

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO
MESTRADO EM TECNOLOGIAS COMPUTACIONAIS PARA O AGRONEGÓCIO

MARCIELI PAULA LANGER

**TECNOLOGIAS E PROTOCOLOS DE COMUNICAÇÃO PARA REDES
DE SENSORES SEM FIO APLICADOS À CENÁRIOS AGRÍCOLAS**

Medianeira

2021

MARIELI PAULA LANGER

**TECNOLOGIAS E PROTOCOLOS DE COMUNICAÇÃO PARA REDES
DE SENSORES SEM FIO APLICADOS À CENÁRIOS AGRÍCOLAS**

**Communication Technologies and Protocols for Wireless Sensor Networks
Applied to Agricultural Scenarios**

Dissertação apresentada como requisito para
obtenção do título de Mestre em Tecnologias
Computacionais para o Agronegócio da Universidade
Tecnológica Federal do Paraná (UTFPR).

Orientador: Prof. Dr. Claudio Leones Bazzi

Coorientadora: Profa. Dra. Gloria Patricia Lopez
Sepulveda

MEDIANEIRA

2021

TERMO DE LICENCIAMENTO

Esta obra está licenciada sob uma Licença Creative Commons Atribuição, Não Comercial, Compartilha Igual 4.0 Internacional (CC BY-NC-SA 4.0). Para ver uma cópia desta licença, visite o endereço <http://creativecommons.org/licenses/by-nc-sa/4.0/> ou envie uma carta para Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.



[4.0 Internacional](http://creativecommons.org/licenses/by-nc-sa/4.0/)



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Medianeira



MARCIELI PAULA LANGER

**TECNOLOGIAS E PROTOCOLOS DE COMUNICAÇÃO PARA REDES DE SENSORES SEM FIO
APLICADOS À CENÁRIOS AGRÍCOLAS**

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre em Tecnologias Computacionais Para O Agronegócio da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Tecnologias Computacionais Aplicadas À Produção Agrícola E Agroindústria.

Data de aprovação: 08 de Julho de 2021

Prof Claudio Leones Bazzi, Doutorado - Universidade Tecnológica Federal do Paraná

Prof.a Fabiana Costa De Araujo Schutz, Doutorado - Universidade Tecnológica Federal do Paraná

Prof Luciano Gebler, Doutorado - Embrapa

Prof Paulo Cesar Tonin, Doutorado - Universidade Tecnológica Federal do Paraná

Documento gerado pelo Sistema Acadêmico da UTFPR a partir dos dados da Ata de Defesa em 08/07/2021.

Aos meus amados pais.

AGRADECIMENTOS

Ao professor orientador Dr. Claudio Leones Bazzi pela oportunidade concedida e confiança.

À professora co-orientadora Dra. Gloria Patricia Lopez Sepulveda pelo apoio prestado.

À minha família e amigos, pelo apoio, paciência, compreensão e motivação.

Aos professores e colegas do programa, sem os quais não haveria conhecimento, direcionamentos e apoio para o cumprimento do projeto.

Aos acadêmicos de Engenharia Elétrica Frederico (AGRILAB), Matheus, Jailson e Victor (LAD) pela ajuda na confecção das placas nas quais os módulos foram acomodados e demais conexões elétricas sem as quais os módulos não operariam de acordo com o projeto.

Ao colega, Alfredo Erdmann por toda ajuda e ensinamentos prestados para a configuração do gateway LoRa/LoRaWAN.

Ao CNPQ (projeto da bolsa de pesquisador nº 309983/2020-7), à UTFPR (HPP2019010000555 e E3K2021020000077) e ao MAPA - Ministério da Agricultura, Pecuária e Abastecimento (Termo de Execução Descentralizada TC001/2018; documentos 2018NC000083 e 2018NC001744), pelo apoio e recursos oferecidos para a compra dos materiais necessários à montagem das arquiteturas previstas nos cenários de testes.

“Viva como se fosse morrer amanhã.
Aprenda como se fosse viver para
sempre.”

Mahatma Gandhi

RESUMO

LANGER, Marcieli P. **Tecnologias e Protocolos de Comunicação para Redes de Sensores Sem Fio Aplicados à Cenários Agrícolas**. 2021. Dissertação (Mestrado em Tecnologias Computacionais para o Agronegócio) - Universidade Tecnológica Federal do Paraná.

A agricultura é um importante setor da economia brasileira e mundial, e assim como outros setores, é demandante do uso das tecnologias. Porém, o que se percebe são barreiras tecnológicas, principalmente no que se refere à infraestrutura de rede, para que as automações agrícolas sejam ainda mais intensas. Com base nestes desafios e propondo minimizá-los, o presente projeto objetiva avaliar tecnologias e protocolos de comunicação para redes de sensores sem fio (WSN) apropriados à comunicação máquina à máquina (M2M) em diferentes cenários de automação de atividades agrícolas. Por meio de pesquisa bibliográfica foram descritas as características das principais tecnologias WSN utilizadas em soluções IoT (internet das coisas), avaliando variáveis importantes às características destas aplicações, buscando dirimir problemas comuns, principalmente à aplicações em cenários agrícolas, onde a extensão variável das áreas pode requerer maior alcance de comunicação, além da inerente ausência de alimentação por energia elétrica. Das tecnologias estudadas, quatro demonstraram ser aplicáveis à cenários agrícolas: LoRa/LoRaWAN, ZigBee, SigFox e WiFi(ah), entretanto, destas duas foram elencadas como objetos do estudo: ZigBee e LoRa. Foram determinados cenários de testes considerando estudos bibliográficos, com abrangência a Mesorregião Oeste do Paraná. A partir da seleção das tecnologias e determinação dos cenários de testes, as topologias foram montadas e os testes executados nas culturas de soja e posteriormente de milho. Através dos testes foram estabelecidos os resultados, estes por sua vez apontam a tecnologia LoRa/LoRaWAN como a ideal para cenários que exigem maior alcance e possuem maior incidência de bloqueio por vegetação, especialmente percebido pelo não funcionamento do ZigBee nos cenários da cultura de milho. ZigBee por sua vez é uma boa opção para situações em que a distância de transmissão é curta e com nenhum ou pouco bloqueio físico da transmissão.

Palavras-chave: Internet das Coisas; Agricultura Inteligente; Agricultura de Precisão; LoRa; LoRaWAN; ZigBee.

ABSTRACT

LANGER, Marcieli P. **Communication Technologies and Protocols for Wireless Sensor Networks Applied to Agricultural Scenarios**. 2021. Dissertação (Mestrado em Tecnologias Computacionais para o Agronegócio) - Universidade Tecnológica Federal do Paraná.

Agriculture is an important sector of the Brazilian and world economy, and like other sectors, it demands the use of technologies. However, what are perceived are technological barriers, mainly with regard to network infrastructure, so that agricultural automation is even more intense. Based on these challenges and proposing to minimize them, this project aims to evaluate communication technologies and protocols for wireless sensor networks (WSN) appropriate for machine-to-machine (M2M) communication in different scenarios for automation of agricultural activities. Through bibliographic research, the characteristics of the main WSN technologies used in IoT (Internet of Things) solutions were described, evaluating important variables to the characteristics of these applications, seeking to solve common problems, mainly to applications in agricultural scenarios, where the variable extension of the areas can require greater communication range, in addition to the inherent absence of power supply. Of the technologies studied, four proved to be applicable to agricultural scenarios: LoRa/LoRaWAN, ZigBee, SigFox and WiFi(ah), however, of these two were listed as study objects: ZigBee and LoRa. Test scenarios were determined considering bibliographic studies, covering the West Mesoregion of Paraná. From the selection of technologies and the determination of test scenarios, the topologies were assembled and the tests were carried out on soybean and later corn crops. Through the tests the results were established, these in turn point out the LoRa/LoRaWAN technology as the ideal one for scenarios that require greater reach and have a higher incidence of blockage by vegetation, especially perceived by the non-functioning of the ZigBee in the corn culture scenarios. ZigBee in turn is a good option for situations where the transmission distance is short and with no or little physical blockage of the transmission.

Key-words: Internet of Things; Smart Agriculture; Precision agriculture; LoRa; LoRaWAN; ZigBee.

LISTA DE ILUSTRAÇÕES

Figura 1 – Áreas de aplicação de IoT.....	20
Figura 2 – Arquitetura IoT proposta para conectar áreas rurais.....	23
Figura 3 – Arquitetura IoT proposta para aplicações agroindustriais e ambientais....	24
Figura 4 – Arquitetura de rede para projetos LoRaWAN.....	30
Figura 5 – Arquitetura ChirpStack LoRaWAN.....	32
Figura 6 – Topologias ZigBee.....	34
Figura 7 – Topologia de redes de sensores sem fio proposta para solução de agricultura de precisão usando ZigBee.....	34
Figura 8 – Exemplificação da topologia LoRa utilizada no projeto.....	38
Figura 9 – Projeto do Nó LoRa.....	40
Figura 10 – Nó LoRa.....	41
Figura 11 – Projeto do <i>Gateway</i> LoRaWAN.....	45
Figura 12 – <i>Gateway</i> LoRaWAN montado.....	46
Figura 13 – Servidor de aplicação do projeto LoRaWAN.....	47
Figura 14 – Seleção de valores de temperatura armazenados na base InfluxDB.....	47
Figura 15 – Exemplificação da topologia ZigBee utilizada no projeto.....	48
Figura 16 – Projeto do Dispositivo Final ZigBee.....	50
Figura 17 – Projeto do Roteador ZigBee.....	51
Figura 18 – Dispositivo final ZigBee montado.....	52
Figura 19 – Roteador ZigBee montado.....	52
Figura 20 – Projeto do Coordenador ZigBee.....	55
Figura 21 – Coordenador ZigBee montado.....	56
Figura 22 – Seleção de valores capturados pelos sensores e armazenados na base InfluxDB.....	58
Figura 23 – Instalação dos dispositivos finais.....	59
Figura 24 – Pontos de instalação dos componentes da rede LoRa/LoRaWAN na cultura do milho.....	61
Figura 25 – Pontos de instalação dos componentes da rede ZigBee na cultura da soja.....	62
Figura 26 – Gráfico reproduzindo os percentuais de transferência de pacotes utilizando LoRa/LoRaWAN na soja.....	64
Figura 27 – Gráfico reproduzindo os percentuais de transferência de pacotes utilizando ZigBee na soja.....	65
Figura 28 – Gráfico reproduzindo os percentuais de transferência de pacotes utilizando LoRa/LoRaWAN no milho.....	67

LISTA DE QUADROS

Quadro 1 - Cenários de aplicação das principais tecnologias WSN.	26
Quadro 2 - Características das principais tecnologias WSN.	27
Quadro 3 - Vantagens e desvantagens das principais tecnologias WSN.	28
Quadro 4 - Cenários de testes.....	37
Quadro 5 – Esquema de ligações entre Arduino Mega 2560 e o Módulo LoRa.....	41
Quadro 6 – Esquema de ligações entre Arduino Mega 2560 e o sensor GUVa.....	41
Quadro 7 – Esquema de ligações entre Arduino Mega 2560 e o sensor BME.....	42
Quadro 8 – Bibliotecas utilizadas para execução do nó LoRa no Arduino	42
Quadro 9 – Código alterado no código fonte da biblioteca LoRaWan	43
Quadro 10 – Configuração do <i>end device</i> LoRa usando a biblioteca LoRaWan.....	44
Quadro 11 – Coleta da temperatura, encapsulamento usando Cayenne LPP e envio do pacote LoRaWAN.....	44
Quadro 12 – Configuração dos rádios ZigBee.	53
Quadro 13 – Bibliotecas utilizadas para execução do dispositivo final ZigBee no Arduino.....	53
Quadro 14 – Coleta da temperatura e envio dos dados via escrita na porta serial. ..	54
Quadro 15 – Recebimento e retransmissão de dados por parte do Roteador.	54
Quadro 16 – Bibliotecas utilizadas para execução do programa Python no Coordenador ZigBee	57
Quadro 17 – Código python responsável pela instanciação e inicialização das variáveis e conexões.....	57
Quadro 18 – Código python responsável pela leitura e persistência dos dados recebidos	58
Quadro 19 – Exemplo de registros de leituras dos dispositivos finais LoRa, compilados em uma única tabela	63
Quadro 20 – Exemplo de registros de leituras dos dispositivos finais ZigBee.....	64

LISTA DE TABELAS

Tabela 1 – Análise de correlação entre as variáveis lidas e a taxa de envio de pacotes utilizando LoRa/LoRaWAN na cultura da soja	65
Tabela 2 – Análise de correlação entre as variáveis lidas e a taxa de envio de pacotes na cultura do milho	67

LISTA DE SIGLAS

ah	<i>HaLow</i>
IoT	<i>Internet of Things</i>
IMS	<i>Information Management System</i>
WSN	<i>Wireless Sensor Network</i>
M2M	<i>Machine to Machine</i>
LPWAN	<i>Low-Power Wide Area Network</i>
LR-WPAN	<i>Low-Rate Wireless Personal Area Network</i>
SENAR	<i>Serviço Nacional de Aprendizagem Rural</i>
P2P	<i>Peer to Peer</i>
IEEE	<i>Institute of Electrical and Electronic Engineers</i>
IP	<i>Internet Protocol</i>
OTAA	<i>Over-the-Air Activation</i>
ABP	<i>Activation by Personalization</i>
API	<i>Application Programming Interface</i>
PAN	<i>Personal Area Network</i>
ID	<i>Identify</i>
PAN ID	<i>PAN Addresses</i>
JV	<i>Join Verification</i>
CE	<i>Coordinator Enable</i>
CH	<i>Channel</i>
SC	<i>Scan Channels</i>
DH	<i>Destination Address High</i>
MAC	<i>Media Access Control</i>
SH	<i>Serial Number High</i>
DL	<i>Destination Address Low</i>
SL	<i>Serial Number Low</i>
NI	<i>Node Identifier</i>
SP	<i>Cyclic Sleep Period</i>
SM	<i>Sleep Mode</i>
SO	<i>Sleep Options</i>
ST	<i>Time Before Sleep</i>
W	<i>Watt</i>
v	<i>Volt (tensão elétrica alternada)</i>
A	<i>Amper</i>
Ah	<i>Ampère/Hora</i>
dBi	<i>Decibel Isotrópico</i>
MHz	<i>Mega-hertz</i>
DevAddr	<i>Device Address</i>
DevEUI	<i>Device EUI</i>
EUI	<i>Extended Unique Identifier</i>
NwkSKey	<i>Network Session Key</i>
AppSKey	<i>Application Session Key</i>
mW	<i>Megawatt</i>

SUMÁRIO

1 INTRODUÇÃO	14
2 JUSTIFICATIVA	15
3 OBJETIVOS	17
3.1 OBJETIVO GERAL	17
3.2 OBJETIVOS ESPECÍFICOS	17
4 REVISÃO DE LITERATURA	18
4.1 REDES DE SENSORES SEM FIO	18
4.2 COMUNICAÇÃO M2M	19
4.3 REDES IOT E SUAS CARACTERÍSTICAS	20
4.3.1 Restrições de uso da tecnologia IoT	21
4.3.2 Soluções IoT para agricultura	21
5 MATERIAIS E MÉTODOS	25
5.1 DEFINIÇÃO DAS TECNOLOGIAS	26
5.1.1 Levantamento das Tecnologias Passíveis de Aplicação em Cenários Agrícolas 26	
5.1.2 Estudo das Principais Tecnologias WSN	27
5.1.3 Seleção das Tecnologias a serem utilizadas como objeto da pesquisa	28
5.1.3.1 LoRa	29
5.1.3.2 ZigBee	32
5.2 DEFINIÇÃO DOS CENÁRIOS DE TESTES	36
5.3 DEFINIÇÃO DAS TOPOLOGIAS PARA CADA TECNOLOGIA	38
5.3.1 Topologia LoRa/LoRaWAN	38
5.3.1.1 Dispositivos utilizados e configuração realizada na solução LoRa/LoRaWAN 39	
5.3.1.2 Armazenamento dos dados na solução LoRa/LoRaWAN	46
5.3.2 Topologia ZigBee	48
5.3.2.1 Dispositivos e configurações utilizadas na solução ZigBee	48
5.3.2.2 Armazenamento dos dados na solução ZigBee	57
5.4 INSTALAÇÃO DOS DISPOSITIVOS FINAIS	59
6 RESULTADOS E DISCUSSÕES	60
6.4 EXECUÇÃO DOS TESTES E COLETA DOS RESULTADOS	60
6.4.1 Execução dos testes na cultura da soja	60
6.4.2 Execução dos Testes na Cultura do Milho	66
6.5 DISCUSSÕES	67
7 CONCLUSÃO	70
7.1 TRABALHOS FUTUROS	70
REFERÊNCIAS	71
APÊNDICE A – CÓDIGO ARDUINO PARA O NÓ LORA	76
APÊNDICE B – CONFIGURAÇÃO DO GATEWAY E SERVIDORES LORA/LORAWAN	79
APÊNDICE C – CÓDIGO ARDUINO PARA O DISPOSITIVO FINAL ZIGBEE	88
APÊNDICE D – CÓDIGO ARDUINO PARA O ROTEADOR ZIGBEE	90
APÊNDICE E - CÓDIGO PYTHON PARA O COORDENADOR ZIGBEE	91

1 INTRODUÇÃO

As tecnologias aplicadas às soluções de Internet das Coisas (Sigla IoT do termo em inglês: *Internet of Things*) vêm melhorando a forma como são realizadas tarefas do dia-a-dia, em meio aos crescentes desafios humanos, que compreendem o crescimento populacional, o consumo de energia, o meio ambiente e sustentabilidade. Os dispositivos envolvidos nessas soluções são caracterizados por baixas taxas de dados, consumo de energia e custo (QADIR et al., 2018).

Neste trabalho criou-se um referencial teórico sobre as principais tecnologias WSN (Redes de Sensores sem Fio ou do termo em inglês: *Wireless Sensor Network*) utilizadas em soluções IoT, sendo a partir disso, identificadas e coletadas características relevantes às aplicações de IoT com foco em cenários agrícolas, detectando pontos positivos e negativos das tecnologias estudadas, baseando-se assim, na construção de cenários práticos que contemplam seu uso no contexto agrícola, para realização de testes práticos. Os cenários previstos foram definidos considerando principais condições agrícolas da Mesorregião Oeste do Paraná, tais como calendário de cultivo, cultivares e extensão das áreas cultivadas.

Buscou-se em cada cenário, avaliar por meio da execução de testes práticos o desempenho de parâmetros de transmissão em soluções IoT, visando identificar quais as tecnologias mais apropriadas à cada situação apresentada nos cenários previstos.

2 JUSTIFICATIVA

Teixeira e Almeida (2017) colocam o setor primário da economia, que envolve atividades como a agricultura, em constante crescimento e responsável por fornecer matéria prima para a indústria, justificando a necessidade de uso de tecnologia, visto que, como os autores apontam, a agricultura representa a evolução mais relevante do setor.

Segundo Da Rosa (2017), produtores estão próximos a realidades em que aliar as tecnologias já desenvolvidas às práticas agrícolas intensivas são inevitáveis para que se busque o melhor uso dos recursos naturais, e conseqüentemente o aumento da produtividade e lucratividade das culturas. Como parte dessas tecnologias o autor cita o uso de dispositivos IoT, descrevendo como uma rede que tende a ser onipresente e global, capaz de coletar, processar e permitir que sejam realizadas análises de dados gerados por sensores presentes, que se integram por meio de redes de comunicação.

Conforme apresentado pelo Programa Agricultura de Precisão do SENAR (2017), a redução dos custos da microeletrônica e das telecomunicações, tornaram viáveis economicamente alternativas de IoT, o que deve contribuir para o aparecimento de soluções focadas no agronegócio. Ainda como parte do trabalho apresentado, o programa expõe a visão de que a infraestrutura de conectividade é um dos principais desafios para a difusão da IoT na agropecuária, não só no cenário brasileiro, como também no cenário mundial.

Segundo Teixeira e Almeida (2017), as WSNs são alternativas para a comunicação entre os equipamentos e um nó central de comunicação (*gateway*), entretanto, sua aplicação em cultivos abertos enfrenta dificuldades devido à baixa infraestrutura da tecnologia da informação e a limitação de energia. Portanto, os autores citam a importância da escolha de forma correta da rede de comunicação aplicada nos cenários agrícolas de campo aberto, bem como os protocolos de comunicação a serem utilizados por essas redes, pois possuem características distintas que os tornam adaptáveis à cada aplicação, representando grande importância quando aplicados de forma correta.

Jardim (2018) aponta que, a combinação de tecnologias pode representar um aumento de até 67% no rendimento das lavouras, entretanto, problemas de

conectividade no campo, devido principalmente, à limitação da cobertura de telefonia celular e *internet*, são gargalos para os avanços da IoT no setor. Em complemento, Seixas e Contini (2017) lembram que a IoT é altamente dependente dos sistemas e serviços de telecomunicações, devendo-se considerar as características, qualidades, velocidades e confiabilidade das redes para verificar o pleno uso do ambiente de IoT.

Neste sentido, verifica-se a importância em se estudar as características das tecnologias e protocolos de comunicação Máquina à Máquina (M2M do termo em inglês: *Machine to Machine*), e sua aplicação em diferentes cenários agrícolas, visando a contribuição nas soluções IoT e conseqüentemente na automação de atividades no setor agrícola.

3 OBJETIVOS

3.1 OBJETIVO GERAL

Avaliar tecnologias e protocolos de comunicação WSN utilizados na comunicação M2M em diferentes cenários de automatização de atividades agrícolas, por meio da utilização destas tecnologias e protocolos em testes práticos aplicados em cenários reais de cultivo de grãos em propriedades na Mesorregião Oeste do Paraná.

3.2 OBJETIVOS ESPECÍFICOS

- a) Determinar por meio de pesquisas bibliográficas as tecnologias WSN aplicáveis em soluções IoT para a agricultura, mais especificamente no que tange o cultivo de grãos na Mesorregião Oeste do Paraná.
- b) Avaliar nos cenários propostos, por meio da reprodução de testes práticos, as tecnologias e seus protocolos de comunicação, a fim de compará-los e recomendar seu uso ou não em certas condições no contexto de IoT para o cultivo de grãos na Mesorregião Oeste do Paraná.

4 REVISÃO DE LITERATURA

4.1 REDES DE SENSORES SEM FIO

Uma rede de sensores sem fio é formada por pequenos sistemas de sensores, chamados nós, implantados normalmente a curtas distâncias uns dos outros e que coletam, por meio de sensores, dados do ambiente, tais como: temperatura, pressão, umidade do ar, composição do solo, umidade do solo, movimento, velocidade, níveis de ruído, presença ou ausência de certos tipos de objetos, entre outros (ALAHÍ et al., 2018; AKYILDIZ et al., 2002; JIBER, HARROUD e KARMOUCH, 2011), e podem ser aplicadas à diversas áreas, como aplicações militares, saúde, agricultura e segurança (AKYILDIZ et al., 2002).

