

CENTRO ESTADUAL DE EDUCAÇÃO TECNOLOGIA PAULA SOUZA

FACULDADE DE TECNOLOGIA DE INDAIATUBA

DR. ARCHIMEDES LAMMOGLIA

BRUNA STRAZZA QUINTERO

**IMPLEMENTAÇÃO DE UMA REDE LORAMESH PARA
APLICAÇÃO NA AGRICULTURA 4.0**

Indaiatuba

Junho 2024

CENTRO ESTADUAL DE EDUCAÇÃO TECNOLOGIA PAULA SOUZA

FACULDADE DE TECNOLOGIA DE INDAIATUBA

DR. ARCHIMEDES LAMMOGLIA

BRUNA STRAZZA QUINTERO

**IMPLEMENTAÇÃO DE UMA REDE LORAMESH PARA
APLICAÇÃO NA AGRICULTURA 4.0**

Projeto de Trabalho de Graduação apresentado por Bruna Strazza Quintero como pré-requisito para a conclusão do Curso Superior de Tecnologia em Redes de Computadores, da Faculdade de Tecnologia de Indaiatuba, elaborado sob a orientação do Prof. Welligton Roque.

Indaiatuba

Junho 2024

CENTRO ESTADUAL DE EDUCAÇÃO TECNOLOGIA PAULA SOUZA

FACULDADE DE TECNOLOGIA DE INDAIATUBA

DR. ARCHIMEDES LAMMOGLIA

CURSO DE TECNOLOGIA EM REDES DE COMPUTADORES

BRUNA STRAZZA QUINTERO

Banca Avaliadora:

Wellington Roque	Orientador
Sergio Gustavo Medina Pereira	Banca Avaliadora
Giovana Fadini de Oliveira	Banca Avaliadora

Data da Defesa: 02/07/2024

AGRADECIMENTOS

Primeiramente, agradeço a Deus por guiar cada passo desta jornada acadêmica. À minha família querida, Paulo, Andreia, Camila e Giovana, vocês foram meu suporte constante, sempre me incentivando a alcançar meus objetivos. Aos meus amigos de classe, por compartilharem comigo as alegrias desta conquista e estarem ao meu lado nos desafios.

Um agradecimento especial ao meu noivo, Marcos, por seu amor incondicional e apoio constante ao longo dessa caminhada.

Aos meus professores da Fatec de Indaiatuba, pela dedicação em transmitir conhecimento e pela orientação valiosa que moldou meu aprendizado.

E, por último, mas não menos importante, ao meu orientador, Prof. Me. Wellington Roque, pela orientação experiente, paciência e pelo suporte fundamental na elaboração deste trabalho de graduação.

A todos vocês, meu mais sincero agradecimento por fazerem parte desta conquista significativa em minha vida.

EPÍGRAFE

*“A diferença entre o impossível e o possível reside
na determinação de uma pessoa”*

Tommy Lasorda.

RESUMO

O LoRaMESH, que utiliza a tecnologia LoRa®, é uma solução avançada para a Agricultura 4.0, permitindo a coleta eficiente de dados em áreas remotas e de difícil acesso. Além de oferecer comunicação com baixo consumo de energia, o dispositivo também realiza o roteamento dos dados de forma eficaz, contribuindo para melhorar a produtividade e a sustentabilidade no setor agrícola. Com isso, o objetivo deste projeto é configurar e implementar uma rede LoRaMESH utilizando o *Radioenge* RD42C. Para atingir esse objetivo, dois rádios foram configurados para serem utilizados em uma rede Mesh usando o *software* da *Radioenge*. Além disso, o rádio LoRaMESH disponibiliza integrado no seu *hardware*, portas de entrada e saída digitais (*GPIO – General Purpose Input Output*) e portas analógicas que podem ser configuradas para leitura de sensores e controle de atuadores. Após os testes de configuração do rádio utilizando o *software* da *Radioenge* foi utilizando um microcontrolador Arduino Uno, que ao ser programado permitiu a realização de testes de comunicação entre os rádios na rede Mesh e das portas analógicas e digitais do *Radioenge*. Com os resultados da configuração do rádio pode-se observar o correto funcionamento da rede Mesh e a visualização dos dados das portas digitais e analógicas. Com relação aos testes realizados utilizando o Arduino foi possível estabelecer e analisar a comunicação dos dispositivos e a troca de dados das portas digitais e analógicas. Conclui-se que a com a realização dos testes de comunicação entre os rádios utilizando a rede Mesh os dispositivos demonstraram um funcionamento estável viabilizando o envio de dados das portas digitais e analógicas permitindo a realização do monitoramento e controle de sensores e equipamentos na rede Mesh.

