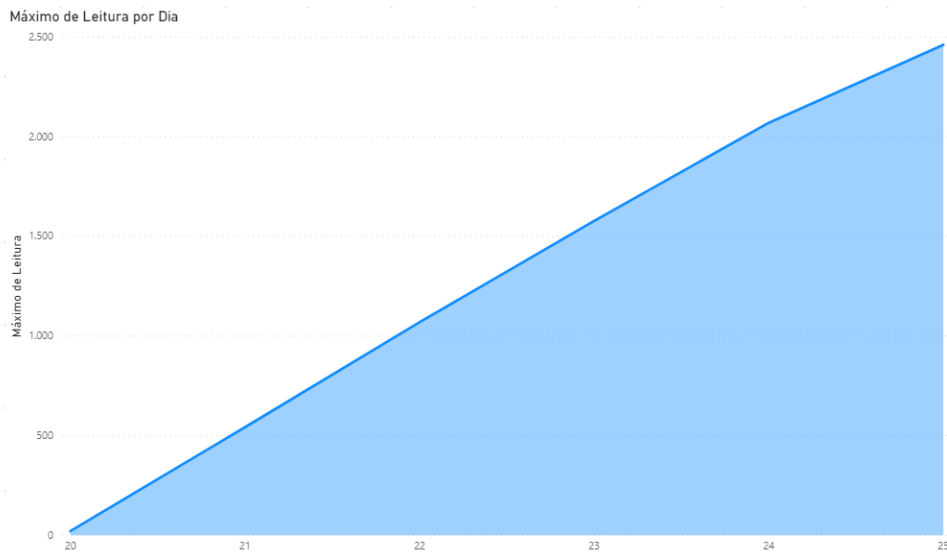


Fonte: Autor (2020).

Como observado na Figura 35 se pode observar que a coleta foi constante e que não houve interrupção ou interferência críticas a fim de derrubar as conexões entre Nó e Receptor, através da rede LoRa, ou entre Receptor e Banco de dados, através da query SQL. De forma geral o sistema se comportou bem ao fluxo contínuo.

O segundo método, com vazão oscilante, registrou um total de 42460 leituras durante 5 dias, sendo a leitura máxima 2462 litros. Para essa validação o sistema permaneceu ativo por 116,4 horas, onde o sistema de vazão ligava e desligava a cada 5 minutos, as leituras eram coletadas a cada 10 segundos, independentemente da existência ou não de vazão. Da mesma forma que o primeiro método, este não se fez necessário a interrupção das leituras fora do horário de aferimento.

Figura 36 - Gráfico das leituras do método 2



Fonte: Autor (2020).

A Figura 36 retrata o fluxo das leituras do segundo método, que possui uma curva fora do padrão no dia 24, esta que pode ser explicada pelo fato de que o sistema de vazão ficou desligado por mais tempo que o necessário. Desta forma pode-se dizer que o sistema se comportou corretamente a oscilações de vazão. Além de que, da mesma forma que o método anterior, este não foi afetado por intempéries que ocasionassem a queda das comunicações.

Como descrito na metodologia a Tabela 6 foi completa com os as grandezas referentes ao horário do aferimento e os valores naquele momento em litros, do hidrômetro e da instrumentação respectivamente. É importante reforçar que o dado da instrumentação foi coletado direto no banco de dados através do “PHPMyAdmin” para evitar inconsistência.

Tabela 6 - Padrão para coleta de dados completo

Dia	Método 1 - Vazão continua			Método 2 - Vazão Oscilante		
	Hora	Leitura Hidrômetro	Leitura Instrumentação	Hora	Leitura Hidrômetro	Leitura Instrumentação
1	15:35	908,1	908,6	16:33	466,8	448,39
2	15:33	925,11	866	16:05	516,4	517,24
3	15:57	942,5	917,15	16:10	525,5	505,24
4	16:20	924,2	846,5	16:00	530,1	542,18

5	15:27	892,00	897,36	16:24	486,4	449,03
---	-------	--------	--------	-------	-------	--------

Fonte: Autor (2020).