O funcionamento de uma WSN se dá por meio da comunicação sem fio dos nós com os *gateways*, necessária para a transmissão das informações coletadas; o *gateway* por sua vez pode processar essas informações ou enviá-las à um servidor (geralmente em ambiente de nuvem), chamado de Sistema de Gerenciamento de Informação (IMS do termo em inglês: *Information Management System*) (ALAHÍ et al., 2018).

Ainda segundo Alahi et al. (2018) a estrutura de uma WSN consiste em nós, o *gateway* e o IMS, em que:

- a) Nó: é composto por um ou mais sensores, um microcontrolador, uma unidade de fonte de alimentação, e um dispositivo de comunicação sem fio.
- b) *Gateway*: que tem a função de representar o núcleo da rede de sensores sem fio, responsável por receber os dados provindos dos nós sensores, e enviá-los ao IMS. Este é, na maioria das vezes, geograficamente instalado em uma posição fixa, conectado à uma fonte de alimentação, com sistema de comunicação sem fio que recebe as informações enviadas pelos nós.
- c) IMS: é responsável pelo armazenamento, processamento e disponibilização das informações geradas pelos nós e transmitidas pelo *gateway* aos usuários finais.

Akyildiz et al. (2002) apontam o consumo de energia como uma das mais importantes restrições deste tipo de rede, enquanto Badihi Olyaei (2013) complementa os desafios nas WSNs, incluindo: escalabilidade, mobilidade, roteamento, confiabilidade, pontualidade e localização.

4.2 COMUNICAÇÃO M2M

Comunicações M2M são um paradigma de comunicação, onde os dispositivos podem se comunicar uns com os outros sem a intervenção humana (XIA e YANG, 2016). Holler et al. (2014) complementam a descrição como as soluções que permitem a comunicação entre dois dispositivos do mesmo tipo e uma aplicação específica, geralmente implantados para obtenção de ganhos de produtividade, redução de custos e aumento de segurança, embora geralmente não permitam o amplo compartilhamento de dados ou a conexão dos dispositivos diretamente à Internet. A IoT corresponde à interconexão e comunicação entre estes diferentes dispositivos (XIA e YANG, 2016).

Quanto à aplicabilidade desse tipo de comunicação para soluções de IoT, Chen et al. (2016) consideram a comunicação M2M capaz de aumentar a eficiência e melhorar a produtividade, reduzindo custos e podendo ser aplicada e conectada a diferentes infraestruturas, trazendo resultados importantes e de forma eficiente, desempenhando um papel importante nas comunicações de IoT's. Em complemento a isto, segundo Montori et al. (2018) as tecnologias de comunicação M2M são importantes para permitir a transmissão de dados sem fio entre os dispositivos IoT e o *gateway*, e deste para um repositório remoto na Internet.

Badihi Olyaei (2013) aponta as redes de sensores sem fio como uma das principais aplicações do sistema M2M. Para Bellalta et al. (2016), o paradigma M2M está ampliando o escopo do conceito de WSN, pois permite que dispositivos se comuniquem e troquem informações, sem a interferência humana, o que nos faz entender que as comunicações M2M enfrentam a maioria dos desafios técnicos das WSNs.

4.3 REDES IOT E SUAS CARACTERÍSTICAS

Segundo Tzounis et al. (2017) o termo “Internet das Coisas” foi citado pela primeira vez pelo visionário britânico Kevin Ashton, em 1999. Čolaković e Hadžialić (2018) apoiados por Tzounis et al. (2017), descrevem IoT como um paradigma baseado na integração de uma variedade de processos envolvendo sensores, ferramentas e equipamentos cotidianos, denominados objetos ou “coisas”, aprimorados pelo poder da computação e pelos recursos de rede, podendo interagir entre si e desempenhar um papel importante na solução dos mais variados problemas. Tzounis et al. (2017) ainda apontam a agricultura como sendo um dos setores que deve ser altamente influenciado pelos avanços no domínio da IoT.

Khanna e Kaur (2019) apresentam em seu estudo diversas áreas de aplicação de soluções IoT, dentre as quais está contida a agricultura, conforme Figura 1.



Figura 1 – Áreas de aplicação de IoT
Fonte: KHANNA e KAUR et al. (2019).

4.3.1 Restrições de uso da tecnologia IoT

Conforme relatado por Rawat, Singh e Bonnin (2016), nos últimos anos tem ocorrido uma evolução do uso de dispositivos IoT, apesar de que, problemas como heterogeneidade e interoperabilidade; disponibilidade de rede e comunicação flexíveis, dinâmicas e eficientes; e eficiência energética, têm dificultado seu uso em certos cenários de aplicação. Haxhibeqiri et al. (2018) expõem a necessidade de que as aplicações de IoT têm se apresentado com aspectos tecnológicos cada vez mais apropriados e destinados à comunicação sem fio em longas distâncias, com baixo consumo de energia, disponibilizando dispositivos finais de baixo custo e baixa complexidade.

4.3.2 Soluções IoT para agricultura

Sundmaeker et al. (2016) lembram que nas últimas décadas houve uma expansão significativa da agricultura, caracterizada pela extensão de áreas, aumento da produção e uso de tecnologia, oferecendo inclusive novos modelos de negócio com o surgimento de potenciais soluções de IoT. Os mesmos autores se referem ao conceito de *Smart Farming* (do termo em português: Agricultura Inteligente) como um processo baseado em tarefas de gerenciamento que não se limitam apenas ao local, mas também nos dados, aprimorados pelo conhecimento do contexto e da situação, acionados por eventos em tempo real, utilizando de recursos que geralmente incluem assistência inteligente na implementação, manutenção e uso da tecnologia. Ahmed, De e Hussain (2018) apontam a IoT como responsável por proporcionar uma nova dimensão na área da agricultura inteligente, pela qual é possível conectar propriedades agrícolas de maneira eficiente, auxiliando de várias formas os agricultores, propiciando a eles a obtenção de dados abundantes de temperatura, fertilizante usado, volume de água utilizado, entre outros, por meio da implantação de sensores nas áreas cultivadas e nos maquinários utilizados.

Teixeira e Almeida (2017) indicam algumas características relevantes para a escolha da solução IoT quando se trata de aplicação no setor agrícola:

- a) Mobilidade dos nós sensores devido às possíveis irregularidades no terreno;
- b) Escalabilidade, de maneira a permitir a reconfiguração do sistema a cada aplicação;
- c) Longo alcance, devido às vastas áreas de cultivo a serem cobertas;
- d) Baixo consumo de energia, dado a falta de estrutura para alimentação;
- e) Baixa atenuação do sinal, devido à possibilidade de presença de obstáculos como vegetação e umidade.

Em seu trabalho, Ahmed, De e Hussain (2018) propõem uma solução de arquitetura de rede IoT para conectar áreas rurais, com foco no baixo custo e melhor qualidade. De acordo com essa proposta e conforme apresentado na Figura 2, a arquitetura combina múltiplas redes para garantir escalabilidade e larga área de cobertura. A rede de sensores sem fio baseada em 6LoWPAN é um conjunto de nós sensores utilizados em cenários agrícolas para detecção e acionamento de diversas operações. Um nó sensor baseado em 6LoWPAN é um pequeno dispositivo acessado pelo nó *gateway* através de endereçamento IPV6, caracterizado por baixa capacidade de armazenamento, baixa capacidade de processamento computacional e baixo consumo de energia. O *gateway* de borda 6LoWPAN faz uso do protocolo 802.15.4 utilizado por redes sem fio pessoais de baixas taxas de transmissão, e por sua vez é responsável por coletar as informações dos sensores e encaminhar para um servidor em nuvem IoT, ao mesmo tempo que também carrega os comandos disponíveis de um usuário remoto ou de um servidor de automação. A partir do *gateway* de borda, a comunicação passa a ser realizada por links WiFi até um *gateway* centralizado que por sua vez entrega os dados para a nuvem.

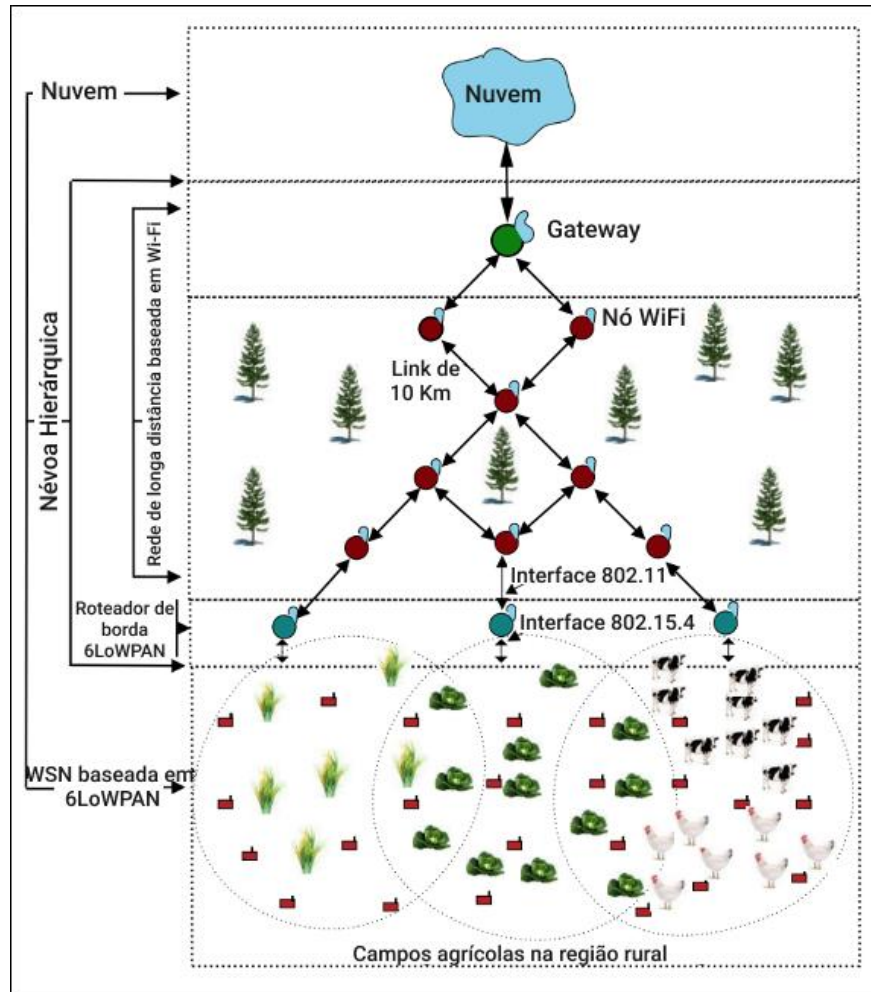


Figura 2 – Arquitetura IoT proposta para conectar áreas rurais.
Fonte: AHMED, DE e HUSSAIN (2018).

A fim de sintetizar os resultados encontrados em seu trabalho, Talavera et al. (2017) propuseram uma arquitetura para aplicações agroindustriais e ambientais, composta de 4 camadas principais: física, comunicação, serviços e aplicação. A arquitetura pode ser vista na Figura 3, na qual é possível observar na camada física uma rede WSN, composta de diversos sensores realizando o sensoriamento e controle de uma área agrícola, na camada de comunicação algumas tecnologias M2M enviando dados desta rede ao *gateway* que por sua vez disponibiliza as informações na nuvem, na camada de serviço diversos serviços atuando sobre a camada de comunicação para provimento e disponibilização dos dados pela camada de aplicação, que por sua vez, expõem ferramentas para acesso do usuário final aos dados coletados e transmitidos pelas camadas anteriores.

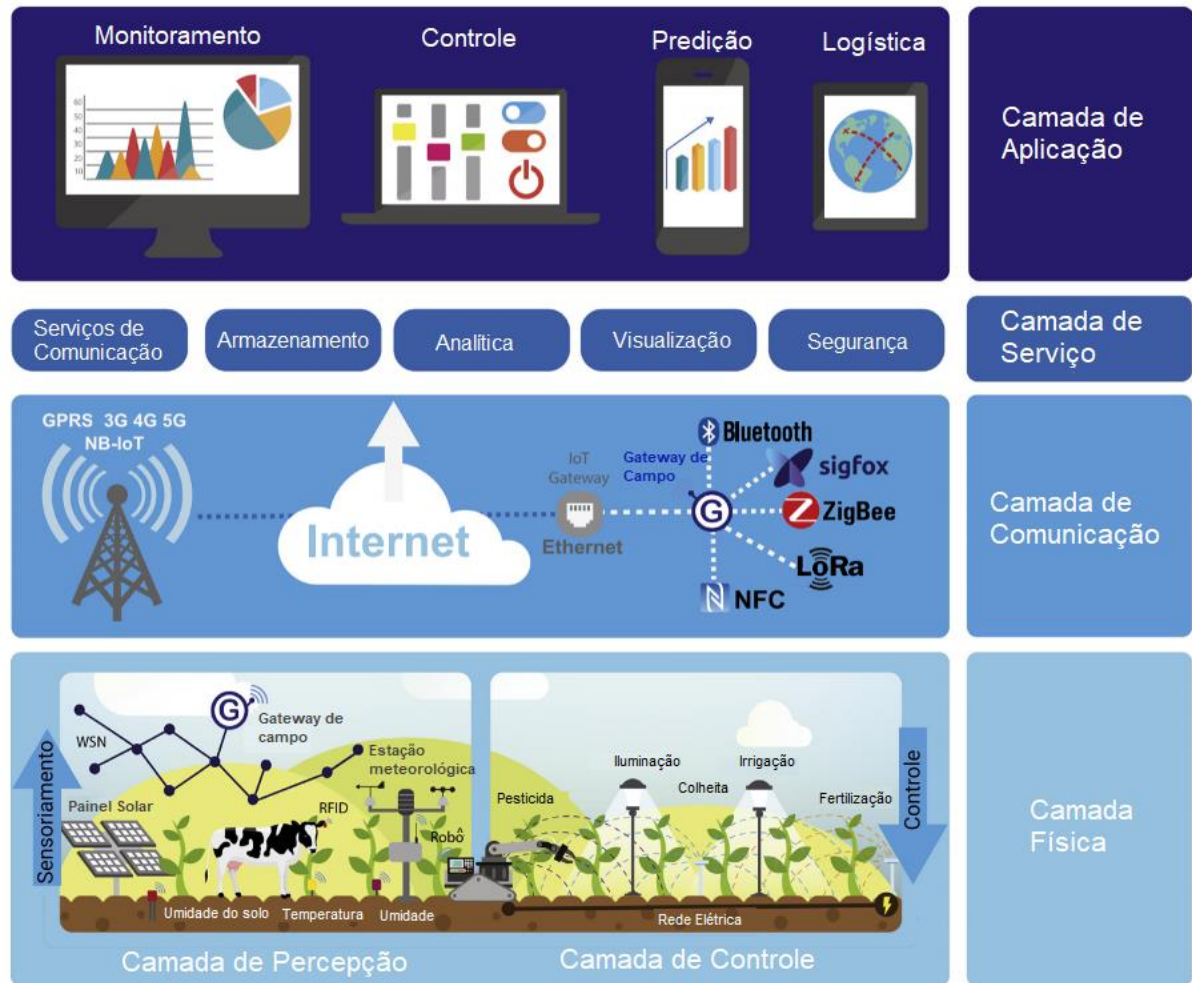


Figura 3 – Arquitetura IoT proposta para aplicações agroindustriais e ambientais.
 Fonte: TALAVERA et al. (2017).

5 MATERIAIS E MÉTODOS

Como ponto de partida para o alcance dos objetivos do projeto, por meio de obtenção de um referencial teórico, foram abordadas as principais tecnologias WSN utilizadas em soluções IoT. Para a elaboração deste referencial teórico foram consideradas as palavras-chave “Wireless Sensor Network”, “Internet of Things” e “Machine to Machine” com critério de combinação “OR”, e suas respectivas correspondências no idioma português: “Rede de Sensores sem Fio”, “Internet das Coisas” e “Máquina à Máquina” também utilizando critério “OR” para a combinação, nas bases de pesquisa Google Acadêmico, Science Direct, Periódicos CAPES e Scopus. Foram considerados resultados no idioma inglês, português e espanhol, por serem idiomas de domínio dos autores do trabalho, e com data de publicação de no máximo 10 anos a partir do ano de elaboração deste trabalho.

O mesmo referencial teórico elaborado anteriormente serviu para verificar as principais aplicações das tecnologias estudadas, visando identificar aquelas que durante estudos anteriores foram testadas ou propostas em situações agrícolas.

Por meio da etapa anterior, foram selecionadas apenas as tecnologias passíveis de aplicação em cenários agrícolas e para estas, também através de estudos bibliográficos foram identificadas e estruturadas as características relevantes aos cenários agrícolas. Para esse estudo bibliográfico, por sua vez, foram consideradas as palavras-chave “ZigBee”, “LoRa”, “LoRaWAN”, “WiFi (ah)”, “WiFi HaLow”, “NB-IoT”, “Telensa” e “SigFox” combinadas cada uma delas através de “AND” com as seguintes palavras-chave “Internet of Things”, “Smart Agriculture” e “Precision Agriculture” e suas respectivas correspondências no idioma português: “Internet das Coisas”, “Agricultura Inteligente” e “Agricultura de Precisão”. Os demais critérios para pesquisa e seleção foram os mesmos já utilizados na pesquisa anterior.

Tendo como fonte o quadro de características construído nas etapas anteriores, foram identificados os pontos positivos e negativos de cada tecnologia com foco em soluções agrícolas, visando apoiar a seleção daquelas que seriam na sequência elencadas como foco do estudo.

Após a avaliação dos pontos positivos e negativos, baseada nas características levantadas e na pré-seleção de tecnologias aplicáveis à cenários agrícolas, houve a seleção das tecnologias passíveis de aplicação na região oeste do estado do Paraná, no cenário agrícola. Para a seleção das tecnologias foram também

consideradas duas situações determinantes: tecnologias de uso gratuito e disponíveis no mercado nacional.

Para que as tecnologias pudessem ser testadas em cenários reais, houve a necessidade de elencar variáveis que posteriormente compuseram esses cenários, e na sequência a sua construção propriamente dita. O elenco das variáveis se deu através de fatores regionais de clima, cultivo e características conhecidas e com presença significativa nos cenários agrícolas.

Tendo os cenários definidos e as topologias definidas, iniciou-se a fase de execução dos testes práticos e conseqüentemente a coleta dos resultados. Por fim, de posse dos resultados dos testes foi possível desenvolver as conclusões para o presente trabalho.

5.1 DEFINIÇÃO DAS TECNOLOGIAS

5.1.1 Levantamento das Tecnologias Passíveis de Aplicação em Cenários Agrícolas

Com o intuito de auxiliar na seleção das tecnologias foco do estudo, com base em consultas bibliográficas (AHMED, RAHMAN e HUSSAIN (2016); ALAHI et al. (2018); JAWAD et al. (2018); BALMOS et al. (2016); BELLALTA et al. (2016); FENG, YAN e LIU (2019); QADIR et al. (2018); KHANNA e KAUR (2019)), elaborou-se o Quadro 1, realizando o cruzamento das tecnologias WSN e suas respectivas áreas de aplicação no contexto de soluções IoT.

Áreas de aplicação/ Tecnologias	WiFi (ah)	NB-IoT	LoRa	BLE	ZigBee	Telensa	RPMA	SigFox	WiMAX	6LoWPAN
Agricultura	X	X	X		X	X		X		
Indústria	X	X	X				X			
Logística		X					X			
Cidades Inteligentes	X	X	X			X		X		
Casas Inteligentes	X	X					X			X
Saúde		X	X	X		X	X			
Monitoramento ambiental			X		X					
Redes elétricas inteligentes		X	X				X			

Quadro 1 - Cenários de aplicação das principais tecnologias WSN.

Fonte: Autoria Própria (2021).

5.1.2 Estudo das Principais Tecnologias WSN

Com base no Quadro 1 foi possível verificar a aplicabilidade de 6 (seis) tecnologias WSN em cenários agrícolas. Utilizando de consultas em fontes bibliográficas (AUST e ITO (2012); OJHA, MISRA e RAGHUWANSHI (2015); FENG, YAN e LIU (2019); GENG e DONG (2017); DOS SANTOS et al. (2019); QADIR et al. (2018); BADIHI OLYAEI (2013); KEYSIGHT TECHNOLOGIES (2016); MONTORI et al. (2018); ADAME et al. (2014); KHANNA e KAUR (2019); ARAGUES et al. (2012)) e com o propósito de proporcionar a comparação destas tecnologias WSN quanto ao que tange algumas importantes características, foi elaborado o Quadro 2. As comparações compreendem aspectos como: protocolo baseado para comunicação, frequência utilizada, se o espectro é licenciado ou não, a faixa de dados máxima alcançada para envio e recebimento de dados, consumo de energia, distância máxima alcançada em áreas rurais, custo de implantação e topologia utilizada na sua arquitetura.

Parâmetros/ Tecnologias	WiFi (ah)	NB-IoT	LoRa	ZigBee	Telensa	SigFox
Padrão	IEEE 802.11ah	IEEE 802.15.4	IEEE 802.15.4g	IEEE 802.15.4	WIoTF	SigFox
Frequência	900 MHz	7-900 MHz	869/915 MHz	2.4 GHz	EU (868 MHz), US (915 MHz)	EU (868 MHz), US (902 MHz)
Espectro licenciado	Não	Sim	Não	Não	Não	Não
Faixa de dados (Limite download)	40 Mbps	250 Kbps	5 Kbps	250 Kbps	500 bps	600 bps
Faixa de dados (Limite upload)	40 Mbps	200 Kbps	5 Kbps	250 Kbps	62.5 bps	600 bps
Consumo de energia	Baixo	Baixo	Baixo	Baixo	-	Baixo
Alcance em área externa (rural)	1 km	15 km	40 km	100 m	8 km	50 km
Custo	Baixo	Baixo	-	Baixo	-	Baixo
Topologia	Single-hop	Estrela	Estrela de Estrela	Malha/ Estrela/ Árvore de cluster	Estrela/ Árvore	Estrela

Quadro 2 - Características das principais tecnologias WSN.

Fonte: Autoria Própria (2021).

Com base nas análises realizadas por meio das informações apresentadas no Quadro 2, foi elaborado o Quadro 3, cujo propósito foi apresentar as principais vantagens e desvantagens de cada tecnologia WSN considerando os desafios

enfrentados por soluções IoT para agricultura, a fim de contribuir para a seleção das tecnologias a serem utilizadas nos testes do presente estudo.