Palavras-chave: redes de sensores sem fio, agricultura de precisão, tomada de decisão.

ABSTRACT

LoRaMESH, which utilizes LoRa® technology, is an advanced solution for Agriculture 4.0, enabling efficient data collection in remote and hard-to-reach areas. In addition to offering low-power communication, the device also effectively routes data, contributing to improved productivity and sustainability in the agricultural sector. Therefore, the aim of this project is to configure and implement a LoRaMESH network using the *Radioenge* RD42C. To achieve this objective, two radios were configured to be used in a Mesh network using *Radioenge*'s *software*. Additionally, the LoRaMESH radio integrates digital input and output ports (GPIO – General Purpose Input Output) and analog ports that can be configured for sensor reading and actuator control. After configuring the radio using *Radioenge*'s *software*, an Arduino Uno microcontroller was used, which, when programmed, enabled communication tests between the radio in the Mesh network and the analog and digital ports of the *Radioenge* device. The configuration results of the radio showed the correct functioning of the Mesh network and the visualization of data from the digital and analog ports. Regarding the tests conducted using the Arduino, it was possible to establish and analyze the communication between devices and the exchange of data through the digital and analog ports. It was concluded that the communication tests between the radio using the Mesh network demonstrated stable operation, making it feasible to send data from the digital and analog ports, thus enabling the monitoring and control of sensors and equipment within the Mesh network.

Keywords: wireless sensor networks, precision agriculture, decision making.

LISTA DE ILUSTRAÇÕES

Figura 1: Apresentação das etapas da agricultura de Precisão	21
Figura 2: Classificação de sistemas de rádio	24
Figura 3: Demonstração de tipo de ondas eletromagnéticas	25
Figura 4: Infraestruturas WMNs.....	27
Figura 5: Visualização da Placa Arduino UNO	28
Figura 6: Pinagem do Arduino Uno – Diagrama.....	29
Figura 7: Frequência de operação LoRa.....	32
Figura 8: Camadas do LoRa Informações das Transmissões	32
Figura 9: Camada de Aplicação LoRaWAN.....	35
Figura 10: Recebimento do pacote na primeira janela Classe A	36
Figura 11: Recebimento do pacote na segunda janela Classe A.....	36
Figura 12: Nenhum pacote recebido Classe A.....	37
Figura 13: Transmissão de um pacote classe B	37
Figura 14: Janela de recebimento Classe C	38
Figura 15: Pinagem do dispositivo LoRaWAN	39
Figura 16: Arquitetura de rede LoRaWAN.....	40
Figura 17: Visualização do dispositivo LoRaMESH Radioenge - RD24C	42
Figura 18: Visualização dos dispositivos LoRa Radioenge - RD29C e RD42C	43
Figura 19: Diagrama de rede LoRaMESH	43
Figura 20: Pinout numerada do LoRaMESH	44
Figura 21: Representação Classe A e Classe C.....	47
Figura 22: Dispositivos físicos do projeto	50
Figura 23: Demonstração de comunicação entre os dispositivos	51
Figura 24: Localizando software no site Radioenge	52
Figura 25: Primeiras impressões com o Configurador LoRaMESH – v2r7p0.....	53
Figura 26: Montagem circuitos RD42C + FTDI232	53
Figura 27: Conectando a porta COM	54
Figura 28: Visualização das informações da 1ª placa.....	55
Figura 29: Configuração do BW, SR e CR.....	56
Figura 30: Configuração a classe.....	57
Figura 31: Configuração o ID Master.....	57
Figura 32: Configuração da senha	58