Analisando os dados pode se observar que existe uma variação tanto para mais quanto para menos de forma similar nos dois métodos de teste. Para expressar a perda de uma companhia se somou os valores de cada teste e se tirou o valor em porcentagem da diferença da leitura esperada para a atingida. No método de vazão constante a leitura do hidrômetro resultou em 3,4% maior que o resultado do nó, este que representa 156,3L. Por sua vez o teste de vazão variada apresentou a diferença da leitura do hidrômetro para a instrumentação de 2,5%, que equivale a 63,12L.

O sistema de instrumentação mostrou grau de inconsistência em sua acurácia, podendo ser proveniente do método de teste, da calibração do sensor ou ainda de oscilações na alimentação de componentes, tanto do protótipo quando dos usados para os testes. Em questão de precisão o protótipo se mostrou confiável, pois o sensor de vazão não apresentou perda de dados ou oscilações abruptas das leituras durante a medição.

Levando-se em conta que o projeto é apresentado em nível de protótipo, que possuiu algumas limitações durante os testes e que o custo do nó foi de aproximadamente R\$170,00 os resultados obtidos foram formidáveis. Sendo que produtos similares e com variações mínimas em nível comercial tem custo de aproximadamente R\$1500, e muitas vezes sem conexão remota. Além disso, se coloca que por serem produtos mecânicos os hidrômetros sofrem desgaste do tempo, o que coloca certa tolerância no seu valor de leitura, muitas vezes maiores que os apresentados aqui.

Com as corroborações apresentadas acima se viu a necessidade da aplicação do sistema em situações reais, sem interferência físicas e de grandezas, como no método utilizado neste projeto, como em um projeto piloto, por exemplo.

CONCLUSÃO

O estudo apresentado buscou desenvolver uma solução, a nível de protótipo, de uma IoT para coleta de dados em hidrômetros analógicos, utilizando tecnologia de transmissão LoRa e conexão com banco de dados, analisar quesitos como viabilidade, confiabilidade e precisão.

Obteve-se que as tecnologias utilizadas se comportam de forma satisfatória umas com as outras. Contudo, o sensor de vazão empregado apresentou não ser a melhor opção para aplicação, principalmente devido ao seu tamanho, sendo assim existem muitos outros modelos que podem ser comparados e colocados a campo para encontrar a melhor opção.

Em questão de confiabilidade a tecnologia LoRa obteve bons resultados, pois após dez dias de funcionamento contínuo os dados mantinham sendo enviados e recebidos corretamente a cada 10 segundos. A utilização de um banco externo pode ser citada aqui, pois facilita questões de operação humano-máquina e também não apresentou falha de comunicação.

As discrepâncias das leituras mostradas nos resultados remetem a acurácia do sensor, que foi acima da expectativa em decorrência as condições impostas pelo projeto. Questionando precisão, é importante gabar a atuação não só do sensor, mas do nó como um todo. De forma resumida o sistema apresentou diferenças que não foram grandes o suficiente para impactar negativamente o protótipo.

Infere-se, portanto, os objetivos foram todos atingidos, sendo alguns com maior maestria que outros. De ampla forma, o protótipo mostrou grande potencial futuro e aspectos interessantes na interação entre novas e antigas tecnologias. Com esse protótipo, a ideia inicial se mostrou funcional e pode ser melhorada ao longo de futuros projetos.

SUGESTÃO DE MELHORIA E TRABALHOS FUTUROS

Como sugestões de melhoria cita-se testes em ambientes reais com mais intempéries e ao mesmo tempo com uma vazão mais estável em questão volumétrica, realizar o dimensionamento do sensor de vazão ou pesquisa para escolha de um outro modelo, mais preciso e confiável. Além disso, seria importante testar o *gateway* em comunicação com mais de um nó, validando assim não só a coerência da coleta, mas também a comunicação com o banco de dados.

Para trabalhos futuros sugere-se alguns pontos específicos que podem ser desenvolvidos mais a fundo. Primeiramente, em questão de *hardware*, seria interessante desenvolver um estudo para levar todos os componentes do nó para uma PCB, afim de utilizar somente componentes necessários e reduzir o tamanho e o consumo energético do mesmo. A utilização de um *gateway* padrão de LoRa e a comparação com um protótipo nos quesitos de custo contra capacidade e precisão.

Vale também um estudo voltado aos quesitos de segurança da rede LoRa, tendo como foco questões de interferência e melhoria na criptografia dos dados transmitidos. Além disso, em questão de software, o código tanto do nó quanto do receptor pode ser estudado, focado em reduzir seu tamanho, torná-los mais rápidos e mais eficiente para a movimentação de dados.