Tecnologia	Vantagens	Desvantagens
WiFi (ah)	- Melhor consumo de energia se comparado aos demais padrões de WiFi - Baixo custo	
NB-IoT	- Baixo consumo de energia - Longas distâncias - Baixo custo	
LoRa	- Baixo consumo de energia - Longas distâncias - Baixo custo	
ZigBee	- Baixo consumo de energia - Baixo custo	- Curtas distâncias
Telensa	- Longas distâncias	
SigFox	- Longas distâncias - Baixo custo - Baixo consumo de energia	

Quadro 3 - Vantagens e desvantagens das principais tecnologias WSN.

Fonte: Autoria Própria (2021).

5.1.3 Seleção das Tecnologias a serem utilizadas como objeto da pesquisa

Por meio das informações apresentadas no Quadro 1, Quadro 2 e Quadro 3, que descrevem respectivamente os principais cenários de aplicação das principais tecnologias WSN utilizadas em soluções IoT, as características destas tecnologias e suas vantagens e desvantagens, foi realizada a seleção de duas tecnologias, levando em consideração sua adequação à proposta da pesquisa e a possibilidade de aplicação em diferentes contextos agrícola:

- a) LoRa: por alcançar longas distâncias, o que pode ser imprescindível para a maioria dos cenários de testes em campo agrícola, baixo custo e baixo consumo de energia, e apesar da baixa faixa de transmissão de dados, embora atualmente as atividades de sensoriamento agrícola estejam se desenhando para pacotes maiores de dados por conta da captura e transmissão de imagens e vídeos, a maior parte dessas atividades ainda tendem a não exigir altas taxas de transmissão de pacotes de dados, além de ser recomendado para soluções agrícolas.

- b) ZigBee: por apresentar consumo baixo de energia e baixo custo, e apesar de alcançar curtas distâncias pode ser interessante para a variação dos cenários de testes, além da sua aplicação em soluções agrícolas;

Essas tecnologias serão apresentadas nas próximas seções, a fim de proporcionar melhor entendimento sobre seu funcionamento e estrutura arquitetural, necessários à implementação dos cenários de testes.

Apesar de SigFox e WiFi(ah) se mostrarem interessantes para serem incluídas no estudo, as mesmas não foram selecionadas por serem tecnologia proprietária e que exige compra de licença, e tecnologia ainda não disponível em mercado nacional na época do desenvolvimento do projeto, respectivamente.

5.1.3.1 LoRa

Implementada sob o protocolo LoRaWAN e o padrão IEEE 802.15.4g, os domínios de sua aplicação vão desde cidades inteligentes até a agricultura de precisão (ADELANTADO et al., 2017).

LoRaWAN pertence à categoria de tecnologias de comunicação sem fio chamada *Low-Power Wide Area Networks* (LPWAN, do termo em português: Redes de Área Ampla e de Baixa Potência) a qual recentemente tem ganhado um impulso significativo, principalmente da comunidade de pesquisadores (CASALS et al., 2017). LPWAN segundo Adelantado et al. (2017) combina baixa taxa de dados e modulação robusta para atingir um alcance de comunicação de vários quilômetros. Esta categoria de tecnologia, segundo Haxhibeqiri et al. (2018) possibilita que os sensores e demais dispositivos de uma aplicação de IoT, geralmente alimentados por bateria, possam ter sua vida útil prolongada e se comunicarem através de uma faixa de comunicação que pode ir de várias centenas de metros até vários quilômetros.

LoRa, que do inglês *Long Range* significa Longo Alcance, é uma tecnologia nova, desenvolvida pela empresa norte americana Semtech, patrocinada por grandes empresas como Cisco, IBM, Orange, entre outras (COSSINI, 2016). O que essas empresas possuem em comum é o interesse no desenvolvimento de redes LoRa para IoT.

A Figura 4 representa uma arquitetura IoT usando a tecnologia LoRa e o protocolo LoRaWAN para comunicação entre o *gateway* e os nós, onde cada nó é

composto por sensores e um módulo LoRa. As redes LoRa/LoRaWAN são compostas de dois elementos: pelo menos um nó e pelo menos um *gateway*, descritos a seguir:

- a) Os nós coletam e recebem dados do *gateway* por meio de *links* sem fio sob o protocolo LoRaWAN. Os nós não se comunicam entre si, nós comunicam-se apenas com *gateways*, podem receber (*downlink*) e enviar pacotes (*uplink*). Essa comunicação é realizada sob o protocolo LoRaWAN.
- b) O *gateway* por sua vez envia estes dados recebidos dos nós para a nuvem através de um *link* IP, de onde os servidores de aplicação os consomem e disponibilizam para o usuário final.

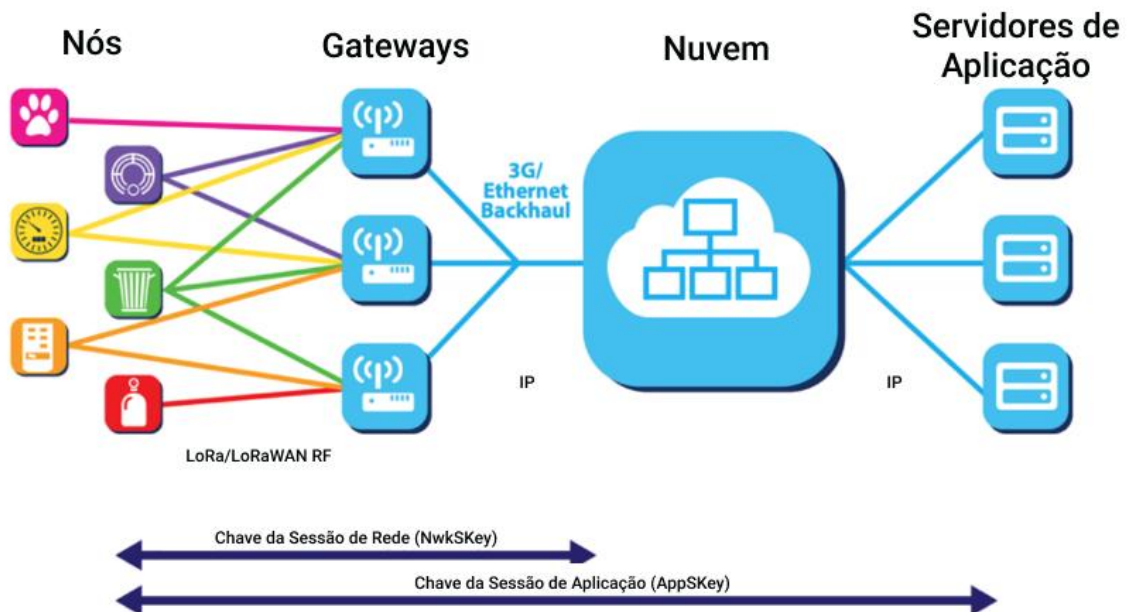


Figura 4 – Arquitetura de rede para projetos LoRaWAN.
Fonte: LORA ALLIANCE (2017).

Os *end devices* devem ser configurados para implementar uma das seguintes classes descritas (LORA ALLIANCE, 2020):

- a) Classe A: Permite a comunicação bidirecional dos dispositivos finais, onde cada transmissão disparada por um *end device* é seguida de duas janelas de recepção curtas. Essa classe representa os *end devices* de menor consumo de energia, entretanto, é necessário se atentar para a sua restrição à cenários onde é requerida a comunicação de *downlink* do servidor apenas após o envio de uma transmissão de *uplink*. Essa é a classe padrão para os dispositivos finais.

- b) Classe B: Permite aos dispositivos finais o recebimento de mais janelas de recebimento de pacotes. Além das janelas de recebimento já previstas para a Classe A, os dispositivos Classe B abrem janelas de recepção extras em horários programados. A programação dos horários é realizada através do recebimento de um pacote proveniente do *gateway*, o que permite ao servidor de rede saber quando o dispositivo final está escutando.
- c) Classe C: permitem janelas de recepção quase que de maneira contínua, fechadas apenas durante a transmissão de um pacote. Conseqüentemente, esses dispositivos são os que mais consomem energia, embora apresentem a menor latência para comunicação entre servidores e dispositivos finais.

Ainda conforme as especificações do protocolo LoRaWAN (LORA ALLIANCE, 2020), para participar de uma rede LoRaWAN cada dispositivo final precisará ser personalizado e ativado. Essa ativação pode ser realizada de duas maneiras: via Ativação pelo Ar (Sigla OTAA do termo em inglês: *Over-the-Air Activation*), quando o dispositivo é implantado ou redefinido, ou via Ativação por Personalização (Sigla ABP do termo em inglês: *Activation by Personalization*), em que as duas etapas de personalização e ativação são realizadas em uma única etapa. Após ativado algumas informações são armazenadas no dispositivo: um endereço do dispositivo (DevAddr), uma chave de sessão de rede (NwkSKey) e uma chave de sessão de aplicação (AppSKey):

- a) DevAddr: consiste em 32 bits que identificam o dispositivo final na rede.
- b) NwkSKey: é uma chave de sessão de rede específica para o dispositivo final, utilizada pelo servidor de rede e pelo dispositivo final para verificar a integridade dos dados enviados e recebidos.
- c) AppSKey: é uma chave de sessão de aplicação específica para o dispositivo final, utilizada pelo servidor de aplicação e pelo dispositivo final para criptografar e descriptografar os pacotes enviados e recebidos.

Além das chaves acima especificadas, cada dispositivo final deve possuir um identificador único, chamado de DevEUI (LORA ALLIANCE, 2020).

No lado do servidor, utilizando a solução proposta pela ChirpStack (CHIRPSTACK, 2021), a arquitetura é composta de alguns importantes componentes, como mostra a Figura 5, cujos principais são:

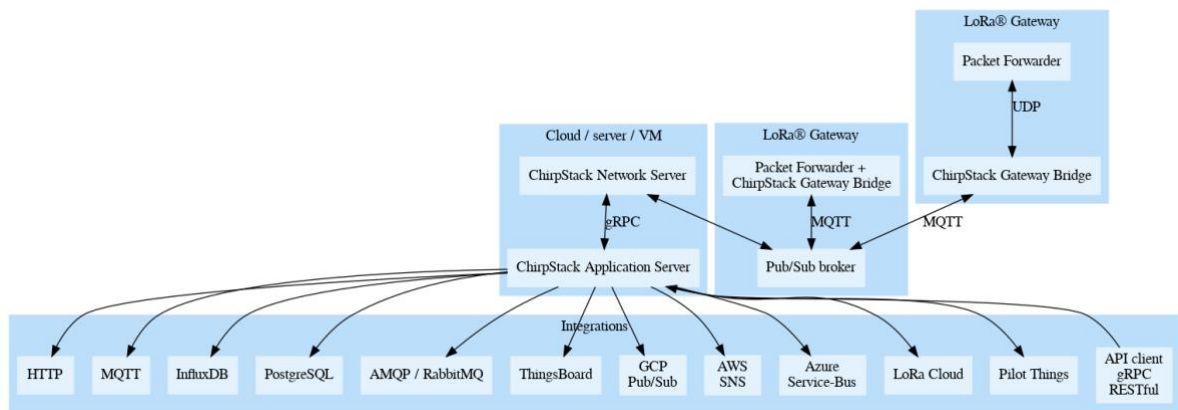


Figura 5 – Arquitetura ChirpStack LoRaWAN.
Fonte: ChirpStack (2021).

- a) *Gateway*: o *gateway* LoRa escuta geralmente 8 canais simultaneamente e encaminha os pacotes recebidos, através do *Packet Forwarder*, para cada servidor de rede LoRaWAN.
- b) *Gateway Bridge*: posicionado entre o *Packet Forwarder* e o MQTT *broker*, transforma os pacotes recebidos em um formato compatível aos demais componentes Chirpstack, além de fornecer integrações com plataformas de nuvem.
- c) *Network Server*: responsável por gerenciar a rede, tendo conhecimento sobre todas as ativações de dispositivos, eliminando a duplicidade de dados, gerenciando a fila de envio de dados proveniente dos servidores de aplicação, entre outras funções.
- d) *Application Server*: fornece uma interface *web* e APIs para o gerenciamento de usuários, organizações, aplicativos, *gateways* e dispositivos finais. É responsável por encaminhar os dados recebidos para as integrações configuradas.

5.1.3.2 ZigBee

Segundo relatado por Zulkifli e Noor (2017), ZigBee pode suportar um largo número de nós à um baixo custo. Baseado no padrão IEEE 802.15.4, o qual define as camadas física e MAC, enquanto as camadas de rede e aplicação são definidas pelo ZigBee, proporcionando menor consumo de energia e maior vida útil da bateria.

Conforme documentação disponibilizada pela Digi (2020), fabricante de componentes ZigBee, uma solução ZigBee pode conter até três diferentes tipos de dispositivos:

- a) Coordenador: obrigatoriamente uma rede ZigBee deve conter um e apenas um dispositivo coordenador. Ele é responsável por iniciar a rede, selecionando o canal e o identificador da rede (PAN ID). Também é responsável por controlar os pacotes enquanto os *end devices* estão dormindo, além de outros aspectos como segurança e manutenção da rede. Por desempenhar as responsabilidades citadas anteriormente esse dispositivo não pode dormir, devendo permanecer ligado o tempo todo. Esse dispositivo comumente recebe os pacotes dos dispositivos finais e roteadores e os envia para a nuvem.
- b) Roteador: é um elemento opcional em uma rede ZigBee, sendo utilizado desempenhando a função de repetidor de sinal, recebendo os pacotes dos dispositivos finais e os encaminhando para o coordenador, e vice e versa. Sendo assim, podem ingressar em redes existentes, enviando, recebendo e roteando informações. Assim como o coordenador, ele pode ser capaz de armazenar temporariamente os pacotes enquanto os dispositivos finais estão dormindo, além de permitir que outros roteadores ou dispositivos finais ingressem na rede. Esse tipo de dispositivo deve estar o tempo todo ligado, não podendo dormir. Uma rede ZigBee pode ter nenhum ou muitos roteadores.
- c) Dispositivos finais: também conhecidos como nós sensores, representam uma versão reduzida dos roteadores, podendo ingressar em redes ZigBee existentes, apenas recebendo e enviando informações. Sempre necessitam de um coordenador ou um roteador para ser seu “pai” na rede. Comumente esses dispositivos são configurados para economizarem energia podendo entrar temporariamente em modo de hibernação. Uma rede ZigBee deve ter ao menos um dispositivo deste tipo, entretanto, é comum que haja vários em uma mesma rede.

Ainda segundo Zulkifli e Noor (2017), apesar das soluções Zigbee comumente utilizarem a topologia de malha, outras duas também são providas: estrela e árvore de *cluster* (Figura 6), onde na topologia estrela cada nó sensor é diretamente conectado a um nó coordenador, sem a presença do roteador; na topologia de malha por sua vez cada nó sensor pode se comunicar com outros nós sensores, desde que estejam dentro do alcance da rede; enquanto que na topologia de árvore de *clusters*

há um único caminho de roteamento entre todos os nós sensores, centralizado no nó coordenador.

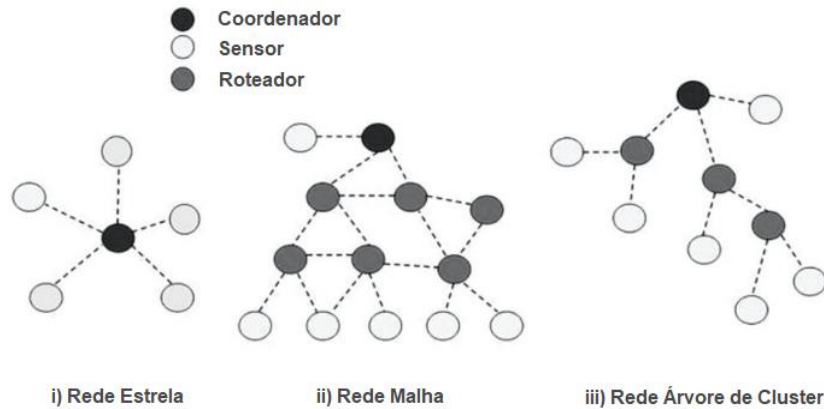


Figura 6 – Topologias ZigBee.
 Fonte: adaptado ZULKIFLI e NOOR (2017).

Jawad et al. (2018) propuseram em seu estudo uma rede de sensores para agricultura baseada em ZigBee que conta com 16 (dezesesseis) nós sensores distribuídos em uma área rural, conectados à 4 nós roteadores que por sua vez são conectados à um nó roteador principal. A solução é responsável por cobrir uma área de 50m², coletando informações tais como: temperatura e umidade do ar, e umidade do solo. A topologia pode ser observada na Figura 7.

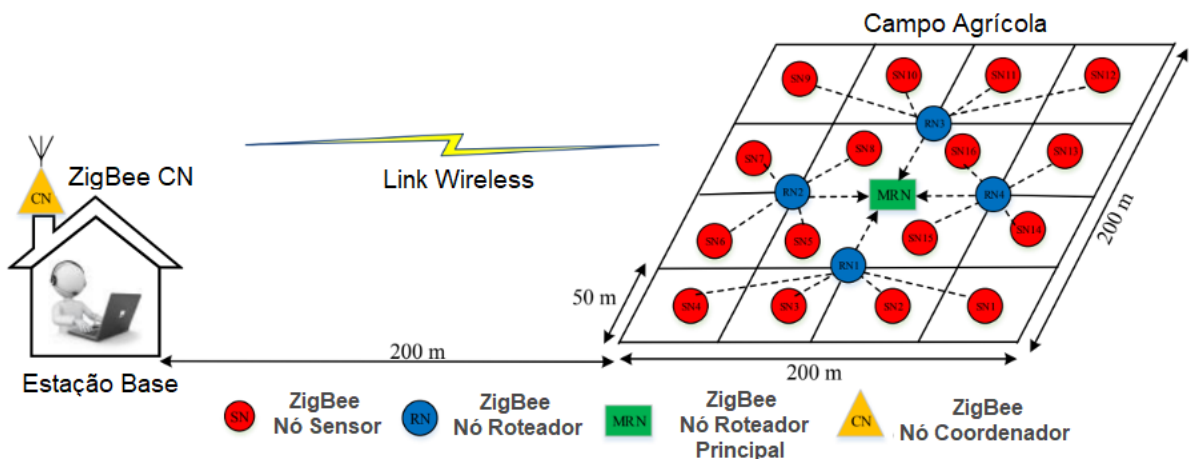


Figura 7 – Topologia de redes de sensores sem fio proposta para solução de agricultura de precisão usando ZigBee.
 Fonte: JAWAD et al. (2018).

Pertence ao grupo de tecnologias *Low-Rate Wireless Personal Area Network* (LR-WPAN, do termo em português: Rede de Área Pessoal sem Fio de Baixa Taxa), que segundo Badihi Olyaei (2013) são categorizadas por consumo de energia

extremamente baixo, baixa taxa de transferência de dados, baixa complexidade e conseqüentemente maior facilidade de instalação e gerenciamento da rede.

Badihi Olyaei (2013) em sua dissertação de mestrado, a qual objetivou o estudo e comparação do padrão IEEE 802.15.4 com outras tecnologias como WiFi e Bluetooth, conclui que apesar de não ter atingido níveis altos de taxa de transferência de dados, o que não é um problema para o objeto deste estudo, em termos de consumo de energia se demonstrou um padrão adequado.

Módulos ZigBee dispõem de dois diferentes modos de operação, que estabelecem a maneira como o dispositivo *host* se comunica com o módulo por meio da interface serial. Esses modos de operação são (DIGI, 2020):

- a) Modo transparente: (do termo em inglês: *transparent mode*) é assim chamado pois o rádio retransmite as informações recebidas exatamente como as recebe. É um modo limitado, entretanto ideal para situações simples, por ser de fácil configuração.
- b) Modo de operação API: (do termo em inglês: *Application Programming Interface operation mode*) corresponde ao modo de operação onde os dados são transmitidos em pacotes, com informações adicionais além das informações que representam os dados propriamente ditos. Permite a construção de redes maiores e é mais indicado às redes de sensores cuja necessidade é coletar dados de vários diferentes locais, controlar dispositivos remotamente ou automatizar residências.

A seguir estão elencados os principais parâmetros de configuração dos rádios ZigBee (DIGI, 2020):

- a) PAN ID: *PAN Addresses* - É o ID (identificador) da rede. Todos os rádios que pertencem à mesma rede devem compartilhar do mesmo ID.
- b) JV: *Join Verification* - Verifica se existe um coordenador no mesmo canal para ingressar na rede ou para sair caso não seja encontrado.
- c) CE: *Coordinator Enable* - Define se o rádio é um coordenador.
- d) CH: *Channel* - Define o canal de operação do rádio. Para que os dispositivos possam se comunicar, eles necessitam operar na mesma frequência.
- e) SC: *Scan Channels* - Define a lista de canais a serem varridos.
- f) DH: *Destination Address High* - Define o endereço de destino (parte alta) para transmitir o dado para. Corresponde aos 8 (oito) primeiros dígitos do endereço MAC do rádio destino, também chamado de *Serial Number High* (SH).

- g) DL: *Destination Address Low* - Define o endereço de destino (parte baixa) para transmitir o dado para. Corresponde aos 8 (oito) últimos dígitos do endereço MAC do rádio destino, também chamado de *Serial Number Low (SL)*.
- h) NI: *Node Identifier* - Identificação do nó, ou seja, é o nome que identifica o rádio. Uma *string* curta que permite aos usuários identificarem o módulo através de uma nomenclatura humanamente amigável.
- i) SP: *Cyclic Sleep Period* - Define a duração do tempo gasto dormindo.
- j) SM: *Sleep Mode* - Define o modo de suspensão do dispositivo.
- k) SO: *Sleep Options* - Mantém o módulo acordado durante todo o período.
- l) ST: *Time Before Sleep* - Define o tempo em que o dispositivo deve estar inativo antes de dormir. Se durante esse tempo o dispositivo enviar ou receber dados, ele não irá dormir.

5.2 DEFINIÇÃO DOS CENÁRIOS DE TESTES

Os 16 (dezesesseis) cenários para os testes, ilustrados no Quadro 4, foram traçados na Mesorregião Oeste do Paraná, visando considerar as condições regionais, tais como: clima, cultivo, calendário de cultivo, relevo e extensão da área cultivada. Sendo assim, a definição dos cenários de testes levou em consideração as seguintes variáveis:

- a) Tipo de cultivo (Soja/Milho);
- b) Distância (Faixas);
- c) Interferência (Ausência/Presença);

As culturas foram assim escolhidas por serem as de maior relevância no cultivo da Mesorregião Oeste do Paraná. Segundo dados do IBGE (2020) a produção de soja e milho no Paraná para o ano de 2018, ocupou cada um o posto de segundo maior do Brasil, com respectivamente 19.026.204 toneladas e 12.758.122 toneladas produzidas, ficando atrás apenas do Estado do Mato Grosso. Desta produção, aproximadamente 19% e 32% correspondem à Mesorregião Oeste do Paraná, composta por 50 (cinquenta) municípios.