Figura 33: Visualização das informações da 2ª placa.....	59
Figura 34: Configuração do BW, SR e CR do slave.....	60
Figura 35: Configuração da classe do slave	60
Figura 36: Configuração ID e senha do slave.....	61
Figura 37: Teste de conexão remota do master para o slave	62
Figura 38: Teste do RSSI do master até o slave	63
Figura 39: Montagem circuito RD42C + FTDI232 + Potenciômetro	64
Figura 40: Leitura da porta AD do Configurador LoRaMESH.....	64
Figura 41: Visualização do formato do pacote enviado com o comando 0XC2	65
Figura 42: Visualização do formato do pacote recebido com o comando 0XC2	65
Figura 43: Leitura da porta AD do Configurador LoRaMESH.....	66
Figura 44: Montagem circuito RD42C + Arduino + LED	67
Figura 45: Programação da configuração no IDE Arduino para o dispositivo Master.....	67
Figura 46: Programação na IDE Arduino para o dispositivo master.....	68
Figura 47: Montagem circuito RD42C + Sensor de Umidade do solo + Arduino	68
Figura 48: Programação da configuração no IDE Arduino para o dispositivo slave	69
Figura 49: Recebimento dos dados do slave para o master.....	70

LISTA DE QUADROS

Quadro 1: Lista de definição dos pinos LoRaWAN	39
Quadro 2: Lista de Comando AT	41
Quadro 3: Valores Máximos e Absolutos	44
Quadro 4: Definição dos pinos LoRaMESH	45
Quadro 5: Taxa de dados LoRaMESH	45

LISTA DE ABREVIACOES

AD	Analogico/Digital
AM	Amplitude Modulada
ANATEL	Agencia Nacional de Telecomunicaoes
AP	Agricultura de precisao
APs	<i>Access Points</i>
ARPA	<i>Advanced Research Projects Agency</i>
ARPANET	<i>Advanced Research Projects Agency Network</i>
BNDES	Banco Nacional de Desenvolvimento Econmico e Social
BW	<i>Bandwidth</i>
CAGR	<i>Compound Annual Growth Rate</i>
CR	<i>Code Rate</i>
CSS	<i>Chirp Spread Spectrum</i>
DARPA	<i>Defense Advanced Research Projects Agency</i>
dBm	<i>Decibel-milliwatt</i>
dBi	<i>Decibel-isotropic</i>
DCA	<i>Defense Communications Agency</i>
FM	Frequncia Modulada
GPIOs	<i>General Purpose Input/ Output</i>
HF	<i>High Frequency</i>
IDE	<i>Integrated Development Environment</i>
ISM	<i>Industrial Scientific and medical</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IoT	<i>Internet of Things</i>
LoRa	<i>Longe Range</i>

LoRaWAN	<i>Long Range Wide-Area Network</i>
LPWAN	<i>Low-Power Wide-Area Network</i>
M2M	<i>Machine to Machine</i>
MAC	<i>Media Access Control</i>
MAP	<i>Mesh Access Point</i>
MC	<i>Mesh Client</i>
MCTIC	Ministério da Cultura Tecnologia, Inovações e Comunicações
MGW	<i>Mesh Gateway</i>
MR	<i>Mesh Router</i>
PHY	<i>Physical Layer</i>
PWM	<i>Pulse Width Modulation</i>
RSSI	<i>Received Signal Strength Indication</i>
SF	<i>Spreading Factor</i>
TCP	<i>Transmission Control Protocol</i>
TIC	Tecnologias da Informação e Comunicação
UART	<i>Universal Asynchronous Receiver-Transmitter</i>
UHF	<i>Ultra high Frequency</i>
USB	<i>Universal Serial Bus</i>
VHF	<i>Very High Frequency</i>
WLAN	<i>Wireless Local Area Network</i>
WMAN	<i>Wireless Metropolitan Area Network</i>
WMN	<i>Wireless Mesh Network</i>
WPAN	<i>Wireless Personal Area Network</i>
WWAN	<i>Wireless Wide Area Network</i>