Por fim, com um modelo mais estruturado do projeto, a criação de um sistema para monitoramento dos dados coletados e até mesmo do status dos nós em uma cidade. Para isso, sugere-se a utilização de tecnologias como sistemas supervisórios ou *business intelligence* (BI), ambas bem difundidas quando se trata de tecnologia.

REFERÊNCIAS

- 6 RASPBERRY PI (TRADING) LTD. **Raspberry Pi Compute Module 3 (CM3)**. Disponível em: < https://www.raspberrypi.org/documentation/hardware/compute-module/datasheets/rpi_DATA_CM_1p0.pdf >. Acesso em: 18 maio. 2020.
- ALVES, William Pereira. **BANCO DE DADOS**. 1. ed. São Paulo: Érica, 2014.
- AUGUSTIN, A.; YI, J.; CLAUSEN, T.; TOWNSLEY, W. M. **A Study of LoRa: Long Range & Low Power Networks for the Internet of Things** [artigo científico]. Disponível em: < <https://www.mdpi.com/1424-8220/16/9/1466/htm#> >. Acesso em: 18 maio. 2020.
- ATMEL. **Atmel 8-bit AVR Microcontroller with 2/4/8K Bytes In-System Programmable Flash**. Disponível em: < https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2586-AVR-8-bit-Microcontroller-ATtiny25-ATtiny45-ATtiny85_Datasheet.pdf >. Acesso em: 18 maio. 2020.
- BARNES, Russell *et al.*(ed.). **Raspberry Pi Beginner's Guide: How to use your new computer**. 1. ed. Cambridge: Raspberry Pi PRESS, 2018.
- BERTOLETI, Pedro. **Projetos com ESP32 e LoRa**. 1. Ed. São Paulo: Instituto Newton C Braga, 2019.
- BOR, M.; VIDLER, J.; ROEDIG, U. **LoRa for the Internet of Things** [artigo científico]. Disponível em: <https://eprints.lancs.ac.uk/id/eprint/77615/1/MadCom2016_LoRa_MAC.pdf>. Acesso em: 18 maio. 2020.
- DAMAS, Luís. **SQL: structured query language**. 6. ed. Rio de Janeiro: LTC, 2014.
- DELGADO, José; RIBEIRO, Carlos. **Arquitetura de Computadores**. 5. ed. Rio de Janeiro, 2017.
- DUNN, William C. **Fundamentos de Instrumentação industrial e Controle de Processos**. 1. ed. Porto Alegre: Bookman, 2013.
- EASYTRONICS. **YFS201 Sensor de Vazão 30L/min | YF-S201**. Disponível em: <<https://www.easytronics.com.br/sensor-de-vazao-yfs201>>. Acesso em: 1 outubro. 2020.
- ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de Banco de Dados**. 4. ed. São Paulo: Pearson Addison Wesley, 2005.
- FAROOQ, M. U. et. al. **A Review on Internet of Things (IoT)** [artigo científico]. Disponível em: <<https://espaciodigital.pe/wp-content/uploads/2020/04/AReviewonInternetofThingsIoT.pdf>>. Acesso em: 13 setembro. 2020.
- FRANCHI, Claiton Moro. **Instrumentação de Processos Industriais: Princípios e aplicações**. 1. ed. São Paulo: Érica, 2015.

GEORGIU, Orestis; RAZA, Usman. **Low Power Wide Area Network Analysis: Can LoRa Scale?**, Disponível em: <<https://ieeexplore.ieee.org/abstract/document/780360>>. Acesso em: 13 setembro. 2020.

GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. 6. ed. São Paulo: Atlas, 2017.

HEUSER, Carlos Alberto. **Projeto de banco de dados**[recurso eletrônico]. 6.ed. Porto Alegre: Bookman, 2009.

LORA ALLIANCE. Technical Marketing Workgroup 1.0. **A technical overview of LoRa and LoRaWAN**. Estados Unidos: Fremont. Novembro 2015.