A distância e a presença de causadores de interferência foram consideradas por apresentarem importância na latência e alcance do sinal. Como causadores de interferência foram considerados: relevo, residências, galpões, matas, árvores e outras vegetações.

Quanto ao tipo dos dados transferidos, foi considerado o envio de dados textuais por serem os predominantemente coletados em atividades de sensoriamento e agricultura de precisão envolvendo o cultivo de grãos, ou seja, compreendem captura de dados de sensores, neste caso especificamente serão utilizados sensores de luz ultravioleta e pressão, umidade e temperatura do ar.

Identificador	Descrição	Tipo de Cultivo	Distância (metros)	Interferência
#1	200m CI	Soja	200	Com
#2	200m SI	Soja	200	Sem
#3	500m CI	Soja	500	Com
#4	500m SI	Soja	500	Sem
#5	1Km CI	Soja	1000	Com
#6	1Km SI	Soja	1000	Sem
#7	3Km CI	Soja	3000	Com
#8	3Km SI	Soja	3000	Sem
#9	200m CI	Milho	200	Com
#10	200m SI	Milho	200	Sem
#11	500m CI	Milho	500	Com
#12	500m SI	Milho	500	Sem
#13	1Km CI	Milho	1000	Com
#14	1Km SI	Milho	1000	Sem
#15	3Km CI	Milho	3000	Com
#16	3Km SI	Milho	3000	Sem

Quadro 4 - Cenários de testes

Fonte: Autoria Própria (2021).

Além das variáveis citadas, durante os testes foi observada a taxa de recebimento de pacotes em função de variações nas condições climáticas. As condições climáticas da região onde os testes foram realizados permitiram cenários em condições variáveis de temperatura e umidade. Segundo dados do Instituto Agrônomo do Paraná - IAPAR (2020), no ano de 2019 as temperaturas médias

mensais para a Mesorregião Oeste do Paraná tiveram variação de 17°C à 29°C e precipitação acumulada mensal esteve variando de 0mm à 320mm. A interferência das condições climáticas sobre a transmissão nas diferentes tecnologias é uma suposição e será verificada.

5.3 DEFINIÇÃO DAS TOPOLOGIAS PARA CADA TECNOLOGIA

Para coleta de dados de cada cenário descrito no Quadro 4, foram projetadas topologias LoRa/LoRaWAN e ZigBee.

5.3.1 Topologia LoRa/LoRaWAN

A topologia LoRa/LoRaWAN conta com um *gateway* e ao menos um *end device*, também chamado de “nó”. A exemplificação da topologia LoRa/LoRaWAN projetada pode ser vista na Figura 8. A composição detalhada de cada nó e do *gateway* estão respectivamente compreendidas na Figura 9 e na Figura 11.

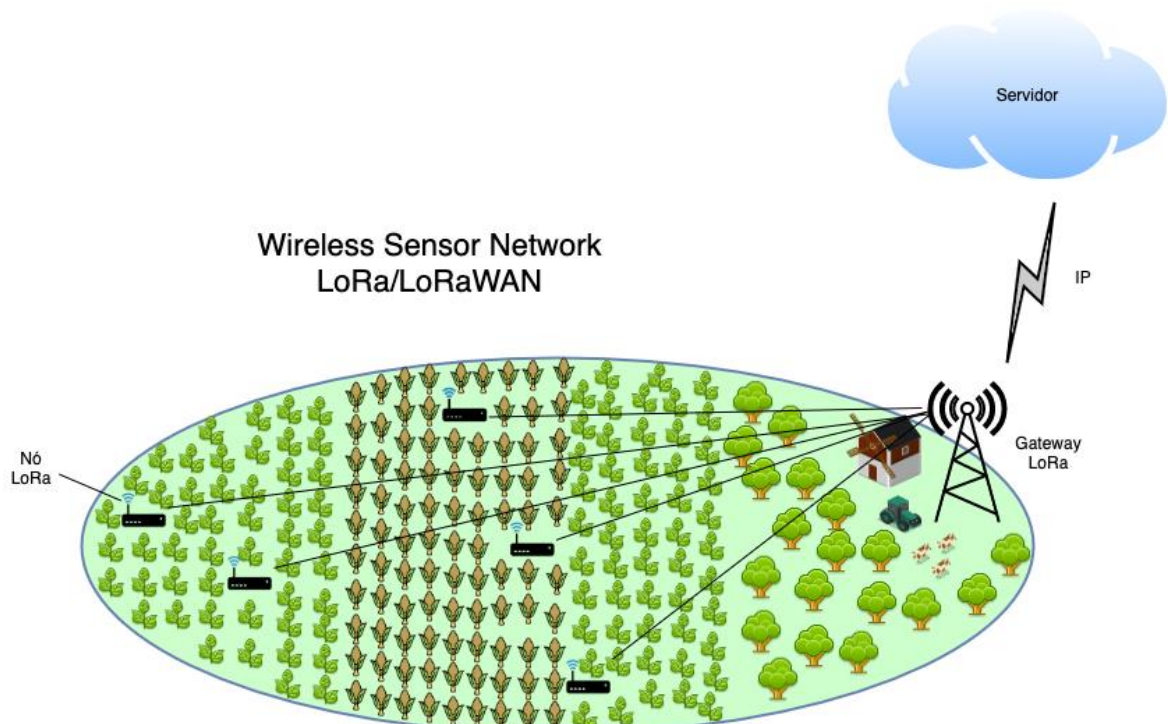


Figura 8 – Exemplificação da topologia LoRa utilizada no projeto.
Fonte: Autoria Própria (2021).

5.3.1.1 Dispositivos utilizados e configuração realizada na solução LoRa/LoRaWAN

Como pode ser observado na Figura 9, cada nó LoRa é composto de um painel solar para alimentação de uma bateria recarregável, um controlador de carga, um regulador de tensão um módulo LoRa, sensores e uma antena. Cada componente utilizado na composição do nó LoRa é abaixo descrito:

- a) Painel solar: com o objetivo de propor uma solução estacionária e tendo como conhecimento a carência de alimentação através de redes de energia elétrica adentrando a lavoura, cada nó conta com um painel solar de energia fotovoltaica de 10W, que será instalado junto aos demais componentes através de um suporte com altura regulável, prevendo e atendendo assim diferentes alturas de cultivos bem como o próprio desenvolvimento da planta.
- b) Controlador de carga: a carga não frequente proveniente do painel solar deve passar por um controlador de carga antes de atingir sua finalidade de alimentar a bateria. Para tanto, foi utilizado para cada painel solar um controlador de carga solar com carga máxima de 10A e detecção automática de voltagem 12v e 24v.
- c) Bateria recarregável: para alimentação diária de cada módulo LoRa foram utilizadas baterias recarregáveis estacionárias de 12v com capacidade de 7Ah.
- d) Regulador de tensão: sendo os módulos LoRa alimentados por corrente de 3.3v, a saída de corrente da bateria deve passar pelo controlador de carga e em seguida pelo regulador de tensão para que seja garantida uma tensão de saída de 3.3v. Para tanto, cada nó conta com um regulador de tensão com capacidade para receber até 12v e regular essa tensão para uma saída de 3.3v.
- e) Microcontrolador: contendo a lógica de coleta dos dados dos sensores, tratamento e envio através do módulo LoRa, o microcontrolador é um elemento importante da topologia. Para o projeto foi utilizado o Arduino Mega 2560.
- f) Módulo LoRa: o principal componente do nó é o módulo LoRa, responsável pela transmissão dos dados coletados pelos sensores até o *Gateway*. Neste projeto foi utilizado especificamente o módulo RHF76-052 da fabricante RisingHF.
- g) Antena: o módulo utilizado não conta com antena de fábrica, portanto, para que a transmissão dos dados do módulo para o *gateway* fosse possível, cada

módulo foi equipado com uma antena monopolo espiral omnidirecional de ~4,5dBi com frequência de transmissão de 915MHz, frequência essa utilizada pelo protocolo LoRaWAN.

- h) Sensor de temperatura, umidade e pressão: para coleta de dados de temperatura, umidade e pressão do ar cada nó foi equipado com um sensor de pressão/temperatura/umidade BME280.
- i) Sensor de luz ultravioleta: para a coleta de dados de radiação ultravioleta incidente cada nó foi equipado com um sensor de luz ultravioleta UV GUA-S12S.

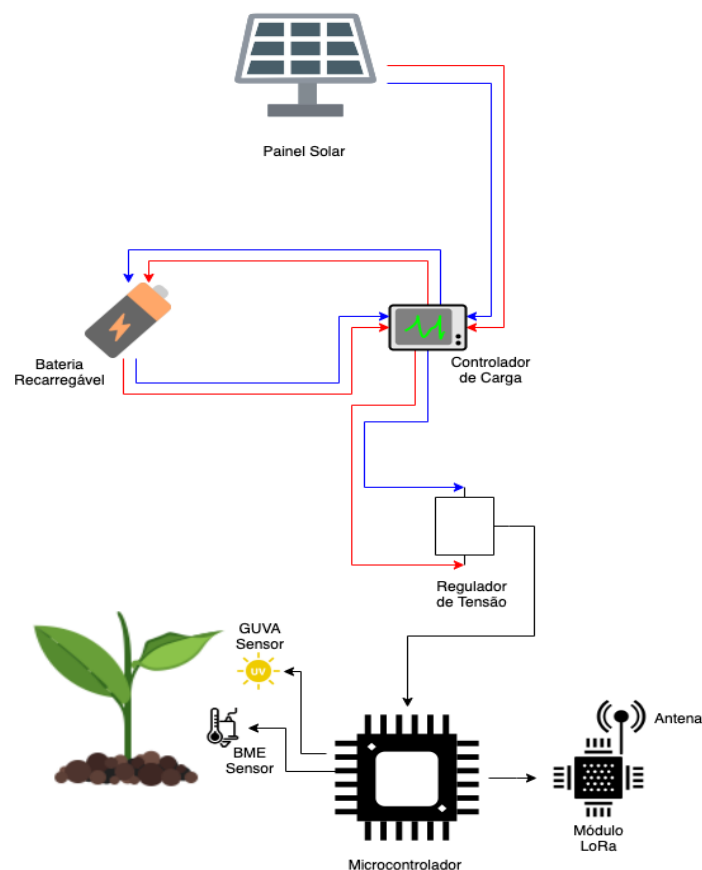


Figura 9 – Projeto do Nó LoRa.
Fonte: Autoria Própria (2021).

Os sensores GUA e BME foram considerados para que, como complemento aos cenários de testes, possam ser verificadas respectivamente a relação da intensidade de raios ultravioleta e da temperatura e umidade no alcance do sinal e envio dos pacotes.

A Figura 10 apresenta um nó montado, conforme proposto no projeto exibido na Figura 9.

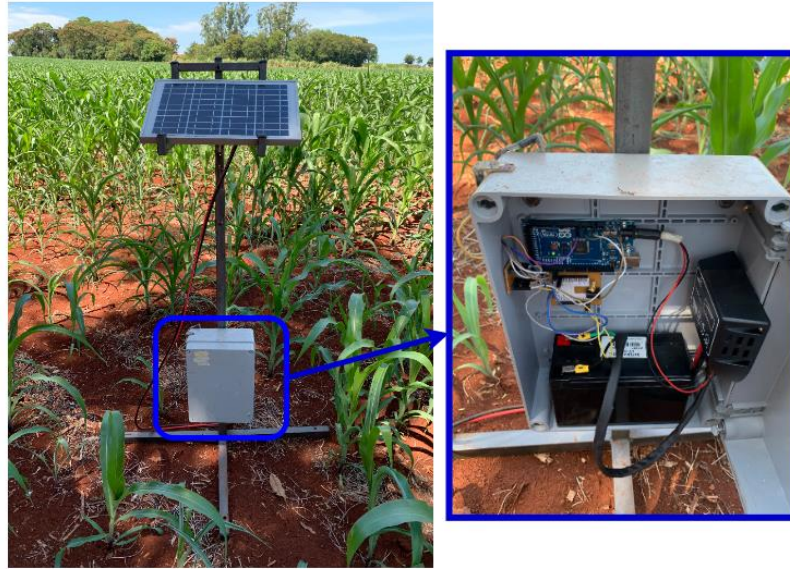


Figura 10 – Nó LoRa.
Fonte: Autoria Própria (2021).

O Quadro 5 apresenta o esquema das ligações realizadas entre o Arduino Mega 2560 e o Módulo LoRa.

Arduino	Módulo LoRa
3.3V	VCC
GND	GND
TX2 – pin 16	USART_RX
RX2 – pin 17	USART_TX

Quadro 5 – Esquema de ligações entre Arduino Mega 2560 e o Módulo LoRa
Fonte: Autoria Própria (2021).

O Quadro 6 e o Quadro 7, por sua vez apresentam respectivamente o esquema das ligações realizadas entre o Arduino Mega 2560 e os sensores GUA e BME.

Arduino	UV GUA-S12S
A0	SIG
GND	GND
VCC	3.3V

Quadro 6 – Esquema de ligações entre Arduino Mega 2560 e o sensor GUA
Fonte: Autoria Própria (2021).

Arduino	BME280
SCL	SCL
SDA	SDA
GND	GND
3.3V	VCC

Quadro 7 – Esquema de ligações entre Arduino Mega 2560 e o sensor BME
Fonte: Autoria Própria (2021).

O Apêndice A contém o código executado no microcontrolador responsável por obter os valores lidos pelos sensores, encapsular em um pacote utilizando Cayenne LPP e enviar através do módulo LoRA para o *Gateway* LoRaWAN. A seguir são apresentadas partes importantes desse código.

No Quadro 8 é possível visualizar as bibliotecas utilizadas para a execução da lógica no Arduino.

- a) LoRaWan é uma biblioteca desenvolvida pela Seeeduno, construída para executar em Arduino Nano, portanto, para o projeto precisou ser minimamente modificada para a pinagem do Arduino Mega 2560. O trecho modificado no código fonte da biblioteca LoRaWan é apresentado no Quadro 9.
- b) CayenneLPP é a biblioteca desenvolvida pela The Things Network, responsável por permitir dispositivos Arduino codificar dados utilizando *Cayenne Low Power Payload* (LPP).
- c) As duas últimas bibliotecas Adafruit são utilizadas para manipular o sensor BME280.

```
// -- BIBLIOTECA USADAS
#include <LoRaWan.h>
#include <CayenneLPP.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
```

Quadro 8 – Bibliotecas utilizadas para execução do nó LoRa no Arduino
Fonte: Autoria Própria (2021).

```
//Arduino UNO
//SoftwareSerial SerialLoRa(2, 3);

//Arduino Mega
#define SerialLoRa Serial2
```

**Quadro 9 – Código alterado no código fonte da biblioteca LoRaWan
Fonte: Autoria Própria (2021).**

O Quadro 10 apresenta a configuração do *end device* LoRa. Começando com a inicialização do dispositivo e a redefinição de suas configurações de fábrica. Em seguida, a configuração dos endereços e chaves. Lembrando que para a configuração ABP, definida nas próximas linhas do código, são indispensáveis as definições de *Device Address* (DevAddr), *Device EUI* (DevEUI), *Network Session Key* (NwkSKey) e *Application Session Key* (AppSKey). As linhas seguintes definem os canais a serem utilizados para o envio dos pacotes e o *data rate* inicial, sendo que, como pode ser visto, a configuração determina o uso de *data rate* adaptativo, ou seja, com base em informações de qualidade de sinal e tamanho dos pacotes, enviadas pelo *gateway* no recebimento dos pacotes, o *end device* ajusta seu *data rate*. Nas duas penúltimas instruções são sinalizadas as configurações referentes aos ciclos de trabalho como falsas, ou seja, o dispositivo não estará atuando em modo econômico com ciclos ativos e inativos. A última linha refere-se à configuração de potência do sinal inicial, que também é ajustada a cada envio de pacote devido à configuração de *data rate* adaptativo.

```
lora.init();
lora.setDeviceReset();

//---- setId(char *DevAddr, char *DevEUI, char *AppEUI)
lora.setId("01882f84", "73301be462247b1c", "28d9790df644928c");

//---- setKey(char *NwkSKey, char *AppSKey, char *AppKey)
lora.setKey("5f218b7f546726faa1057c601085ce3a",
"077b481ca67956d77c4c0bc16af12eb", "077b481ca67956d77c4c0bc16af12eb");

lora.setDeciveMode(LWABP);
```

```

lora.setDataRate(DR3, AU915);

lora.setChannel(0, 916.8);
lora.setChannel(1, 917.0);
lora.setChannel(2, 917.2);
lora.setChannel(3, 917.4);
lora.setChannel(4, 917.6);
lora.setChannel(5, 917.8);
lora.setChannel(6, 918.0);
lora.setChannel(7, 918.2);

lora.setReceiveWindowFirst(1, 923.9);
lora.setReceiveWindowSecond(923.3, DR8);

lora.setAdaptiveDataRate(true);

lora.setDutyCycle(false);
lora.setJoinDutyCycle(false);

lora.setPower(14);

```

**Quadro 10 – Configuração do *end device* LoRa usando a biblioteca LoRaWan.
Fonte: Autoria Própria (2021).**

Finalizando a lógica Arduino para a transmissão dos pacotes pelo *end device* LoRa, no Quadro 11 vemos a leitura da temperatura através do sensor BME280, encapsulamento dos dados utilizando Cayenne LPP e o envio do pacote via protocolo LoRaWAN, utilizando a estratégia de envio com confirmação de recebimento pelo Gateway.

```

float temperatura = bme.readTemperature();

lpp.addTemperature(1, temperatura);

result = lora.transferPacketWithConfirmed(lpp.getBuffer(), lpp.getSize());

```

**Quadro 11 – Coleta da temperatura, encapsulamento usando Cayenne LPP e envio do pacote LoRaWAN.
Fonte: Autoria Própria (2021).**

A topologia conta obrigatoriamente com pelo menos um *gateway* LoRaWAN, que irá receber os dados enviados pelos nós e transmitir a um servidor na nuvem. O *gateway* é composto por três elementos abaixo apresentados e que também podem ser observados na Figura 11:

- a) *Gateway* LoRAWAN: o projeto utilizou o modelo RHF0M301 da fabricante RisingHF.
- b) Antena: conectada ao *gateway* para o recebimento dos pacotes enviados pelos *end devices*, foi utilizada uma antena OMNI direcional de 6 dBi com frequência de transmissão de 915MHz.
- c) Raspberry Pi3: responsável por comportar a instalação dos *softwares* que garantem o funcionamento do *gateway*. Adiante veremos quais são esses *softwares* e suas responsabilidades.

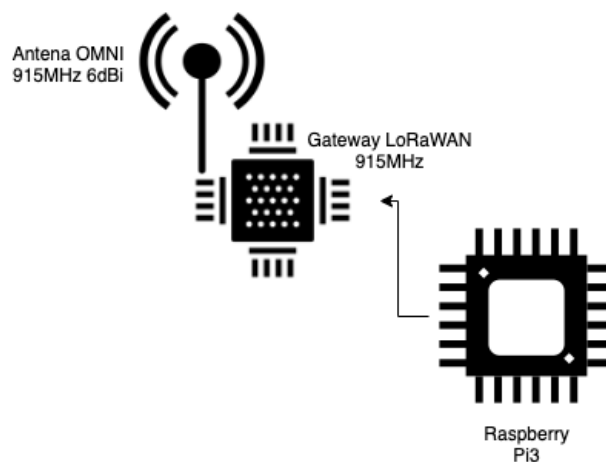


Figura 11 – Projeto do Gateway LoRaWAN.
Fonte: Aatoria Própria (2021).

A Figura 12 representa o *Gateway* montado, de acordo com os componentes previstos no esquema da Figura 11.

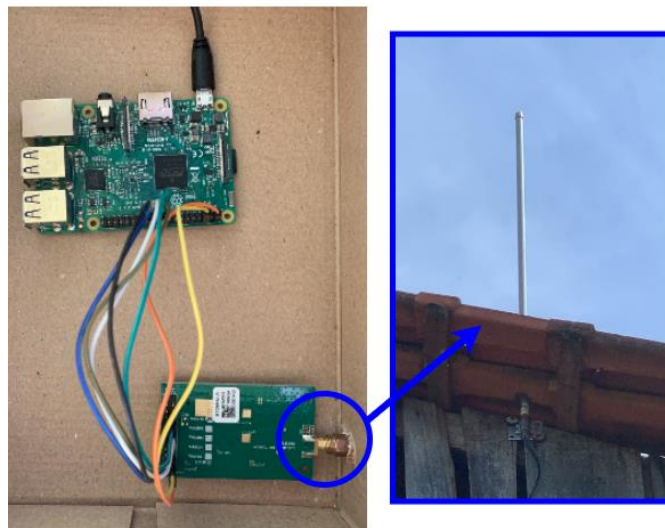


Figura 12 – Gateway LoRaWAN montado.
Fonte: Aatoria Própria (2021).

5.3.1.2 Armazenamento dos dados na solução LoRa/LoRaWAN

Para a arquitetura de *software* de comunicação do *Gateway* com os servidores de Rede e Aplicação, foi utilizada a solução da Chirpstack: a *stack* de servidores de rede ChirpStack LoRaWAN, que é um projeto que provê componentes *open-source* para redes LoRaWAN, que juntos formam uma solução pronta para uso, incluindo interface *web* amigável para o gerenciamento dos dispositivos finais e *gateways* e API para integrações. O Apêndice B disponibiliza as etapas para configuração dessa solução. A Figura 13 apresenta a interface de *dashboard* do servidor de aplicação configurado para o projeto.