SUMÁRIO

INTRODUÇÃO	14
CAPÍTULO I	17
FUNDAMENTAÇÃO TEÓRICA	17
1.1. Internet Of Things	17
1.1.1. Dispositivos IoT	18
1.2. Agricultura 4.0.....	20
1.3. Redes Sem Fio.....	22
1.3.1. Tipos de redes sem fio	23
1.4. Redes MESH	26
1.5. Arduino.....	28
1.5.1. Pinout Arduino.....	29
1.6. Protocolo TCP/IP.....	30
1.7. LoRa	31
1.7.1. Topologia Básica	31
1.7.2. Parâmetros de transmissão.....	32
1.7.2.1. Frequência de Operação	33
1.7.2.2. Largura de banda	33
1.7.2.3. Potência de Transmissão	33
1.7.2.4. Fator de Espalhamento	34
1.7.2.5. Taxa de Código.....	34
1.7.3. LoRaWAN	35
1.7.3.1. Classes.....	35
1.7.3.2. Pinout.....	38
1.7.3.3. Arquitetura de rede LoRaWAN.....	40
1.7.3.4. Comando AT.....	41
1.7.4. LoRaMESH	42
1.7.4.1. Gerenciamento do LoRaMESH.....	43
1.7.4.2. Tensão dos dispositivos	44
1.7.4.3. Pinout da placa RD42C	44
1.7.4.4. Taxa de dados	45

1.7.4.5. Modo de Operação	46
1.7.4.6. Protocolo Modbus	47
1.8. Trabalhos Relacionado	48
CAPÍTULO II.....	49
METODOLOGIA.....	49
2.1. Caracterização Da Pesquisa.....	49
2.2. Materiais para implantação do projeto	49
2.3. Procedimentos para coleta e análise de dados	50
2.3.1. Técnicas para coleta de dados	51
2.3.2. Natureza da análise de dados.....	52
2.3.3. Testes e implementação do código no projeto.....	52
2.3.3.1. Baixando <i>software</i> Configurador LoRaMESH – v2r7p0	52
2.3.3.2. Montagens FTDI232 e RD42C	53
2.3.3.3. Configuração o dispositivo <i>Master</i>	54
2.3.3.4. Configuração do dispositivo <i>Slave</i>	58
2.3.3.5. Testes no <i>software</i> do Configurador LoRaMESH – v2r7p0.....	61
2.3.3.6. Aplicação de configurações do Arduino do dispositivo <i>Master</i>	66
2.3.3.7. Aplicação de configurações do Arduino do dispositivo <i>Slave</i>	68
2.3.4. Resultados e Discussão.....	70
CAPÍTULO III	72
3. Conclusão	72
BIBLIOGRAFIA.....	73

INTRODUÇÃO

A tecnologia vem se expandindo significativamente em diversas áreas. Podemos nos definir dependentes desse manifesto, em virtude da praticidade no cotidiano. Com base nesta realidade, podemos observar diante a sociedade, que é de grande importância que as tecnologias funcionem com eficiência e com rapidez.

A Internet das Coisas, conhecida mais por seu nome simbólico IoT, cresce continuamente em quesito de conectar diversos dispositivos à rede em nós, e com isso obter dados sobre eles em tempo real (SILVA e ESPEJO, 2020). O principal objetivo da implementação da tecnologia Internet das coisas é facilitar cada vez mais a interação entre o humano e a máquina para gerenciar, alertar, priorizar tarefas do dia a dia. De acordo com um artigo publicado na revista "*IEEE Internet of Things Journal*", a IoT desempenha um papel fundamental em uma variedade de setores, incluindo agricultura, saúde, transporte e cidades inteligentes. (ZHANG E TAO, 2020)

Uma pesquisa conduzida em colaboração entre o Banco Nacional de Desenvolvimento Econômico e Social (BNDES) e o Ministério da Cultura, Tecnologia, Inovações e Comunicações (MCTIC) aponta que a implementação da Internet das Coisas (IoT) na agricultura poderá resultar em um crescimento econômico global de aproximadamente R\$ 350 bilhões até o ano de 2025 (BNDES,2020).