MEDEIROS, Emmanuel Leite de. **Desenvolvimento de um Sistema de Aquisição de Dados em Rede Híbrida de Comunicação de uma Planta Hidráulica focado em LoRa** [artigo científico]. Disponível em: <http://www.cear.ufpb.br/arquivos/cgee/TCC/TCC_-_Emmanuel_Leite_de_Medeiros_-_Versão_Final.pdf>. Acesso em: 13 setembro. 2020.

MINISTÉRIO DO DESENVOLVIMENTO, INDÚSTRIA E COMÉRCIO EXTERIOR - MDIC. **Portaria nº nº 246, de 17 de outubro de 2000**. 1. Rio de Janeiro, 17 out. 2000.

MORAIS, I. S.; et. al. **Introdução a Big Data e Internet das Coisas (IOT)**. São Paulo: SAGAH EDUCAÇÃO S.A., 2018.

MCEWEN, Adrian; CASSIMALLY, Hakim. **Designing the Internet of Things**. 1. ed. Reino Unido: John Wiley and Sons, Ltd., 2014.

ORACLE. **MySQL Database Service: easy to use, secure, enterprise-ready**[recurso eletrônico]. Disponível em: <<https://www.oracle.com/a/ocom/docs/mysql/mysql-database-service-ebook.pdf>>. Acesso em: 18 maio. 2020.

RIBEIRO, Jonatas Magno Tavares. **Uma aplicação da tecnologia LoRa em um ambiente hospitalar** [artigo científico]. Disponível em: <<https://repositorio.ufsc.br/handle/123456789/197683>>. Acesso em: 13 setembro. 2020.

TEIXEIRA, G. B.; ALMEIDA, J. V. P. de. **REDE LORA® E PROTOCOLO LORAWAN® APLICADOS NA AGRICULTURA DE PRECISÃO NO BRASIL** [artigo científico]. Disponível em: <http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/8436/1/PG_COELE_2017_2_01.pdf>. Acesso em: 18 maio. 2020.

SANTOS, B. P.; et. al. **Internet das Coisas: da Teoria à Prática** [artigo científico]. Disponível em: <<https://homepages.dcc.ufmg.br/~mmvieira/cc/papers/internet-das-coisas.pdf>>. Acesso em: 13 setembro. 2020.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Sistema de banco de dados**. 7. ed. Rio de Janeiro: LTC, 2020.

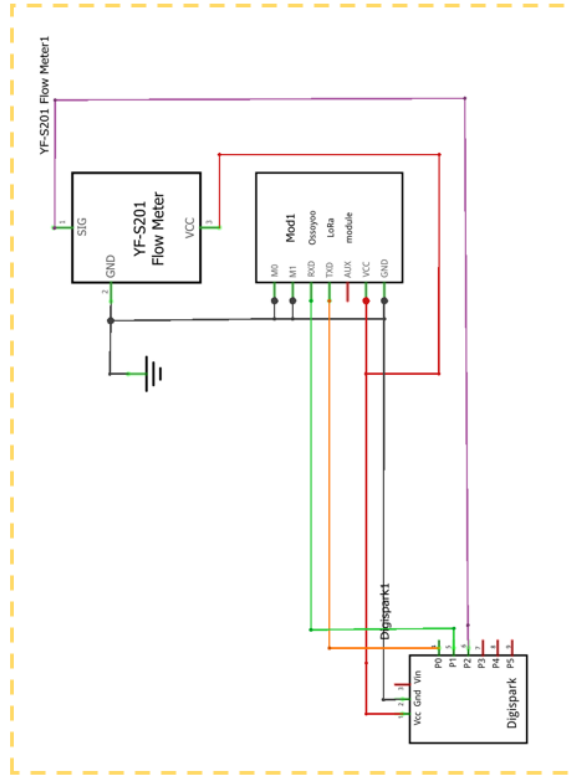
SILVA, Juliana. **COMO APLICAR HIDRÔMETROS INDIVIDUAIS EM EDIFICAÇÕES**. Disponível em: <<http://maisengenharia.altoqi.com.br/wp-content/uploads/2018/07/ebook-hidro-hidrometro-individual.pdf>>. Acesso em: 13 setembro. 2020.

SOISSON, Harold E. **Instrumentação Industrial**. 1. Ed. Rio de Janeiro: Hemus, 2002.