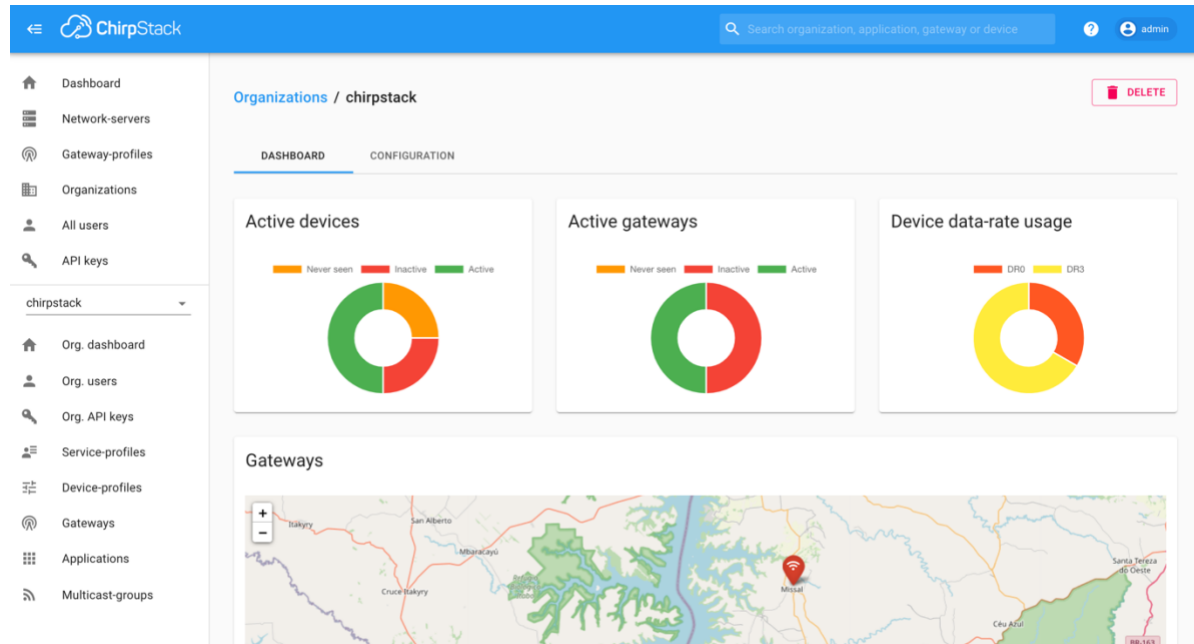


Figura 13 – Servidor de aplicação do projeto LoRaWAN.
Fonte: Autoria Própria (2021).

Os dados recebidos pelo servidor de rede foram salvos em uma base de dados InfluxDB através da integração disponibilizada na solução ChirpStack. InfluxDB é um banco de dados de série temporal construído com o propósito de coletar, armazenar, processar e visualizar métricas e eventos. Na Figura 14 podemos observar o resultado da seleção de registros da medida `device_frmpayload_data_temperatureSensor_1`, correspondente às leituras de temperatura dos sensores da rede LoRa/LoRaWAN.

```
[> select * from device_frmpayload_data_temperatureSensor_1;
name: device_frmpayload_data_temperatureSensor_1
time                application_name dev_eui                device_name f_port value
-----
1609436400000000000 Mestrado        1d7b2462e41b3073 Device04    8      28.1
1609436460000000000 Mestrado        1d7b2462e41b3073 Device04    8      28.3
1609436700000000000 Mestrado        1d7b2462e41b3073 Device04    8      29
1609437240000000000 Mestrado        1d7b2462e41b3073 Device04    8      28.7
1609437300000000000 Mestrado        1d7b2462e41b3073 Device04    8      28.8
1609437360000000000 Mestrado        1d7b2462e41b3073 Device04    8      28.8
1609437600000000000 Mestrado        1d7b2462e41b3073 Device04    8      28.8
1609437840000000000 Mestrado        1d7b2462e41b3073 Device04    8      28.6
1609438020000000000 Mestrado        1d7b2462e41b3073 Device04    8      28.7
1609438140000000000 Mestrado        1d7b2462e41b3073 Device04    8      28.9
1609438200000000000 Mestrado        1d7b2462e41b3073 Device04    8      28.9
1609438320000000000 Mestrado        1d7b2462e41b3073 Device04    8      28.7
```

Figura 14 – Seleção de valores de temperatura armazenados na base InfluxDB.
Fonte: Autoria Própria (2021).

5.3.2 Topologia ZigBee

A topologia ZigBee por sua vez conta com um coordenador, opcionalmente um ou mais roteadores e ao menos um dispositivo final. A exemplificação da topologia ZigBee projetada para o trabalho pode ser vista na Figura 15. A composição detalhada de cada dispositivo final, roteador e coordenador estão respectivamente compreendidas na Figura 16, Figura 17, e Figura 20.

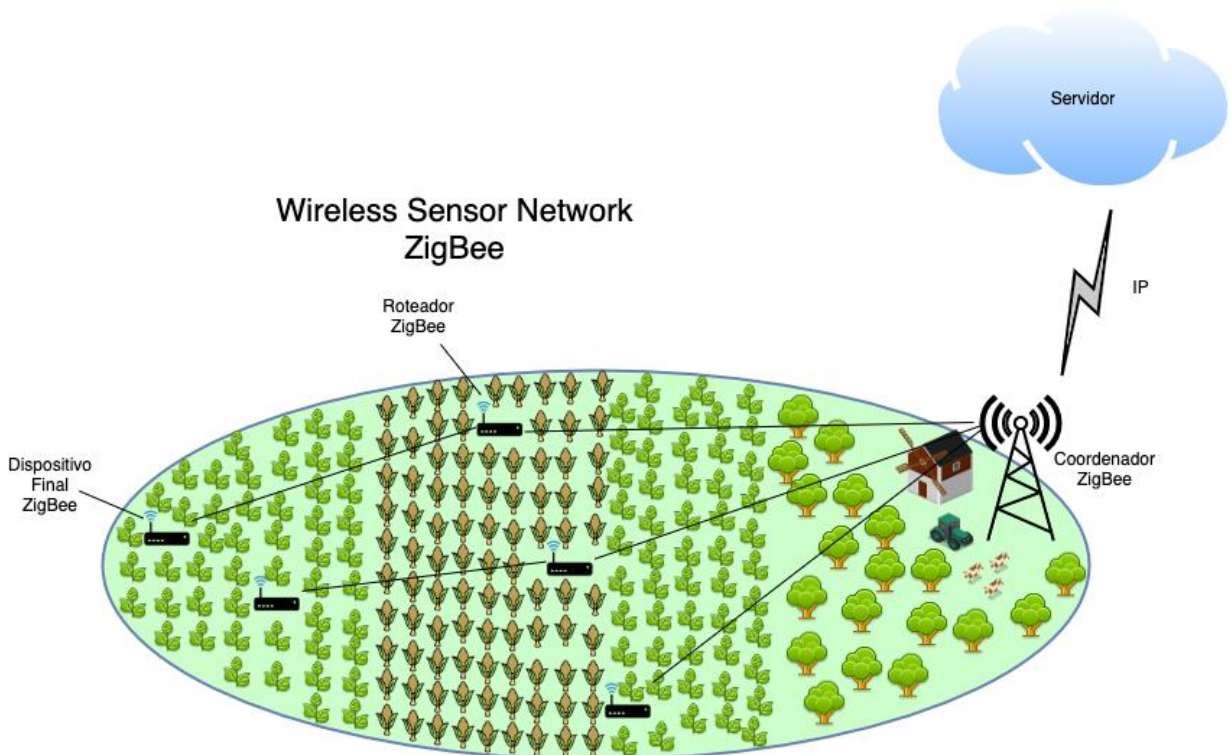


Figura 15 – Exemplificação da topologia ZigBee utilizada no projeto.
Fonte: Autoria Própria (2021).

5.3.2.1 Dispositivos e configurações utilizadas na solução ZigBee

Como pode ser observado na Figura 16, assim como na topologia LoRa/LoRaWAN, cada dispositivo final ZigBee é composto de painel solar, uma bateria recarregável, um controlador de carga, um microcontrolador, um *shield*, um Módulo ZigBee, sensores e uma antena embutida. O roteador ZigBee, apresentado na Figura 17, utiliza dos mesmos componentes utilizados pelo dispositivo final, exceto

os sensores. Cada componente utilizado na composição do dispositivo final e do roteador é abaixo descrito:

- a) Painel solar: com o objetivo de propor uma solução estacionária e tendo como conhecimento a carência de alimentação através de redes de energia elétrica adentrando a lavoura, cada dispositivo final e cada roteador conta com um painel solar de energia fotovoltaica de 10W, que será instalado junto aos demais componentes através de um suporte com altura regulável, prevendo e atendendo assim diferentes alturas de cultivares bem como o próprio desenvolvimento da planta.
- b) Controlador de carga: a carga não frequente proveniente do painel solar deve passar por um controlador de carga antes de atingir sua finalidade de alimentar a bateria. Para tanto, foi utilizado para cada painel solar um controlador de carga solar com carga máxima de 10A e detecção automática de voltagem 12v e 24v.
- c) Bateria recarregável: para alimentação diária de cada dispositivo final e roteador foram utilizadas baterias recarregáveis estacionárias de 12v com capacidade de 7Ah.
- d) *Shield* de expansão Xbee ZigBee: utilizado para encaixe do módulo ZigBee ao microcontrolador Arduino, cada dispositivo final e cada roteador possui um *shield* de expansão acoplado ao microcontrolador, com o intuito de fornecer comunicação serial e alimentação ao módulo.
- e) Microcontrolador: contendo a lógica de coleta dos dados dos sensores, tratamento e envio através do módulo ZigBee, o microcontrolador é um elemento importante da topologia, tanto para dispositivos finais como para os roteadores. Para o projeto foi utilizado o Arduino Mega 2560.
- f) Módulo ZigBee: o principal componente do dispositivo final e do roteador é o módulo ZigBee, responsável pela transmissão dos dados coletados pelos sensores até o roteador ou até o coordenador. Neste projeto foi utilizado especificamente o módulo XBee S2 S2C ZigBee da fabricante Digi, que conta com antena integrada representando potência de transmissão de 3.1mW (+5dBm).
- g) Sensor de temperatura, umidade e pressão: para coleta de dados de temperatura, umidade e pressão do ar cada dispositivo final foi equipado com um sensor de pressão/temperatura/umidade BME280.

- h) Sensor de luz ultravioleta: para a coleta de dados de radiação ultravioleta incidente cada dispositivo final foi equipado com um sensor de luz ultravioleta UV GUYA-S12S.

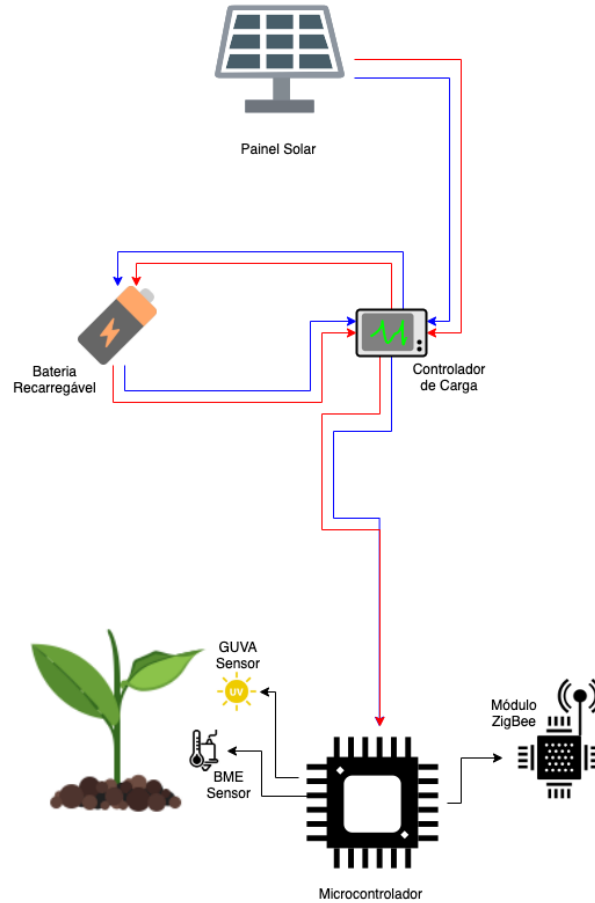


Figura 16 – Projeto do Dispositivo Final ZigBee.
Fonte: Autoria Própria (2021).

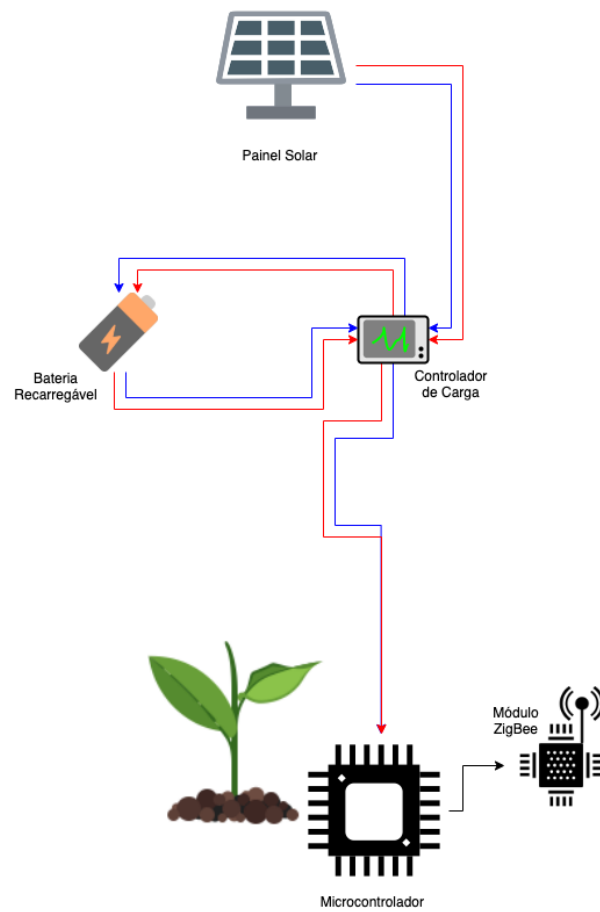


Figura 17 – Projeto do Roteador ZigBee.
Fonte: Autoria Própria (2021).

Assim como para a topologia LoRa/LoRaWAN, os sensores GUVa e BME foram considerados para que, como complemento aos cenários de testes, possam ser verificadas a relação da intensidade de raios ultravioleta e da temperatura e umidade no alcance do sinal e envio dos pacotes.

A Figura 18 e a Figura 19 ilustram respectivamente um dispositivo final e um roteador montados, conforme proposto nos projetos exibidos na Figura 16 e na Figura 17.

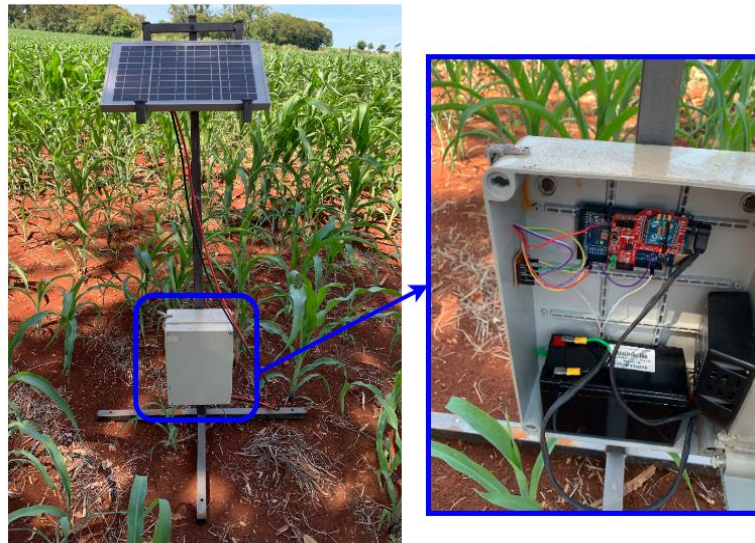


Figura 18 – Dispositivo final ZigBee montado.
Fonte: Autoria Própria (2021).

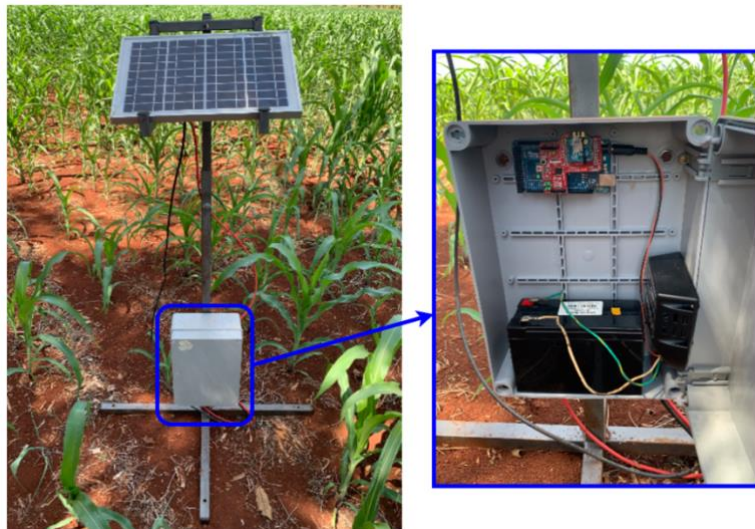


Figura 19 – Roteador ZigBee montado.
Fonte: Autoria Própria (2021).

Por utilizar o *shield* de expansão para Arduino, as ligações entre o Arduino Mega 2560 e o Módulo ZigBee são resolvidas através do encaixe da placa de expansão no Arduino e do ZigBee sobre a placa de expansão.

As ligações realizadas entre o Arduino Mega 2560 e os sensores GUVA e BME são as mesmas apresentadas na topologia LoRa/LoraWAN, representadas no Quadro 6 e no Quadro 7.

Antes de acoplar o módulo ZigBee ao *shield* de expansão foi realizada a configuração dos rádios utilizando a ferramenta XCTU. O Quadro 12 apresenta os parâmetros de configuração e os respectivos valores para cada tipo de dispositivo configurado para o projeto.

Parâmetro	COORDENADOR	ROTEADOR	NÓ
PAN ID	1234	1234	1234
JV	-	Habilitado	Habilitado
CE	Habilitado	-	-
CH	15	15	15
SC	7FFF	7FFF	7FFF
DH	SH do Destino	SH do Destino	SH do Destino
DL	SL do Destino	SL do Destino	SL do Destino
NI	COORDENADOR	ROTEADOR	NÓ
SP	1F4	1F4	1F4
SM	-	-	Cyclic sleep
SO	-	-	2

Quadro 12 – Configuração dos rádios ZigBee.
Fonte: Autoria Própria (2021).

Os valores SH e SL do destino correspondem aos endereços dos rádios de destino das mensagens enviadas pelo rádio a ser configurado, ou seja, ao realizar a configuração do dispositivo final que se conecta a um roteador, os parâmetros DH e DL do dispositivo final correspondem respectivamente ao SH e SL do roteador.

O Apêndice C contém o código executado no microcontrolador responsável por obter os valores lidos pelos sensores e enviar através do módulo ZigBee para o roteador ou coordenador. A seguir são apresentadas partes importantes desse código.

No Quadro 13 vemos as bibliotecas utilizadas para a execução da lógica no Arduino no dispositivo final ZigBee. São duas das mesmas bibliotecas utilizadas na lógica do microcontrolador da topologia LoRa/LoRaWAN, utilizadas para manipular o sensor BME280.

```
// -- BIBLIOTECA USADAS
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
```

Quadro 13 – Bibliotecas utilizadas para execução do dispositivo final ZigBee no Arduino
Fonte: Autoria Própria (2021).

Para o projeto utilizamos o modo de operação transparente, portanto, nenhuma configuração a nível de código foi exigida.

Finalizando a lógica Arduino para a transmissão dos dados pelo rádio ZigBee, no Quadro 14 vemos a leitura da temperatura através do sensor BME280, a cópia dos valores da variável temperatura para o *array* “xbeedata” e a posterior escrita desses valores via porta serial, acompanhada do prefixo “T” que identifica que o dado enviado corresponde à temperatura.

```
float temperatura = bme.readTemperature();
memcpy(xbeedata, &temperatura, 4);
Serial.print("T");
Serial.print(temperatura);
Serial.println("");
```

Quadro 14 – Coleta da temperatura e envio dos dados via escrita na porta serial.
Fonte: Autorial Própria (2021).

O Apêndice D por sua vez contém o código executado no microcontrolador parte integrante da arquitetura do roteador, responsável por obter os valores enviados pelos dispositivos finais e retransmiti-los ao coordenador. A seguir são apresentadas partes importantes desse código.

O código do roteador é o mais simples, não sendo necessário o carregamento de nenhuma biblioteca externa e tampouco configuração do rádio via código, a parte principal da lógica Arduino corresponde à leitura dos dados recebidos via porta serial de entrada e a reescrita desses dados via porta serial de saída. Essa lógica pode ser visualizada no Quadro 15.

```
while (Serial.available() > 0){
  buffer = Serial.read();
  Serial.write(buffer);
  Serial1.println(buffer);
}
```

Quadro 15 – Recebimento e retransmissão de dados por parte do Roteador.
Fonte: Autorial Própria (2021).

A topologia ZigBee conta obrigatoriamente com pelo menos um coordenador, que irá receber os dados enviados pelos dispositivos finais e pelos roteadores e salvar em um banco de dados InfluxDB. O coordenador é composto por três elementos abaixo apresentados e que também podem ser observados na Figura 20:

- a) Módulo ZigBee: a exemplo da arquitetura dos dispositivos finais e dos roteadores, o principal componente do coordenador é o módulo ZigBee, responsável pelo recebimento dos dados enviados pelos dispositivos finais e pelos roteadores. Neste projeto foi utilizado especificamente o módulo XBee S2 S2C ZigBee da fabricante Digi, que conta com antena integrada representando potência de transmissão de 3.1mW (+5dBm).
- b) Adaptador USB para módulos XBee: utilizado para conectar via cabo USB o rádio ZigBee ao microcontrolador. Para o projeto foi utilizado especificamente o componente *Xbee Explorer USB Adapter*.
- c) *Raspberry Pi3*: responsável por comportar o programa que executa a lógica de leitura dos dados e persistência dos mesmos no banco de dados InfluxDB.

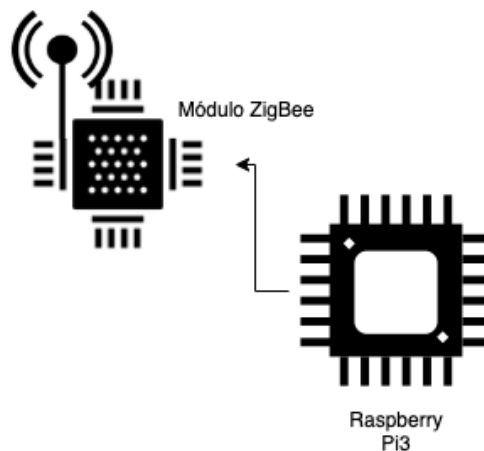


Figura 20 – Projeto do Coordenador ZigBee.
Fonte: Autoria Própria (2021).

A Figura 21 representa o coordenador montado, de acordo com os componentes previstos no esquema da Figura 20.



Figura 21 – Coordenador ZigBee montado.
Fonte: Autoria Própria (2021).

O Apêndice E contém o código executado no programa Python que foi programado para rodar automaticamente ao iniciar o Raspberry Pi. Esse programa é responsável por obter os valores enviados pelos dispositivos finais e pelos roteadores e salvar na base de dados InfluxDB. A seguir são apresentadas partes importantes do código desse programa.