Em um estudo citado no artigo "*The Impact of Internet of Things on Agriculture: A Systematic Review*", destaca-se o potencial da IoT na agricultura. Com a implementação de sensores e dispositivos conectados em fazendas, os agricultores podem coletar dados em tempo real sobre o clima, solo, irrigação e crescimento das plantas. Essas informações permitem uma gestão mais eficiente e precisa das operações agrícolas, otimizando o uso de recursos, reduzindo custos e aumentando a produtividade. (FAROOQ et al, 2020)

As redes sem fio e IoT é uma ótima estratégia para transferência de dados em campo, diminuindo os gastos em fiação e manutenção. A comunicação de redes sem fio é conhecida como uma conexão instável e não segura, entretanto conforme o seu desenvolvimento, vem se tornando umas das conexões mais seguras e flexíveis, garantindo integridade dos dados transmitidos. (ESTES, 2018)

Rede mesh é um tipo de rede sem fio que aprimora a robustez e a cobertura das conexões, na qual cada nó atua tanto como um roteador quanto como um receptor, permitindo

que os dados encontrem automaticamente o melhor caminho até o destino, mesmo em caso de falhas em alguns nós. Isso proporciona uma comunicação mais confiável e eficiente, essencial para aplicações de IoT que exigem alta disponibilidade e baixa latência. (AKYILDIZ; WANG, 2009).

A tecnologia denominada para fazer com que as informações cheguem de forma íntegra através dos dispositivos IoT, é a tecnologia LoRa, uma abreviação do termo "*Long Range*" (ou, em língua portuguesa, "Longo Alcance"), constitui-se como uma solução inovadora no âmbito das comunicações sem fio de baixa potência. A sua finalidade primordial reside na viabilização da conectividade de dispositivos pertencentes ao universo da Internet das Coisas (IoT) em intervalos de distância substanciais, logrando superar barreiras físicas e, simultaneamente, otimizando o consumo energético.

O dispositivo RD42C ou LoRAMEESH é baseado no protocolo LoRa e ideal para aplicações IoT, proporciona uma comunicação de longo alcance e um baixo consumo de energia. A sua funcionalidade permite que cada nó da rede se comunique diretamente com outros nós, criando uma rede mesh. Além disso, a tecnologia permite oferecer confiabilidade na transmissão de dados, contribuindo para a integridade e segurança das informações trocadas entre os dispositivos.

Com base nos conceitos abordados acima, a questão de pesquisa que norteará essa pesquisa é a seguinte: Quais recursos podem ser utilizados para facilitar o colhimento de dados em campo de forma autônoma? E meio de comunicação seria mais adequado e eficiente para garantir a integridade dos dados coletados?

Levando em consideração a tese que estamos relatando, a hipótese deste trabalho consiste na combinação de duas tecnologias, IoT e LoRAMEESH, em virtude de aprimorar a agricultura de precisão. Definindo pontos importantes no monitoramento contínuo de dados agrícolas, como clima, solo e crescimento das plantas, e como essas informações podem ser coletadas e processadas.

Esta pesquisa tem como objetivo conceber e implementar um projeto que utilize o dispositivo LoRAMEESH RD42C e criar uma rede Mesh, para estabelecer comunicação e transferência de dados em áreas agrícolas, visando atender às necessidades dos agricultores e diminuir seus custos, auxiliando-os em suas tomadas de decisões com base nos dados coletados.

Para cumprir com o objetivo, os dispositivos RD42C tiveram suas configurações e implementações para criar uma rede Mesh, na qual possui uma grande vantagem por ter 8 portas de entradas e saída digitais (GPIO) e 2 portas analógicas que facilita a configuração e adaptação de sensores e atuadores.

Este trabalho trata-se de uma pesquisa de natureza descritiva e exploratória, que segundo Gil (2002), uma pesquisa de natureza descritiva e exploratória combina a descrição detalhada de características de uma população ou fenômeno, utilizando métodos padronizados como questionários e observações, com a investigação inicial de questões pouco conhecidas, empregando revisão bibliográfica, entrevistas e análise de exemplos práticos. Essa abordagem permite tanto entender detalhadamente o objeto de estudo quanto identificar novas questões para futuras pesquisas.

Este estudo está estruturado da seguinte forma: no Capítulo I encontra-se a fundamentação teórica, que aborda os conceitos chave que dão embasamento a esta pesquisa, e apresenta os trabalhos acadêmicos correlatos ao presente tema. No Capítulo II, são abordadas as características da pesquisa, os procedimentos para coleta e análise de dados, os materiais utilizados, os testes realizados e as configurações adotadas durante o desenvolvimento do projeto, além dos resultados alcançados. Por fim, no Capítulo III, apresenta-se a conclusão obtida a partir dos resultados gerados.