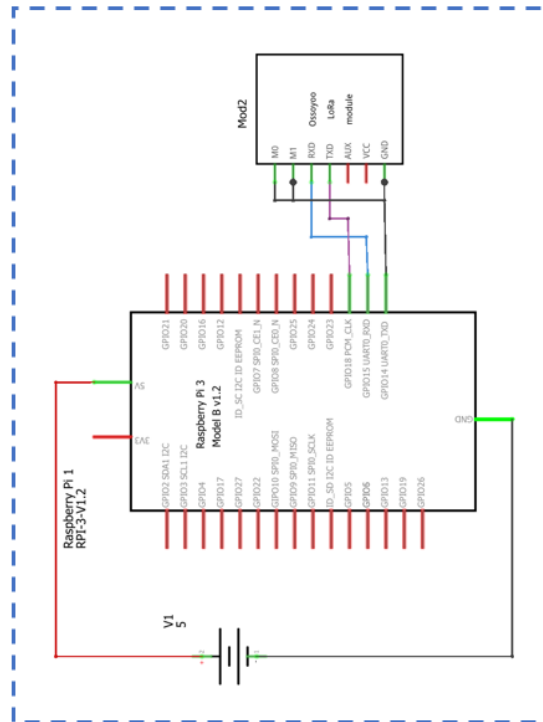
YIFA THE PLASTICS LTD. **YF-S201: Prodcut Introduction**. Disponível em: < <https://www.hobbytronics.co.uk/datasheets/sensors/YF-S201.pdf> >. Acesso em: 18 maio. 2020.

APÊNDICE A – CIRCUITO ELÉTRICO DO PROJETO

Nó



Receptor



APÊNDICE B – DESENHO DETALHADO DO INVÓLUCRO

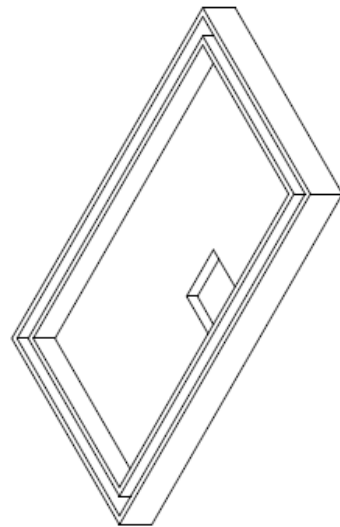
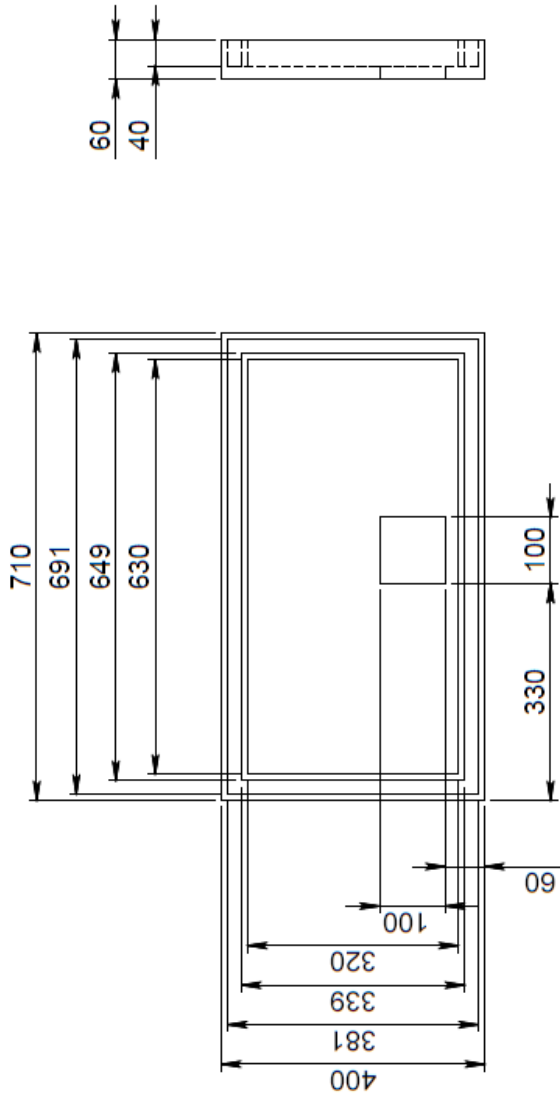
A seguir encontram-se os desenhos detalhados do involucro desenvolvido para o projeto, conforme listado na Tabela 7.