No Quadro 16 verifica-se as bibliotecas utilizadas para a execução do programa Python. A seguir estão brevemente descritas as bibliotecas utilizadas:

- a) influxdb: utilizada conexão com a base de dados InfluxDB e a persistência dos dados recebidos.
- b) datetime: utilizada para recuperação da data e hora atuais do Raspberry para persistência dos registros das medições recebidas.
- c) serial: utilizada para a instanciação da conexão com a porta serial através da qual os dados são recebidos.
- d) logging: utilizada para armazenar o *log* do programa em um arquivo de texto.

```
// -- BIBLIOTECA USADAS
from influxdb import InfluxDBClient
from datetime import datetime
import serial
import logging
```

Quadro 16 – Bibliotecas utilizadas para execução do programa Python no Coordenador ZigBee
Fonte: Aatoria Própria (2021).

5.3.2.2 Armazenamento dos dados na solução ZigBee

O trecho de código apresentado no Quadro 17 representa a inicialização e configuração do *logger*, a instanciação da conexão serial, em seguida a instanciação da conexão com a base de dados e por fim leitura dos dados recebidos e sua persistência na base de dados InfluxDB.

```
logging.basicConfig(level=logging.DEBUG, filename='app.log', filemode='w',
format='%%(asctime)s - %(name)s - %(levelname)s - %(message)s')

ser = serial.Serial('/dev/ttyUSB0',9600,timeout=.5)

client = InfluxDBClient('3.137.140.97', 8086, '', '', 'mestrado_zigbee')
```

Quadro 17 – Código python responsável pela instanciação e inicialização das variáveis e conexões
Fonte: Aatoria Própria (2021).

A leitura dos dados e preparação do registro a ser salvo na base de dados consiste em identificar qual tipo de informação está sendo recebida através do prefixo enviado pelo dispositivo final antes de cada valor. Ao identificar o recebimento de um *byte* vazio, o programa entende que foi concluído o recebimento de toda a informação, e prepara o registro para ser persistido. O trecho de código observado no Quadro 18 apresenta a identificação de um valor de temperatura recebido e em seguida a persistência do registro na base de dados InfluxDB.

```
if type == 'T':
    data_temperatura = data_string[1:]
    print(data_temperatura)
    logging.info('O valor da temperatura lida e: ' + data_temperatura)
    received_data = True

json_body = [
```

```

{
    "measurement": "end_device_zigbee",
    "tags": {
        "host": "end_devices_lotario",
        "device": data_device
    },
    "time": datetime.now(),
    "fields": {
        "umidade_value": float(data_umidade),
        "temperatura_value": float(data_temperatura),
        "uv_value": float(data_uv),
        "pressao_value": float(data_pressao),
        "altitude_value": float(data_altitude)
    }
}
]

client.write_points(json_body)

```

Quadro 18 – Código python responsável pela leitura e persistência dos dados recebidos
Fonte: Autoria Própria (2021).

Na Figura 22 pode se observar o resultado da seleção de registros da medida end_device_zigbee, correspondente às leituras dos sensores da rede ZigBee.

```

-> select * from end_device_zigbee;
name: end_device_zigbee
time                altitude_value device host                pressao_value temperatura_value umidade_value uv_value
-----
1609433025384546048 294.06      CS1   end_devices_lotario 978.43      28.52      69.96      3.37
1609433038346448896 294.09      CS1   end_devices_lotario 978.42      28.55      71.33      3.37
1609433051289103872 294.35      CS1   end_devices_lotario 978.39      28.62      70.43      3.37
1609433064392001024 294.06      CS1   end_devices_lotario 978.42      28.64      71.28      3.37
1609433077208349952 294         CS1   end_devices_lotario 978.43      28.63      70.91      3.37
1609433090153236992 293.98      CS1   end_devices_lotario 978.43      28.58      70.79      3.37
1609433103113195776 294.42      CS1   end_devices_lotario 978.38      28.66      71.18      3.37
1609433116069724928 294         CS1   end_devices_lotario 978.43      28.65      71.92      3.37
1609433129025655040 293.93      CS1   end_devices_lotario 978.44      28.61      69.87      3.37
1609433141971798784 294.09      CS1   end_devices_lotario 978.42      28.59      70.28      3.37
1609433154931632896 294.09      CS1   end_devices_lotario 978.42      28.58      70.44      3.37
1609433167888382976 294.07      CS1   end_devices_lotario 978.42      28.6       71.48      3.37
1609433180835501824 294.17      CS1   end_devices_lotario 978.41      28.62      72.29      3.37
1609433193792886016 294.29      CS1   end_devices_lotario 978.4       28.67      71.9       3.37

```

Figura 22 – Seleção de valores capturados pelos sensores e armazenados na base InfluxDB.

Fonte: Autoria Própria (2021).

5.4 INSTALAÇÃO DOS DISPOSITIVOS FINAIS

Os dispositivos foram instalados sobre uma estrutura metálica projetada especialmente para o projeto. O material utilizado para a confecção da estrutura é o metalon. Conforme pode ser observado na Figura 23, o suporte apoia em sua extremidade superior um pequeno painel solar e mais abaixo a caixa de componentes. Com o propósito de estabilidade em terreno irregular e terra mexida, a base do suporte foi projetada em forma de base cruzada, com estacas que perfuram o solo mantendo o suporte fixo e reduzindo os riscos de ser derrubado por ventos ou animais. A estrutura possui ajuste de altura com níveis de regulagem, podendo variar de 80 cm à 220 cm. Essa regulagem de altura foi proposta para atender às diferentes culturas e acompanhar o crescimento da planta, de forma a não atrapalhar o manejo, por exemplo, a pulverização.



Figura 23 – Instalação dos dispositivos finais.

Fonte: Autoria Própria (2021).

6 RESULTADOS E DISCUSSÕES

6.4 EXECUÇÃO DOS TESTES E COLETA DOS RESULTADOS

A coleta de dados foi realizada por meio de experimentos de campo, nos quais foram criadas e instaladas topologias de cada tecnologia analisada nos diversos cenários propostos (Quadro 4), e com o auxílio de sensores foram observadas as características de cada tecnologia e seu comportamento de acordo com cada cenário.

Os cenários foram montados no município de Missal, na Mesorregião Oeste do estado do Paraná.

Durante os meses de dezembro de 2020 e janeiro de 2021 foram montados os cenários e executados os testes nas lavouras de soja, enquanto nos meses de março e abril de 2021 foram montados os cenários e executados os testes nas lavouras de milho, respeitando o calendário de cultivo da região e considerando o atraso no plantio da soja safra 2020/2021 por conta da escassez de chuvas ocorrida nos meses previstos para a semeadura.

6.4.1 Execução dos testes na cultura da soja

A Figura 24 apresenta os pontos de instalação dos componentes da topologia LoRa/LoRaWAN, plotados sobre o mapa da região utilizada para a execução dos testes.

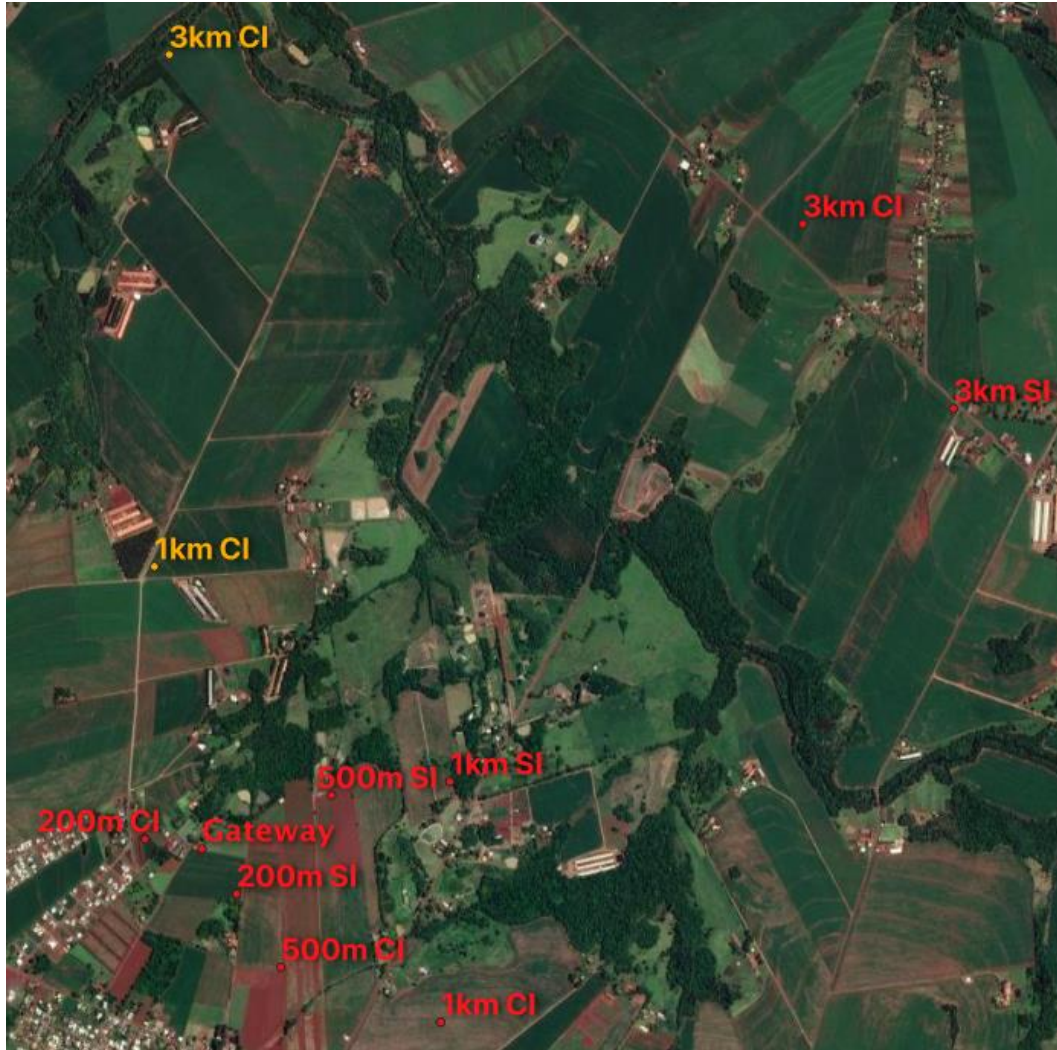


Figura 24 – Pontos de instalação dos componentes da rede LoRa/LoRaWAN na cultura do milho.

Fonte: Aatoria Própria (2021).

O *gateway* foi instalado na parte superior de um galpão, a uma altura de aproximadamente 5 metros do chão, enquanto os dispositivos finais foram espalhados pelas áreas aos arredores, posicionados em um raio de distância conforme determinado por cada cenário de teste previsto no Quadro 4. Além da distância, ainda conforme previsto no Quadro 4, foram considerados para o posicionamento dos nós a presença de fatores de interferência, por vezes representados por vegetação e outras pelo relevo.

Pode-se observar no mapa (Figura 24) a presença de dois pontos amarelos: 1 Km com interferência e 3 Km com interferência. Esses pontos representam instalações de dispositivos finais que não obtiveram sucesso na comunicação. A característica em comum aos dois cenários foi a presença de relevo. Eles foram

reposicionados em outros locais para a coleta de dados representado os seus respectivos cenários.

A Figura 25 apresenta os pontos de instalação dos componentes da topologia ZigBee, plotados sobre o mapa da região utilizada para a execução dos testes.



Figura 25 – Pontos de instalação dos componentes da rede ZigBee na cultura da soja.
Fonte: Autoria Própria (2021).

O coordenador foi instalado na parte superior de um galpão, a uma altura de aproximadamente 5 metros do chão, enquanto os dispositivos finais foram espalhados pelas áreas aos arredores, posicionados em um raio de distância conforme determinado por cada cenário de teste previsto no Quadro 4. Além da distância, ainda

conforme previsto no Quadro 4, foram considerados para o posicionamento dos nós a presença de fatores de interferência, por vezes representados por vegetação e outras pelo relevo.

Percebe-se, observando o mapa (Figura 25), a ausência de dispositivos finais nos cenários representados pelas distâncias acima de 500 metros. A razão da ausência desses pontos é o insucesso obtido na comunicação dos dispositivos finais com um roteador, e a comunicação desse roteador com outro roteador, e por vez, deste último com o coordenador.

Como dito, os dados transferidos foram armazenados em uma base de dados de série temporal. Portanto, para a tabulação e observação dos dados coletados, os registros das mensurações onde as informações foram persistidas foram selecionados. A solução de integração do CayenneLPP, para encapsulamento dos pacotes, e do InfluxDB, para persistência dos dados, utilizada em conjunto com a solução da ChirpStack para a topologia LoRa/LoRaWAN, fez com que cada informação lida pelos sensores fosse armazenada em uma mensuração distinta. Essa organização das informações exigiu a compilação de todos os dados lidos em uma mesma linha, ou seja, juntou-se os registros de temperatura, pressão, umidade e uv em uma única linha, formando assim uma tabela única, conforme pode ser observado na amostra de dados do Quadro 19.

Time	Device_name	Temperatura	Pressão	Umidade	UV
1609437840	3kmC	25,4	978	56	2,03
1609438020	3kmC	25,4	978,2	56,4	2,02
1609438140	3kmC	26,5	979,3	55,3	2,06
1609438200	3kmC	26,5	979,3	57,5	2,15
1609438320	3kmC	26,5	979,3	54,4	2,23

Quadro 19 – Exemplo de registros de leituras dos dispositivos finais LoRa, compilados em uma única tabela

Fonte: Autoria Própria (2021).

Já para os registros correspondentes às informações coletadas pelos dispositivos finais ZigBee não houve a necessidade de compilação dos dados, visto que todas as informações foram persistidas em uma única mensuração. Uma amostra da extração desses dados pode ser observada no Quadro 20.

time	device	temperatura_value	pressao_value	umidade_value	uv_value
1609665458	200mS	29,98	978,34	100	3,31
1609665578	200mS	30,06	978,36	100	3,31
1609665638	200mS	29,73	978,36	100	3,32

1609665698	200mS	29,69	978,41	100	3,31
1609665818	200mS	29,52	978,39	100	3,31

Quadro 20 – Exemplo de registros de leituras dos dispositivos finais ZigBee

Fonte: Autoria Própria (2021).

Em ambas as soluções, a diferenciação do cenário ao qual pertence cada registro armazenado está identificado por um atributo, no caso dos registros LoRa esse atributo corresponde ao “device_name”, enquanto nos registros ZigBee o atributo foi nomeado de “device”.

Os dispositivos finais de ambas as tecnologias foram configurados para realizar a transmissão de informações a cada 1 (um) minuto, sendo assim, a cada 30 (trinta) minutos de leitura, poderiam ser enviados até 30 pacotes de dados. Como a instalação de cada nó demandava esforços diferentes considerando deslocamento até o local e dificuldade de acesso, alguns nós permaneceram coletando dados por mais tempo do que os outros. Essas variações de duração da coleta não interferiram nos resultados, uma vez que foi extraída a taxa de transmissão de pacotes real *versus* a taxa esperada por períodos de 30 (trinta) minutos.

Através dos registros gravados nas *measurements* das bases no InfluxDB, exportados para arquivos CSV e em seguida transformados em planilhas XLS, foram elaborados gráficos para observação dos resultados em termos de taxa de envio de pacotes, conforme podem ser observados na Figura 26 para a tecnologia LoRa/LoRaWAN e na Figura 27 para ZigBee.

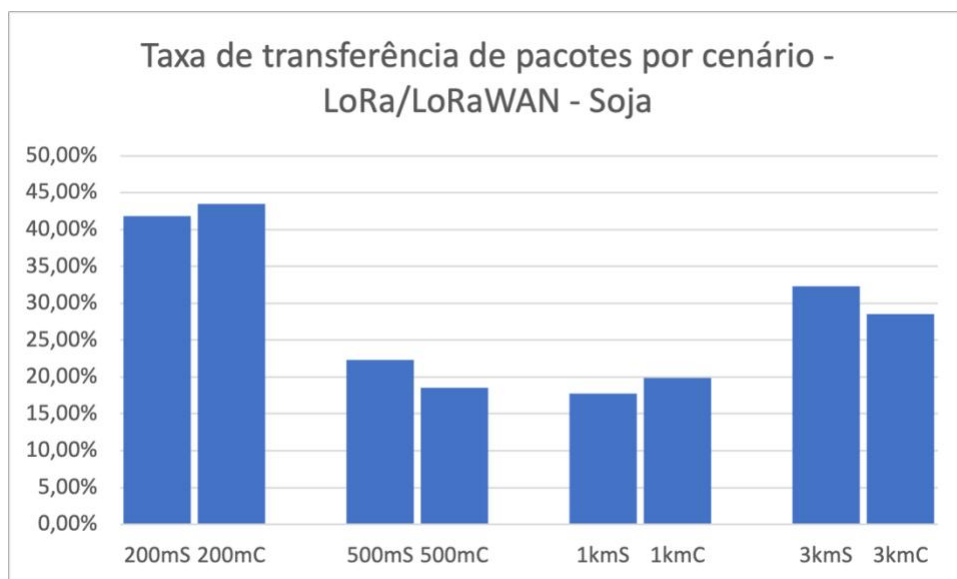


Figura 26 – Gráfico reproduzindo os percentuais de transferência de pacotes utilizando LoRa/LoRaWAN na soja.

Fonte: Aatoria Própria (2021).

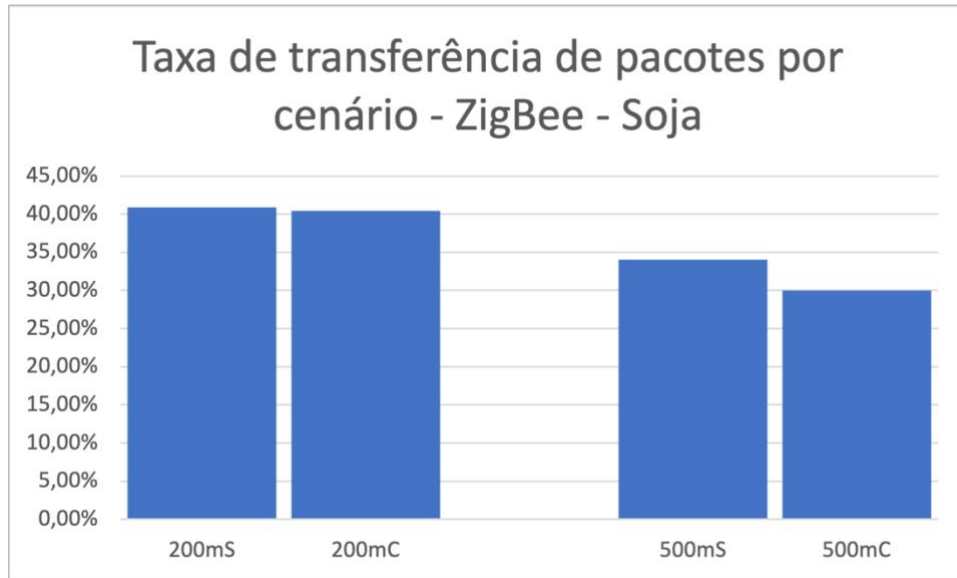


Figura 27 – Gráfico reproduzindo os percentuais de transferência de pacotes utilizando ZigBee na soja.
Fonte: Aatoria Própria (2021).

Na tentativa de verificar a correlação entre a taxa de transferência de pacotes e as demais variáveis lidas, utilizando a ferramenta de análise de dados do Microsoft Excel, extraiu-se os coeficientes de Pearson apresentados nas Tabela 1 e Tabela 2. Para tanto, através da manipulação das planilhas XLS, os registros de pacotes enviados/recebidos provenientes da base InfluxDB foram agrupados em blocos de intervalo de 30 minutos, e para cada bloco foram calculadas as médias das variáveis lidas (temperatura, umidade e UV) e a quantidade de pacotes recebidos. Esses dois resultados (média dos valores das variáveis e quantidade de pacotes recebidos) foram utilizadas como entrada para o cálculo dos coeficientes de Pearson.

	Temperatura	Umidade	UV
Pacotes enviados utilizando LoRa	-0,073899871	-0,23805	-0,0467897
Pacotes enviados utilizando ZigBee	0,24217014	-0,2714983	0,11193348

Tabela 1 – Análise de correlação entre as variáveis lidas e a taxa de envio de pacotes utilizando LoRa/LoRaWAN na cultura da soja
Fonte: Aatoria Própria (2021)

6.4.2 Execução dos Testes na Cultura do Milho

Para a execução dos testes na cultura do milho os dispositivos finais que compõem a topologia LoRa/LoRaWAN foram posicionados aproximadamente nas posições de quando executados os testes na soja. Esse posicionamento pode ser observado na Figura 24.

Com relação aos testes com a tecnologia ZigBee na cultura do milho, os mesmos foram impossibilitados pela limitação da tecnologia, a qual não teve eficiência na transmissão de pacotes de dentro da lavoura sendo cobertos pela densa folhagem e caules do milho.

Da mesma forma que para os testes na cultura da soja, os dados transferidos foram armazenados em uma base de dados de série temporal, cada variável coletada armazenada em uma mensuração distinta, exigindo a compilação de todos os dados lidos em uma mesma linha, formando assim o que pode ser observado na amostra de dados do Quadro 19. Ainda conforme os testes na soja, a diferenciação do cenário ao qual pertence cada registro armazenado foi identificado por um atributo nomeado de "device_name". Também mantendo o padrão correspondente aos testes executados na lavoura de soja, os dispositivos finais foram configurados para realizar a transmissão de informações a cada 1 (um) minuto, sendo assim, a cada 30 (trinta) minutos de leitura, poderiam ser enviados até 30 (trinta) pacotes de dados.

Os resultados em termos de taxa de envio de pacotes podem ser observados na Figura 28 para a tecnologia LoRa/LoRaWAN.

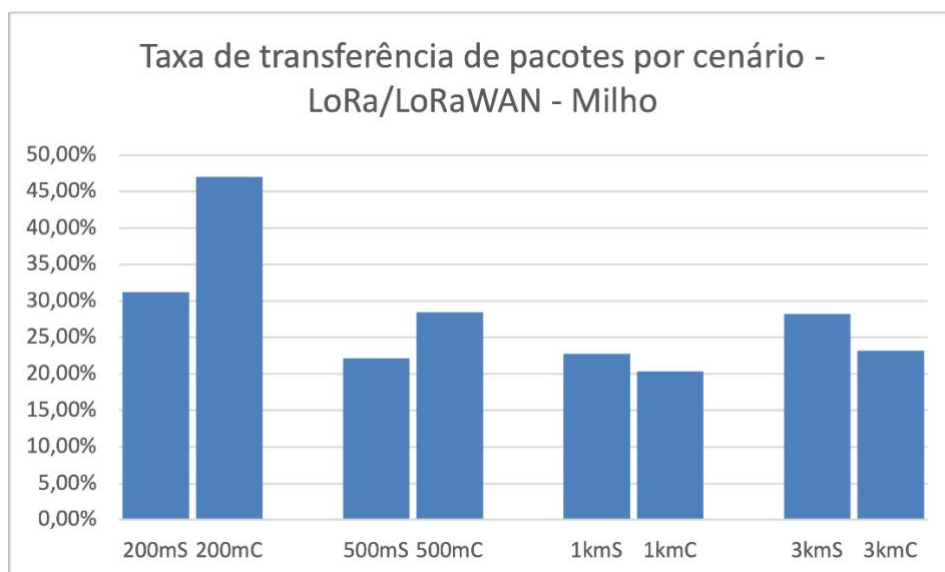


Figura 28 – Gráfico reproduzindo os percentuais de transferência de pacotes utilizando LoRa/LoRaWAN no milho.
Fonte: Autoria Própria (2021).