CAPÍTULO I

Fundamentação Teórica

Neste primeiro capítulo será apresentado os conceitos sobre agricultura 4.0, agricultura de precisão, *Internet Of Things*, dispositivos e sensores que fazem parte do sistema IoT, redes sem fio, redes Mesh, Tecnologia LoRa e LoRaMESH.

1.1. Internet Of Things

Internet Of Things (IoT) traduzido para o português significa Internet das Coisas, e pode-se defini-lo como coisas ou objetos físicos que são conectados e vinculados a internet, em virtude de fazer com que todos os dispositivos conversem. O termo Internet das Coisas foi citado pela primeira vez por Kevin Asthon em 1999, que teve como principal fundamento, otimizar o tempo e recursos. (KRAMP,2013)

A IoT conecta dispositivos e sensores para coletar e trocar dados, melhorando a eficiência em vários setores. Na agricultura, otimiza o uso de recursos; na saúde, permite monitoramento remoto de pacientes; no transporte, melhora a gestão do tráfego; e em cidades inteligentes, monitora infraestruturas urbanas. A integração com computação em nuvem e inteligência artificial potencializa essas capacidades, permitindo análise em tempo real de grandes volumes de dados. (ZHANG E TAO, 2020)

Podemos criar qualquer coisa desde tão pequeno como o tamanho de um botão até mesmo do tamanho de um prédio em objeto inteligentes, através da utilização de *chips* nos dispositivos podemos interligá-los uns aos outros e assim eles podem trocar dados em tempo real. (FELIX, 2022).

Segundo Felix (2022) em seu artigo “IoT (Internet das Coisas)”. As principais vantagens da Internet das Coisas são:

- **Automação** – Conduz a padronização das tarefas, mantendo a qualidade do serviço e controle das tarefas que são realizadas no dia a dia. A comunicação M2M (Máquina – a – Máquina) ajuda na transparência do processo.

- **Eficiência** – Conforme a comunicação M2M, faz com que as pessoas foquem e se concentrem em suas atividades principais, deixando as atividades repetitivas com as máquinas.
- **Redução de Custo** – Faz uma ótima otimização da utilização de energia e de seus recursos, eliminando quais quer problema no sistema.
- **Comunicação** – A tecnologia IoT permite que os dispositivos se comuniquem melhor conforme estejam ligados, assim trocando informações entre eles.
- **Acesso Instantâneo a dados** – As informações são armazenadas, na qual podem ser analisadas e simplificando o processo na tomada de decisão.

1.1.1. Dispositivos IoT

Os dispositivos IoT são os objetos inteligentes na qual se comunicam entre si, auxiliando e facilitando nas tarefas diárias das pessoas que necessitam da tecnologia.

Segundo Gallegos (2020) existem diversos Dispositivos no mercado, como:

- **Dispositivos Domésticos Inteligente:** estes dispositivos estão voltados para o uso doméstico, auxiliando o consumidor no seu dia a dia, exemplo: fechadura eletrônicas, geladeiras e alto falantes inteligentes entre diversos outros dispositivos que existe no mercado.
- **Sensores Industriais:** Estes são dispositivos que a empresa usa em prol da produtividade e praticidade na coleta de dados, podendo ser monitorados e assim encontrar e solucionar problemas.
- **Automóveis Inteligentes:** Estes são dispositivos voltados à automóveis, assim garantindo cada vez mais a segurança e facilidade no trânsito diário.
- **Câmeras Inteligentes:** Estes são dispositivos que podem ser utilizados tanto na área empresarial quanto doméstica. A utilização destes aparelhos serve para auxiliar no reconhecimento e detecção de pessoas, assim alertando o consumidor.
- **Robôs de Fabricação:** São robôs com a tecnologia IoT integrada, na qual podem ser controlados e programados remotamente.
- **Dispositivos de saúde / condicionamento físico:** Estes são dispositivos que são focados a saúde, ajudando o consumidor ter um direcionamento de como ela está percorrendo a cada dia.