Tabela 7 – Componentes detalhados

Código	Descrição	Quantidade
Base	Caixa para circuito	1
Tampa	Tampa para vedação	1

Fonte: Autor 2020.

Tolerâncias não especificadas		
Até 13mm	±0,10mm	±0,5
entre de 13 até 30	±0,2	±0,8
entre de 30 até 100	±0,3	±1,2
entre de 100 até 300	±0,5	±1,8
Dimensões Angulares ±1°		



<p>ESTE DESENHO E AS INFORMAÇÕES NELE CONTIDAS SÃO DE NOSSA PROPRIEDADE E SO PODEM SER USADOS POR TERCEIROS MEDIANTE AUTORIZAÇÃO.</p>				DATA: 16/10/2020		
MÉTRICO	1º DIEDRO	DESENHADO POR: Caio Henrique Penning	CA	REV. DATA	POR	APROV.
PESO:	ERA:	SIMILAR:	DENOMINAÇÃO: Tampa			
MATERIAL REF:		CÓDIGO:		FOLHA: A4		
MATERIAL:		ESCALA: 1:1	REVISÃO: A	VERSÃO:		

APENDICE C – CÓDIGO DO NÓ

```

#include <SoftSerial.h> // Biblioteca para transmissão LoRa
SoftSerial loraSerial(0, 1); // TX - MOSI, RX - MISO

//----- Variaveis da Transmissão -----
    int ID = 1422; // ID representativo da "casa"

//----- Variaveis do Sensor -----
    float vazao; //Variável para armazenar o valor em L/min
    int contaPulso; //Variável para a quantidade de pulsos
    int i = 0; //Variável para segundos
    float Litros = 0; //Variável para Quantidade de agua
    float MiliLitros = 0; //Variavel para Conversão

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

//Sets para Sensor
    pinMode(2, INPUT); // Conexão do sensor ao pino
    attachInterrupt(0, incpulso, RISING); //Configura o pino 2(Interrupção 0)
interrupção

//Sets para Trasmissão LoRa
    loraSerial.begin(9600);
}

void loop() {
    contaPulso = 0; //Zera a variável
    sei(); //Habilita interrupção
    delay (1000); //Aguarda 1 segundo
    cli(); //Desabilita interrupção

```

```
vazao = contaPulso / 5.5; //Converte para L/min
i++;

MiliLitros = vazao / 60;
Litros = Litros + MiliLitros;

loraSerial.print(ID);
loraSerial.print(":");
loraSerial.print(Litros);
loraSerial.print("\r\n");
}

void incpulso () {
    contaPulso++; //Incrementa a variável de pulsos
}
```

APENDICE D – CÓDIGO DO RECEPTOR

```

def sendDB(ID2, Litros):
    try:
        conn = mysql.connector.connect(**db)
        print("Connection established")
    except mysql.connector.Error as err:
        if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
            print("Something is wrong with the user name or password")
        elif err.errno == errorcode.ER_BAD_DB_ERROR:
            print("Database does not exist")
        else:
            print(err)
    else:
        # Criar um objeto para o cursor. Isso executa todas as queries necessarias
        cur = conn.cursor()
        #Mandar Banco
        cur.execute("INSERT INTO Controle (ID_Hidrometro, Leitura) VALUES(%s, %s)",
(ID2, Litros))
        conn.commit()
        cur.close()
        con.close()
        time.sleep(2)

import RPi.GPIO as GPIO
import time
import serial
import mysql.connector
from mysql.connector import errorcode

GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)

config = {

```



```
'host':'35.211.245.111',  
'user':'caio',  
'password':'root',  
'database':'lora_data'  
}
```

```
#Configura a serial e a velocidade de transmissao
```

```
ser = serial.Serial('/dev/serial0', 9600)
```

```
time.sleep(2)
```

```
while(1):
```

```
    print ("Aguardando informacoes do LoRa...")
```

```
    # Aguarda a string na serial
```

```
    x = ser.readline()
```

```
    # Decodifica a string e manda variaveis individuais para o banco
```

```
    X = x.decode('utf-8')
```

```
    ID, L = X.split(':')
```

```
    sendDB(ID, L)
```

```
    # Mostra na tela a string recebida
```

```
    print (ID,L)
```