Na tentativa de verificar a correlação entre a taxa de transferência de pacotes e as demais variáveis lidas, utilizando a ferramenta de análise de dados do Microsoft Excel, extraiu-se os coeficientes de Pearson apresentados na Tabela 2.

	Temperatura	Umidade	UV
Pacotes enviados	0,12148136	-0,1346205	0,1409342

Tabela 2 – Análise de correlação entre as variáveis lidas e a taxa de envio de pacotes na cultura do milho
Fonte: Autoria Própria (2021)

6.5 DISCUSSÕES

Através da exploração dos dados resultado dos testes realizados e expostos na seção anterior, a seguir apresentamos análises objetivas dos resultados.

Analisando a Figura 24, nota-se que LoRa/LoRaWAN apresentou dificuldades e por vezes até falta de funcionamento quando da presença de relevo considerável. ZigBee por sua vez demonstrou insucesso ao montar topologia envolvendo mais que um dispositivo roteador, conforme Figura 25. Ainda se tratando de ZigBee, percebe-se que, nas condições de posicionamento da antena dos dispositivos finais e configurações realizadas para o estudo, houve a impossibilidade de utilização desta tecnologia na lavoura do milho, uma vez que a mesma não realizou com êxito a transferência de pacotes quando posicionada dentro da plantação de milho, mesmo que embora tenha sido inserida à poucos metros adentro e em uma distância de menos de 10 (dez) metros do dispositivo coordenador, ou seja, nas condições empregadas para o estudo os pacotes enviados pelos dispositivos finais ZigBee na lavoura de milho não foram recebidos pelo respectivo coordenador. Isso nos faz concluir que, nas condições do estudo, ZigBee não é a tecnologia ideal a ser aplicada em cenários de lavoura de milho ou semelhantes, entretanto, pode surtir efeitos positivos se utilizados outros parâmetros de configuração nos dispositivos e melhor posicionamento da antena dos dispositivos finais.

Quando o foco é na taxa de perda de pacotes, contemplando a Figura 26 e Figura 28, no que se refere ao LoRa/LoRaWAN, percebe-se uma diferença mínima

quando comparados os cenários com e sem interferência, por vezes, o cenário com interferência alcançou taxa de transferência maior, e especialmente nos cenários de 200m no milho essa diferença foi significativa, o que nos leva a deduzir que as interferências físicas propositalmente inseridas nos cenários não foram suficientes para prejudicar a qualidade do sinal. Apesar da maior taxa de transferência ter sido percebida no cenário de menor distância, o percentual de entrega de pacotes foi considerado baixo para ambas as culturas, ou seja, dos pacotes enviados pelos dispositivos finais, poucos foram devidamente recebidos pelos *gateways* e coordenadores correspondentes, uma vez que atingiu apenas 43,45% no cenário de 200m na soja com interferência, fato que pode ser visto na Figura 26, e apenas 47,06% no cenário de 200m no milho com interferência, a ser percebido na Figura 28. Ainda analisando a taxa de perda de pacotes, nota-se uma queda de praticamente metade de assertividade na transferência de pacote nos cenários de 500m e 1km, se comparados aos cenários de 200m para ambas as culturas. Já os cenários de 3km por sua vez apresentaram redução da assertividade no envio dos pacotes, entretanto, menor do que os outros dois cenários ditos anteriormente.

Quanto à correlação da taxa de transferência de pacotes e as variáveis lidas, entendendo os coeficientes de Pearson apresentados na Tabela 1 para a soja e na Tabela 2 para o milho, observa-se que não há fortes evidências de que a taxa de transferência de pacotes seja afetada por alguma destas variáveis, sendo todos os coeficientes gerados muito distantes dos valores de 1 ou -1, o que indica que não há relação forte entre as duas variáveis observadas. Para os testes na soja a correlação com coeficiente mais próximo de 1 ou -1 foi a da umidade, entretanto, considerada insignificante por ser de apenas -0,23805, enquanto para os testes no milho a correlação com coeficiente mais próximo de 1 ou -1 foi a da intensidade de raios ultravioleta, entretanto, também considerada insignificante por ser de apenas 0,1409342.

Mudando o foco para o ZigBee, testes realizados com sucesso apenas no soja, assim como avaliou-se na tecnologia LoRa/LoRaWAN a taxa de perda de pacotes apresenta mínima diferença se comparados os cenários com e sem interferência física, como pode ser observado pela Figura 27. Ainda analisando esta mesma figura, percebemos mais uma vez o baixo percentual de entrega de pacotes, chegando a apenas 40,89%, mesmo em cenários de curta distância e sem presença de fatores físicos de interferência. Por fim, nota-se pela Figura 27, uma redução pouco

significativa da taxa de entrega de pacotes quando deixamos de olhar para os cenários de 200m e passamos a olhar para os cenários de 500m.

Quanto à correlação da taxa de transferência de pacotes e as variáveis lidas, verificando os coeficientes de Pearson apresentados na Tabela 1, apesar de observarmos coeficientes mais fortes se comparado ao resultado gerado pela correlação dos dados obtidos pelos testes com LoRa/LoRaWAN, continuamos percebendo que não há evidência de que a taxa de transferência de pacotes seja afetada por alguma destas variáveis, sendo todos os coeficientes gerados distantes dos valores de 1 ou -1, o que indica que não há relação entre as duas variáveis observadas. A correlação com coeficiente mais próximo de 1 ou -1 foi novamente a da umidade, entretanto, considerada insignificante por ser de apenas -0,2714983.

A taxa de recebimento de pacotes baixa e sem percepção de interferência significativa pelos obstáculos empregados pode ser explicada por não ter sido considerada a zona de Fresnel ao instalar o Gateway e os Dispositivos Finais. Para Ribas (2002) a elaboração de um projeto para a infra-estrutural vertical é fator determinante para uma boa performance do enlace de rádio, através do qual pode-se determinar o melhor posicionamento das antenas, sendo necessário o projetista considerar dois importantes parâmetros neste projeto: o fator de correção da Terra e a liberação total ou parcial da zona de Fresnel. Piau (2013) descreve a zona de Fresnel como a relação teórico-prático de interferência dependente da altura que existe entre a antena e a superfície refletora do caminho percorrido pelo sinal.

Especialmente se tratando de ZigBee, conforme apontado pelos estudos realizados por Liang, Priyantha, Liu e Terzis (2010), Chong, Hwang, Jung e Sung (2007), Sikora e Groza (2005) e Garroppo, Gazzarrini, Giordano e Tavanti (2011), a baixa taxa de recebimento dos pacotes pode ser justificada pela presença de redes WLAN nas proximidades da instalação da solução, tais quais WiFi e Bluetooth, já que estas tecnologias compartilham da mesma faixa de frequência de transmissão: ISM 2.4GHz.

Quanto aos dados climáticos e atmosféricos, apesar da temperatura e da umidade relativa do ar estarem sendo citados em estudos recentes sobre os efeitos das condições climáticas na qualidade do sinal, o que foi confirmado pelo estudo de Luomala e Hakala (2015), para este estudo, o fato de não ter havido uma variação tão significativa na temperatura e umidade do ar pode explicar a não identificação da correlação dessas variáveis e a taxa de recebimento dos pacotes.

7 CONCLUSÃO

Considerando as revisões de literatura e os estudos realizados verificou-se que LoRa/LoRaWAN, ZigBee, SigFox e WiFi(ah) são tecnologias WSN aplicáveis à cenários agrícolas.

Considerando os resultados obtidos através dos testes práticos, conclui-se assim que se não houver exigência de alta assertividade no envio dos pacotes, ambas as tecnologias atendem à cenários agrícolas, entretanto, sendo indicada a utilização de LoRa/LoRaWAN em cenários que requerem maior distância de transferência e sendo descartada a utilização de ZigBee na cultura do milho, considerando as condições de configuração dos dispositivos e posicionamento das antenas, após atingir altura e densidade de planta adulta, cenário este considerado nos testes envolvendo as tecnologias e a plantação de milho.

Um último ponto de atenção a ser indicado é quanto à presença de relevo, muito embora a solução LoRa/LoRaWAN tenha apresentado resultados satisfatórios, percebeu-se o não funcionamento quando havia consideráveis dobras no relevo entre a sua posição e a do *gateway*.

7.1 TRABALHOS FUTUROS

Portanto, para trabalhos futuros sugere-se considerar a zona de Fresnel e testes com maiores variações de temperatura, umidade, pressão e radiação, para verificação mais eficaz de assertividade e influência das variáveis na taxa de recebimento dos pacotes. Também sugere-se considerar a disposição da antena dos dispositivos finais de forma mais elevada, reduzindo assim a interferência causada pela própria cultura.

REFERÊNCIAS

- ADAME, Toni et al. IEEE 802.11 AH: the WiFi approach for M2M communications. **IEEE Wireless Communications**, v. 21, n. 6, p. 144-152, 2014.
- ADELANTADO, Ferran; VILAJOSANA, Xavier; TUSET-PEIRO, Pere; MARTINEZ, Borja; MELIÀ-SEGUÍ, Joan; WATTEYNE, Thomas. **Understanding the Limits of LoRaWAN**. IEEE Communications Magazine: 2017. Disponível em: <https://arxiv.org/pdf/1607.08011.pdf>. Acesso em: 17 abr. 2019.
- AHMED, Nurzaman; DE, Debashis; HUSSAIN, Iftekhar. Internet of Things (IoT) for smart precision agriculture and farming in rural areas. **IEEE Internet of Things Journal**, v. 5, n. 6, p. 4890-4899, 2018.
- AHMED, N.; RAHMAN, H.; HUSSAIN, Md I. A comparison of 802.11 ah and 802.15.4 for IoT. **Ict Express**, v. 2, n. 3, p. 100-102, 2016.
- AKYILDIZ, Ian F. et al. Wireless sensor networks: a survey. **Computer networks**, v. 38, n. 4, p. 393-422, 2002.
- ALAH, Md Eshrat EE et al. An internet-of-things enabled smart sensing system for nitrate monitoring. **IEEE Internet of Things Journal**, v. 5, n. 6, p. 4409-4417, 2018.
- ARAGUES, Antonio et al. Trends in entertainment, home automation and e-health: Toward cross-domain integration. **IEEE Communications Magazine**, v. 50, n. 6, p. 160-167, 2012.
- AUST, Stefan; ITO, Tetsuya. Sub 1GHz wireless LAN propagation path loss models for urban smart grid applications. In: **2012 International Conference on Computing, Networking and Communications (ICNC)**. IEEE, 2012. p. 116-120.
- BADIHI OLYAEI, Behnam. **Modeling, Performance Evaluation and Suitability Study of Zigbee Technology for Machine-to-Machine Communications Applications**. 2013. Dissertação de Mestrado.
- BALMOS, Andrew D. et al. Investigation of Bluetooth Communications for Low-Power Embedded Sensor Networks in Agriculture. **Transactions of the ASABE**, v. 59, n. 5, p. 1021-1029, 2016.
- BELLALTA, Boris et al. Next generation IEEE 802.11 Wireless Local Area Networks: Current status, future directions and open challenges. **Computer Communications**, v. 75, p. 1-25, 2016.
- CASALS, Lluís et al. Modeling the energy performance of LoRaWAN. **Sensors**, v. 17, n. 10, p. 2364, 2017. Disponível em: <https://www.mdpi.com/1424-8220/17/10/2364>. Acesso em: 17 abr. 2019.

CHEN, Hsing-Chung et al. A security gateway application for End-to-End M2M communications. **Computer Standards & Interfaces**, v. 44, p. 85-93, 2016. DOI: <https://doi.org/10.1016/j.csi.2015.09.001>

CHIRPSTACK. ChirpStack architecture. 2021. Disponível em: <https://www.chirpstack.io/project/architecture/>. Acesso em: 01 jan. 2021.

CHONG, Jo Woon; HWANG, Ho Young; JUNG, Chang Yong; SUNG, Dan Keun (2007). [IEEE 2007 International Symposium on Communications and Information Technologies - Sydney, Australia (2007.10.17-2007.10.19)] 2007 International Symposium on Communications and Information Technologies - Analysis of throughput in a ZigBee network under the presence of WLAN interference. , (), 1166–1170. doi:10.1109/ISCIT.2007.4392193

ČOLAKOVIĆ, Alem; HADŽIALIĆ, Mesud. Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues. **Computer Networks**, v. 144, p. 17-39, 2018.

COSSINI, Fabio. Lorawan: Uma Rede Alternativa Para A Internet Das Coisas. **Mini Paper Series** Ano 12. Novembro, 2016 – Nº 274. Disponível em: <https://www.ibm.com/developerworks/community/blogs/tlcbre/entry/mp274?lang=en>. Acesso em: 16 abr. 2019.

DA ROSA, Rafael Pedroso. Dispositivos IoT aplicáveis à agricultura intensiva e os resultados já alcançados. **Datacenter: projeto, operação e serviços-Unisul Virtual**, 2017.

DIGI. XBee®/XBee-PRO S2C Zigbee®: RF Module: User Guide. 2020. Disponível em: <https://www.digi.com/resources/documentation/digidocs/pdfs/90002002.pdf>. Acesso em: 19 nov. 2020.

DOS SANTOS, Uélison Jean L. et al. AgriPrediction: A proactive internet of things model to anticipate problems and improve production in agricultural crops. **Computers and electronics in agriculture**, v. 161, p. 202-213, 2019.

FENG, Xiang; YAN, Fang; LIU, Xiaoyu. Study of Wireless Communication Technologies on Internet of Things for Precision Agriculture. **Wireless Personal Communications**, p. 1-18, 2019.

GARROPPO, Rosario G.; GAZZARRINI, Loris; GIORDANO, Stefano; TAVANTI, Luca (2011). [IEEE 2011 IEEE International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM) - Lucca, Italy (2011.06.20-2011.06.24)] 2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks - Experimental assessment of the coexistence of Wi-Fi, ZigBee, and Bluetooth devices. , (), 1–9. doi:10.1109/wowmom.2011.5986182

GENG, Liwei; DONG, Tingting. An Agricultural Monitoring System Based on Wireless Sensor and Depth Learning Algorithm. **International Journal of Online Engineering (iJOE)**, v. 13, n. 12, p. 127-137, 2017.

HAXHIBEQIRI, Jetmir et al. A survey of lorawan for iot: From technology to application. **Sensors**, v. 18, n. 11, p. 3995, 2018. Disponível em: <https://www.mdpi.com/1424-8220/18/11/3995>. Acesso em 18 abr. 2019.

HOLLER, J. et al. From Machine-to-Machine to the Internet of Things. **Academic Press**, 2014.

IAPAR. Monitoramento Mensal. Disponível em: <http://www.iapar.br/modules/conteudo/conteudo.php?conteudo=635>. Acessado em: 01 mar. 2020.

IBGE. Conheça as Cidades e Estados o Brasil. Disponível em: <https://cidades.ibge.gov.br/>. Acessado em: 26 fev. 2020.

JARDIM, Arnaldo. Agricultura de precisão: uma nova fronteira agrícola. **AgroANALYSIS**, v. 37, n. 10, p. 48, 2018.

JAWAD, Haider Mahmood et al. Power reduction with sleep/wake on redundant data (SWORD) in a wireless sensor network for energy-efficient precision agriculture. **Sensors**, v. 18, n. 10, p. 3450, 2018.

JIBER, Yassine; HARROUD, Hamid; KARMOUCH, Ahmed. Precision agriculture monitoring framework based on WSN. In: **2011 7th International Wireless Communications and Mobile Computing Conference**. IEEE, 2011. p. 2015-2020.

KEYSIGHT TECHNOLOGIES (2016). **Internet of Things (IoT)**. Disponível em: <https://www.keysight.com/br/pt/assets/7018-05035/posters/5992-1217.pdf>. Acessado em 06 nov. 2019.

KHANNA, Abhishek; KAUR, Sanmeet. Evolution of Internet of Things (IoT) and its significant impact in the field of Precision Agriculture. **Computers and electronics in agriculture**, v. 157, p. 218-231, 2019.

LIANG, Chieh-Jan Mike; PRIYANTHA, Nissanka Bodhi; LIU, Jie; TERZIS, Andreas (2010). [ACM Press the 8th ACM Conference - Zürich, Switzerland (2010.11.03-2010.11.05)] Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems - SenSys '10 - Surviving wi-fi interference in low power ZigBee networks. , (), 309-. doi:10.1145/1869983.1870014

Libelium Comunicaciones Distribuidas S.L. Wasmote Sigfox: Networking Guide. Document Version: v7.3 - 04/2018. Disponível em: <http://www.libelium.com/development/wasmote/documentation/wasmote-sigfox-networking-guide/?action=download>. Acessado em 28 jan. 2020.

LORA ALLIANCE. 2017. Disponível em: <https://www.lora-alliance.org/>. Acessado em 17 out. 2019.

LORA ALLIANCE. 2020. What is LoRaWAN® Specification. Disponível em: <https://lora-alliance.org/about-lorawan/>. Acessado em 05 abr. 2020.

LUOMALA, Jari; HAKALA, Ismo (2015). Annals of Computer Science and Information Systems [IEEE 2015 Federated Conference on Computer Science and Information Systems - (2015.9.13-2015.9.16)] Proceedings of the 2015 Federated Conference on Computer Science and Information Systems - Effects of Temperature and Humidity on Radio Signal Strength in Outdoor Wireless Sensor Networks. , 5(), 1247–1255. doi:10.15439/2015F241

MONTORI, Federico et al. Machine-to-machine wireless communication technologies for the Internet of Things: Taxonomy, comparison and open issues. **Pervasive and Mobile Computing**, v. 50, p. 56-81, 2018. DOI: <https://doi.org/10.1016/j.pmcj.2018.08.002>

OJHA, Tamoghna; MISRA, Sudip; RAGHUWANSHI, Narendra Singh. Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges. **Computers and Electronics in Agriculture**, v. 118, p. 66-84, 2015. DOI: <http://dx.doi.org/10.1016/j.compag.2015.08.011>

PIAU, Diego de Brito. Projeto de dimensionamento de enlace de rádio associado a atenuações devido à chuva utilizando Celplan e Pathloss. 2013.

Programa Agricultura de Precisão do SENAR (2017). Agricultura digital abre perspectivas para pesquisa. Disponível em: <https://www.embrapa.br/busca-de-noticias/-/noticia/28915522/agricultura-digital-abre-perspectivas-para-pesquisa>. Acesso em 20 out. 2019.

QADIR, Qahhar Muhammad et al. Low power wide area networks: a survey of enabling technologies, applications and interoperability needs. **IEEE Access**, v. 6, p. 77454-77473, 2018.

RAWAT, Priyanka; SINGH, Kamal Deep; BONNIN, Jean Marie. Cognitive radio for M2M and Internet of Things: A survey. **Computer Communications**, v. 94, p. 1-29, 2016. DOI: <https://doi.org/10.1016/j.comcom.2016.07.012>

RIBAS, Júlio César da Costa. Perfil de link sem fio em ambiente aberto: avaliação através de medições. 2002.

SEIXAS, M. A.; CONTINI, E. Internet das coisas (IoT): inovação para o agronegócio. **Área de Informação da Sede-Nota Técnica/Nota Científica (ALICE)**, 2017.

SIGFOX. (2020) SIGFOX. Disponível em: <https://www.sigfox.com/>. Acessado em: 07 jan. 2020.

SIKORA, A.; GROZA, V.F. (2005). [IEEE 2005 IEEE Instrumentation and Measurement Technology - Ottawa, ON, Canada (16-19 May 2005)] 2005 IEEE Instrumentation and Measurement Technology Conference Proceedings - Coexistence of IEEE802.15.4 with other Systems in the 2.4 GHz-ISM-Band. , 3(), 1786–1791. doi:10.1109/IMTC.2005.1604479

SUNDMAEKER, Harald et al. Internet of food and farm 2020. **Digitising the Industry-Internet of Things connecting physical, digital and virtual worlds**. Ed: Vermesan, O., & Friess, P, p. 129-151, 2016.

TALAVERA, Jesús Martín et al. Review of IoT applications in agro-industrial and environmental fields. **Computers and Electronics in Agriculture**, v. 142, p. 283-297, 2017.

TEIXEIRA, Grazielle Bonaldi; ALMEIDA, João Vítor Peroni de. **Rede LoRa® e protocolo LoRaWAN® aplicados na agricultura de precisão no Brasil**. 2017. Trabalho de Conclusão de Curso. Universidade Tecnológica Federal do Paraná.

TZOUNIS, Antonis et al. Internet of Things in agriculture, recent advances and future challenges. **Biosystems Engineering**, v. 164, p. 31-48, 2017.

XIA, Nian; YANG, Chu-Sing. Recent advances in machine-to-machine communications. **J. Comput. Commun**, v. 4, p. 107-111, 2016.

ZHANG, Xufeng et al. Thoreau: A subterranean wireless sensing network for agriculture and the environment. In: **2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)**. IEEE, 2017. p. 78-84.

ZULKIFLI, C. Z.; NOOR, N. N. Wireless Sensor Network and Internet of Things (IoT) Solution in Agriculture. **Pertanika Journal of Science & Technology**, v. 25, n. 1, 2017.

APÊNDICE A – CÓDIGO ARDUINO PARA O NÓ LORA

```

// -- BIBLIOTECA USADAS
#include <LoRaWan.h>
#include <CayenneLPP.h>
#include <TinyGPS++.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>

char buffer[256];
unsigned long delayTime;
int leituraGPS = 0;
Adafruit_BME280 bme; // I2C
CayenneLPP lpp(51);
TinyGPSPlus gps;

#define SEALEVELPRESSURE_HPA (1013.25)

void setup(void){
  SerialUSB.begin(9600);
  while(!SerialUSB);
  while(!Serial1);

  //SETUP BME
  unsigned status;
  SerialUSB.println("Setup BME init");
  delay(2000);
  // You can also pass in a Wire library object like &Wire2
  status = bme.begin(0x76);
  SerialUSB.println("SETUP DONE");

  lora.init();
  lora.setDeviceReset();

  memset(buffer, 0, 256);
  lora.getVersion(buffer, 256, 1);
  SerialUSB.print(buffer);

  memset(buffer, 0, 256);
  lora.getId(buffer, 256, 1);
  SerialUSB.print(buffer);

  //---- setId(char *DevAddr, char *DevEUI, char *AppEUI)
  lora.setId("01882f84", "73301be462247b1c", "28d9790df644928c");

  //---- void LoRaWanClass::setKey(char *NwkSKey, char *AppSKey, char *AppKey)
  lora.setKey("5f218b7f546726faa1057c601085ce3a",
    "c077b481ca67956d77c4c0bc16af12eb", "c077b481ca67956d77c4c0bc16af12eb");
  lora.setDeciveMode(LWABP);

```

```

lora.setDataRate(DR3, AU915);

lora.setChannel(0, 916.8);
lora.setChannel(1, 917.0);
lora.setChannel(2, 917.2);
lora.setChannel(3, 917.4);
lora.setChannel(4, 917.6);
lora.setChannel(5, 917.8);
lora.setChannel(6, 918.0);
lora.setChannel(7, 918.2);

lora.setReceiceWindowFirst(1, 923.9);
lora.setReceiceWindowSecond(923.3, DR8);

lora.setAdaptiveDataRate(true);

lora.setDutyCycle(false);
lora.setJoinDutyCycle(false);

lora.setPower(14);

delayTime = 60000;

}

void loop(void){

    lpp.reset();

    float temperatura = bme.readTemperature();
    lpp.addTemperature(1, temperatura);
    SerialUSB.print("====Temperature====");
    SerialUSB.print(temperatura);
    SerialUSB.println(" *C");

    float pressao = bme.readPressure() / 100.0F;
    lpp.addBarometricPressure(2, pressao);
    SerialUSB.print("====Pressure==== ");
    SerialUSB.print(pressao);
    SerialUSB.println(" hPa");

    float umidade = bme.readHumidity();
    lpp.addRelativeHumidity(3, umidade);
    SerialUSB.print("====Humidity====");
    SerialUSB.print(umidade);
    SerialUSB.println(" %");

    float guva = analogRead(A0)/1024.0*5.0;
    lpp.addAnalogInput(4, guva);
    SerialUSB.print("====GUVA====");

```

```

SerialUSB.println(guva); //Write the serial data

float altitude = bme.readAltitude(SEALEVELPRESSURE_HPA);
SerialUSB.print("=====Approx. Altitude=====");
SerialUSB.print(altitude);
SerialUSB.println(" m");

SerialUSB.println("=====BUFFER=====");
uint8_t* texto=lpp.getBuffer();
for(int i=0;i<lpp.getSize();i++){
    SerialUSB.println(texto[i]);
}

bool result = false;
memset(buffer, 0, 256);
lora.getIid(buffer, 256, 1);
SerialUSB.print(buffer);

// Envio do buffer LPP para o servidor
result = lora.transferPacketWithConfirmed(lpp.getBuffer(), lpp.getSize());

if(result) {
    short length;
    short rssi;

    memset(buffer, 0, 256);
    length = lora.receivePacket(buffer, 256, &rssi);

    if(length){

        SerialUSB.print("Length is: ");
        SerialUSB.println(length);
        SerialUSB.print("RSSI is: ");
        SerialUSB.println(rssi);
        SerialUSB.print("Data is: ");

        for(unsigned char i = 0; i < length; i ++){
            SerialUSB.print("0x");
            SerialUSB.print(buffer[i], HEX);
            SerialUSB.print(" ");
        }

        SerialUSB.println();

    }
}

delay(delayTime);
}

```

APÊNDICE B – CONFIGURAÇÃO DO GATEWAY E SERVIDORES LORA/LORAWAN

Configuração de Arquitetura LoRA

Inicialmente deve-se baixar a imagem do sistema operacional, disponível em: <https://www.radioenge.com.br/solucoes/iot/18-gateway-lorawan.html>.

Configuração do Gateway

Na sequência deve-se fazer a configuração do gateway, para tanto, deve ser acessada a pasta `/opt/LoRa/packet_forwarder/lora_pkt_fwd/`, nela os arquivos que devem ser alterados são: ***local_conf.json*** e ***global_conf.json***.

1. A primeira etapa da configuração corresponde ao arquivo ***local_conf.json*** no qual deve-se definir o Gateway ID.

```
GNU nano 2.7.4 File: local_conf.json
/* Put there parameters that are different for each gateway (eg. pointing one gateway to a test server while the others stay in production) */
/* Settings defined in global_conf will be overwritten by those in local_conf */
"gateway_conf": {
  "gateway_ID": "289bee7084150c2f"
}
}
```

Por convenção o Gateway ID segue o padrão EUI-64 definido pela IEEE, sendo os primeiros 24 bits os identificadores do proprietário da rede fornecidos pelo "IEEE Registration Authority" e os outros 40 bits definidos por este proprietário. Para gerar o `gateway_ID` de um equipamento específico deve-se rodar a seguinte instrução na linha de prompt do sistema operacional.

sudo ./update_gwid.sh local_conf.json

2.No arquivo ***global_conf.json***, é o responsável pelas configurações globais do Gateway, tais como: plano de frequência, endereço de rede, taxa de atualização, entre outros. Nesta imagem, este arquivo já vem configurado com a banda AU915 Sub-Band 2. Bastando apenas configurar o endereço do servidor de rede e a porta.

a.Na imagem na linha onde aparece ***"server_address":***

"router.au.thethings.network", deve-se modifica-la para ***"server_address":***
"localhost", como apresentada na figura abaixo


```

GNU nano 2.7.4                                     File: global_conf.json
},
"tx_lut_14": {
  /* TX gain table, index 14 */
  "pa_gain": 3,
  "mix_gain": 12,
  "rf_power": 26,
  "dig_gain": 0
},
"tx_lut_15": {
  /* TX gain table, index 15 */
  "pa_gain": 3,
  "mix_gain": 14,
  "rf_power": 27,
  "dig_gain": 0
}
},
"gateway_conf": {
  "gateway_ID": "AAAAAAAAAAAAAAAA", // Gateway Radioenge
  /* change with default server address/ports, or overwrite in local_conf.json */
  "server_address": "localhost",
  "serv_port_up": 1700,
  "serv_port_down": 1700,
  /* adjust the following parameters for your network */
  "keepalive_interval": 10,
  "stat_interval": 30,
  "push_timeout_ms": 1000,
  /* forward only valid packets */
  "forward_crc_valid": true,
  "forward_crc_error": false,
  "forward_crc_disabled": false,
  /* GPS configuration */
  "gps_tty_path": "/dev/ttyAMA0",
  /* GPS reference coordinates */
  "ref_latitude": 0.0,
  "ref_longitude": 0.0,
  "ref_altitude": 0
}
}
}

```

3.O passo seguinte é a instalação e configuração dos componentes do pacote ChirpStack, conforme guia presente no

link: <https://www.chirpstack.io/project/guides/debian-ubuntu/>

- Deve-se instalar os programa **dirmngr** e **apt-transport-https**, necessários para o uso do repositório provido pela ChirpStack e compatível com o sistema de pacotes apt do Ubuntu. Para tanto, usa-se o comando **sudo apt install apt-transport-https dirmngr**.
- Na sequência configura-se a chave para o novo repositório, utilizando o comando: **sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 1CE2AFD36DBCCA00**
- E posteriormente deve ser adicionado o repositório à lista de repositórios criando um novo arquivo, com o comando: **sudo echo "deb https://artifacts.chirpstack.io/packages/3.x/deb stable main" | sudo tee /etc/apt/sources.list.d/chirpstack.list**

- Por fim, atualiza-se o cache do pacote apt, usando-se o comando: ***sudo apt update***
- O próximo passo é instalar o ChirpStack-gateway-bridge através da instrução: ***sudo apt install chirpstack-gateway-bridge***
- Após instalado o ChirpStack-gateway-bridge, deve-se executá-lo utilizando o comando: ***sudo systemctl start chirpstack-gateway-bridge*** e na sequência definir a sua inicialização no boot do sistema, com o comando: ***sudo systemctl enable chirpstack-gateway-bridge***
- O passo seguinte corresponde à instalação do Mosquitto, que é uma implementação do protocolo MQTT. Para tanto, executa-se o comando: ***sudo apt install mosquitto mosquitto-clients***
- Após instalados os pacotes necessários, deve-se configurar o ChirpStack-gateway-bridge, o arquivo de configuração está localizado no diretório: ***/etc/chirpstack-gateway-bridge/***. Nesta pasta deve-se alterar o arquivo ***chirpstack-gateway-bridge.toml***. Na linha onde se encontra a configuração “servers”, deve-se definir o endereço IP do servidor de rede, no lugar do valor default apresentado abaixo.

```
servers=[
  "tcp://127.0.0.1:1883",
]
```

Assim foi concluída a instalação e configuração dos componentes necessários ao funcionamento do gateway.

Toda operação executada pelo Gateway gera um log gravado no arquivo ***chirpstack-gateway-bridge.log***, armazenado na pasta.

Para visualizar e acompanhar os eventos gerados neste arquivo, utiliza-se o comando: ***journalctl -u chirpstack-gateway-bridge -f -n 50***

Instalação do Servidor de Aplicação e de Rede

Para a instalação do servidor de rede foi usado o tutorial disponível em: <https://www.chirpstack.io/network-server/install/debian/>.

Após a instalação, somente é necessário mudar a Banda para AU915 Sub-Band 2. Para isso modifica-se o arquivo de configuração ***chirpstack-network-server.toml***, disponível na pasta: ***/etc/chirpstack-network-server/***. Na linha ***network_server.band.name*** deve-se modificar o atributo ***name="EU868"*** para ***name="AU915"***.

Para a instalação do servidor de aplicação foi usado o tutorial disponível em:

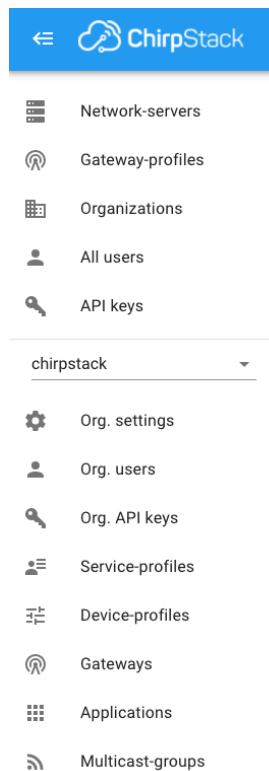
<https://www.chirpstack.io/application-server/install/debian/>.

Para a configuração do gateway e dos dispositivos, basta seguir o tutorial disponível em: <https://www.chirpstack.io/project/guides/first-gateway-device/>.

Para acessar o ambiente web do servidor de aplicação, basta acessar o endereço IP dele, através de um navegador, na porta 8080, por exemplo "<http://127.0.0.1:8080>". Nesse ambiente é possível configurar os **profiles**, **gateway** e **devices** através da interface web disponibilizada.

No primeiro acesso o login e senha são: **admin / admin**

Todas as configurações são acessíveis através do menu lateral exibido após login na página web:



1. Tela de cadastro de um Servidor de Rede:

Network-servers / Add

GENERAL GATEWAY DISCOVERY TLS CERTIFICATES

Network-server name *
Mestrado
A name to identify the network-server.

Network-server server *
localhost:8000
The 'hostname:port' of the network-server, e.g. 'localhost:8000'.

[ADD NETWORK-SERVER](#)

2. Para criar o Gateway Profile, no qual são identificados os canais habilitados para uso.

Gateway-profiles / Create

Name *
Mestrado
A short name identifying the gateway-profile.

Enabled channels *
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 64, 65
The channels active in this gateway-profile as specified in the LoRaWAN Regional Parameters specification. Separate channels by comma, e.g. 0, 1, 2. Extra channels must not be included in this list.

Network-server *
Mestrado

[ADD EXTRA CHANNEL](#) [CREATE GATEWAY-PROFILE](#)

3. Criar o Service Profile, no qual devem ser habilitados os metadados do gateway, para coleta de informações como temperatura e localização, e o data-rate máximo e mínimo.

Service-profiles / Mestrado [DELETE](#)

Service-profile name *
Mestrado

A name to identify the service-profile.

Add gateway meta-data
GW metadata (RSSI, SNR, GW geoloc., etc.) are added to the packet sent to the application-server.

Enable network geolocation
When enabled, the network-server will try to resolve the location of the devices under this service-profile. Please note that you need to have gateways supporting the fine-timestamp feature and that the network-server needs to be configured in order to provide geolocation support.

Device-status request frequency
24

Frequency to initiate an End-Device status request (request/day). Set to 0 to disable.

Report device battery level to application-server

Report device link margin to application-server

Minimum allowed data-rate *
0

Minimum allowed data rate. Used for ADR.

Maximum allowed data-rate *
6

Maximum allowed data rate. Used for ADR.

[UPDATE SERVICE-PROFILE](#)

4. Adicionar o Gateway. Neste momento deve-se recuperar o Gateway ID definido no passo 1 deste guia.

Gateways / Create

GENERAL TAGS METADATA

Gateway name *
Gateway Mestrado
The name may only contain words, numbers and dashes.

Gateway description *
Gateway para os testes

Gateway ID *
28 9b ee 70 84 15 0c 2f **MSB** ↻

Network-server *
Mestrado ▼
Select the network-server to which the gateway will connect. When no network-servers are available in the dropdown, make sure a service-profile exists for this organization.

Gateway-profile
Mestrado ▼
Optional. When assigning a gateway-profile to the gateway, ChirpStack Network Server will attempt to update the gateway according to the gateway-profile. Note that this does require a gateway with ChirpStack Concentrator.

Gateway discovery enabled
When enabled (and ChirpStack Network Server is configured with the gateway discover feature enabled), the gateway will send out periodical pings to test its coverage by other gateways in the same network.

Gateway altitude (meters) *
0
When the gateway has an on-board GPS, this value will be set automatically when the network has received statistics from the gateway.

5. Para criar o Device Profile. Neste momento deve-se definir a versão do protocolo LoRaWAN a ser utilizado na arquitetura, a classe dos dispositivos, o EIRP máximo, permitir que os dispositivos finais trabalhem no modo OTAA/ABP, as configurações de frequências e canais de transmissão, e o CODEC dos pacotes transferidos.

Device-profiles / Mestrado

 **DELETE**

GENERAL JOIN (OTAA / ABP) CLASS-B CLASS-C CODEC TAGS

Device-profile name *
Mestrado
A name to identify the device-profile.

LoRaWAN MAC version *
1.0.3 ▼
The LoRaWAN MAC version supported by the device.

LoRaWAN Regional Parameters revision *
A ▼
Revision of the Regional Parameters specification supported by the device.

Max EIRP *
30
Maximum EIRP supported by the device.

UPDATE DEVICE-PROFILE

Device-profiles / Mestrado [DELETE](#)

GENERAL JOIN (OTAA / ABP) CLASS-B CLASS-C **CODEC** TAGS

Payload codec
Cayenne LPP ▼

By defining a payload codec, ChirpStack Application Server can encode and decode the binary device payload for you.

[UPDATE DEVICE-PROFILE](#)

6. Para criar uma aplicação na qual serão alocados os devices:

Applications / Create

Application name *
Mestrado

The name may only contain words, numbers and dashes.

Application description *
Aplicação para testes

Service-profile *
Mestrado ▼

The service-profile to which this application will be attached. Note that you can't change this value after the application has been created.

[CREATE APPLICATION](#)

7. Para adicionar os Devices e definir as chaves de criptografia a serem usados nos pacotes transmitidos e recebidos. As chaves a serem configuradas variam de acordo com o Device Profile selecionado.

Applications / Mestrado / Devices / Create

GENERAL VARIABLES TAGS

Device name *
Device01

The name may only contain words, numbers and dashes.

Device description *
Dispositivo final para testes 01

Device EUI *
6c 24 5d 62 e1 a8 e0 d4 MSB ↻

Device-profile *
Mestrado ▼

Disable frame-counter validation

Note that disabling the frame-counter validation will compromise security as it enables people to perform replay-attacks.

Device is disabled

ChirpStack Network Server will ignore received uplink frames and join-requests from disabled devices.

[CREATE DEVICE](#)

Após finalizadas as configurações, é possível visualizar os dados do dispositivo trafegando.

The screenshot shows the ChirpStack web interface. The breadcrumb navigation is Applications / weather-station / Devices / rooftop. The left sidebar contains a navigation menu with items like Network-servers, Gateway-profiles, Organizations, All users, and various settings. The main content area is titled 'DEVICE DATA' and shows a list of uplink frames. The first frame is at 2:34:44 PM and is an uplink. Below it, the 'status' frame is expanded, showing the following JSON data:

```

{
  applicationID: "3",
  applicationName: "weather-station",
  battery: 255,
  batteryLevel: 0,
  batteryLevelUnavailable: true,
  devEUI: "0101010101010101",
  deviceName: "rooftop",
  externalPowerSource: false,
  margin: 7
}

```

Below the status frame, another uplink frame is visible at 2:34:15 PM.

APÊNDICE C – CÓDIGO ARDUINO PARA O DISPOSITIVO FINAL ZIGBEE

```

#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
#include <Adafruit_BMP280.h>

char buffer[256];
char xbeedata[4];
unsigned long delayTime;
Adafruit_BME280 bme; // I2C

#define SEALEVELPRESSURE_HPA (1013.25)

void setup() {
  // Serial de debugger
  Serial2.begin(115200);
  //Xbee
  Serial.begin(9600);

  //serial2
  while(!Serial);

  //SETUP BME
  unsigned status;
  delay(2000);
  // You can also pass in a Wire library object like &Wire2
  status = bme.begin(0x76);
  Serial2.println("SETUP DONE");

  delayTime = 60000;
}

void loop() {

  float temperatura = bme.readTemperature();
  Serial2.print("====Temperature====");
  Serial2.print(temperatura);
  Serial2.println(" *C");

  memcpy(xbeedata, &temperatura, 4);
  Serial.print("T");
  Serial.print(temperatura);
  Serial.println("");

  float pressao = bme.readPressure() / 100.0F;
  Serial2.print("====Pressure==== ");
  Serial2.print(pressao);
  Serial2.println(" hPa");
}

```

```

memcpy(xbeedata, &pressao, 4);
Serial.print("P");
Serial.print(pressao);
Serial.println("");

float umidade = bme.readHumidity();
Serial2.print("====Humidity====");
Serial2.print(umidade);
Serial2.println(" %");

memcpy(xbeedata, &umidade, 4);
Serial.print("U");
Serial.print(umidade);
Serial.println("");

float altitude = bme.readAltitude(SEALEVELPRESSURE_HPA);
Serial2.print("====Approx. Altitude==== ");
Serial2.print(altitude);
Serial2.println(" m");

memcpy(xbeedata, &altitude, 4);
Serial.print("A");
Serial.print(altitude);
Serial.println("");

float guva = analogRead(A8)/1024.0*5.0;
Serial2.print("====GUVA====");
Serial2.println(guva); //Write the serial data

memcpy(xbeedata, &guva, 4);
Serial.print("G");
Serial.print(guva);
Serial.println("");

String name = "500mS";
Serial2.print("====NOME====");
Serial2.println(name); //Write the serial data

memcpy(xbeedata, &name, 4);
Serial.print("N");
Serial.print(name);
Serial.println("");

Serial2.print("====ENVIU====");

delay(delayTime);
}

```

APÊNDICE D – CÓDIGO ARDUINO PARA O ROTEADOR ZIGBEE

```
unsigned long delayTime;
int buffer;

void setup() {
  // Serial de debugger
  Serial1.begin(9600);
  //Xbee
  Serial.begin(9600);

  while(!Serial);
}

void loop() {
  //aguarda dados da serial
  while (Serial.available() > 0){
    Serial1.println("Dados recebidos");
    buffer = Serial.read();
    Serial.write(buffer);
    Serial1.println(buffer);
    Serial1.println("Dados enviados");
  }
}
```

APÊNDICE E - CÓDIGO PYTHON PARA O COORDENADOR ZIGBEE

```

from influxdb import InfluxDBClient
from datetime import datetime
import serial
import logging

received_data = False

logging.basicConfig(level=logging.DEBUG, filename='app.log', filemode='w',
format='%(%asctime)s - %(name)s - %(levelname)s - %(message)s')
serial = serial.Serial('/dev/ttyUSB0',9600,timeout=.5)

json_body_t = [
    {
        "measurement": "zigbee_init",
        "tags": {
            "host": "end_devices_lotario",
            "region": "us-west"
        },
        "time": datetime.now(),
        "fields": {
            "value": 0.0
        }
    }
]

client = InfluxDBClient('3.137.140.97', 8086, "", "", 'mestrado_zigbee')

client.create_database('mestrado_zigbee')

while True:
    data_type = serial.readline().strip()

    data_umidade = 0.0
    data_temperatura = 0.0
    data_uv= 0.0
    data_pressao = 0.0
    data_altitude = 0.0
    data_device = "";

    #enquanto nao chegar um dado delimitador vai colhendo os dados

while data_type != b" and data_type != "":
    logging.info('O data_type lido e: ' + data_type)
    data_string = data_type
    type = data_string[0:1]
    logging.info('O tipo e: ' + type)

```

```

if type == 'U':
    data_umidade = data_string[1:]
    logging.info('O valor da umidade lida e: ' + data_umidade)
    received_data = True
elif type == 'T':
    data_temperatura = data_string[1:]
    logging.info('O valor da temperatura lida e: ' + data_temperatura)
    received_data = True
elif type == 'G':
    data_uv = data_string[1:]
    logging.info('O valor de UV lido e: ' + data_uv)
received_data = True
elif type == 'A':
    data_altitude = data_string[1:]
    logging.info('O valor da altitude lida e: ' + data_altitude)
    received_data = True
elif type == 'P':
    data_pressao = data_string[1:]
    logging.info('O valor da pressao lida e: ' + data_pressao)
    received_data = True
elif type == 'N':
    data_device = data_string[1:]
    logging.info('O valor do nome lido e: ' + data_device)
    received_data = True

data_type = serial.readline().strip();

if received_data == True:
    logging.debug('Salvando no banco')
    json_body = [
        {
            "measurement": "end_device_zigbee",
            "tags": {
                "host": "end_devices_lotario",
                "device": data_device
            },
            "time": datetime.now(),
            "fields": {
                "umidade_value": float(data_umidade),
                "temperatura_value": float(data_temperatura),
                "uv_value": float(data_uv),
                "pressao_value": float(data_pressao),
                "altitude_value": float(data_altitude)
            }
        }
    ]

client.write_points(json_body)

#result = client.query('select value from end_device_zigbee;')

```

```
    #print('Result: {0}'.format(result))
else:
    logging.debug('invalid data')

received_data = False
```