Conforme os dispositivos IoT são interligados é realiza-se uma coleta e o compartilhamento de dados entre eles. Observando esse fluxo de comunicação, nota-se que o comportamento possui vários nós na rede.

1.2. Agricultura 4.0

A agricultura 4.0 (Agro 4.0) ou agricultura digital é constituído por dois mundos que são totalmente conectados, o físico e o virtual. Segundo Massruhá et al (2017), em seu artigo “Agro 4.0 – Rumo à agricultura Digital”, o agro utiliza métodos computacionais que podem coletar uma gama de informações para serem analisadas fazendo com que o consumidor realize tomadas de decisões assertivas. Os dispositivos que podem nos disponibilizar essas informações são os sensores, comunicação de máquina para máquina (M2M), conectividade entre dispositivos moveis e computação em nuvem.

Segundo Viola et al (2022), em seu artigo “Agricultura 4.0 e mudanças climáticas no Brasil”, apontam que as tecnologias digitais ou tecnologias da informação e comunicação (TIC), iniciou a partir dos anos 1990, com o lançamento da agricultura 3.0, que agregou no gerenciamento agropecuário e das fazendas. A agricultura 4.0 é um termo que utiliza tecnologias avançadas, na qual podemos assemelhar a indústria 4.0, com isso ela visa melhorar a eficiência e a sustentabilidade. Com os dispositivos IoT coletam dados em tempo real, como exemplo monitoramento de gado e estufas inteligentes, bem como robôs, impressão 3D de alimentos, drones e inteligência artificial para otimizar operações agrícolas, desde o seu plantio até sua colheita.

Segundo Ferreira et al (2023), o crescimento constante da população, faz com que exista uma grande necessidade de água potável e alimentos, e isso resulta em impactos climáticos globais. Com isso, há uma expectativa de que os métodos agrícolas passem por uma reformulação visando a conservação de água, assim, adotando tecnologias fundamentadas nos princípios de Internet das Coisas e Agricultura 4.0. Entretanto, ainda há uma parcela significativa de agricultores que relutam em fazer essa transição, seja devido à falta de acesso à internet, falta de educação formal, ou simplesmente por serem tradicionalistas.

1.2.1. Agricultura de precisão

Agricultura de precisão ou AP é um conjunto de tecnologias e ferramentas aplicada para melhorias em um sistema de gerenciamento agrícola, usada pelos agricultores em prol de trazer mais eficiência, com isso é realizado a divisão de suas terras para que possa monitorá-las e administrá-las separadamente. Segundo Mantovani et al (1998), a agricultura de precisão é

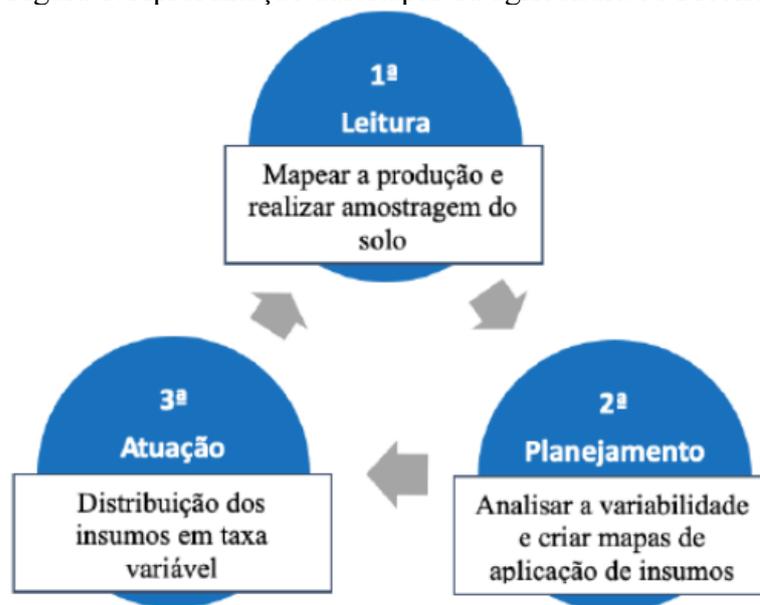
reconhecida quando o agricultor define um local e possui técnicas de monitoramento e acesso as atividades agrícolas dela, assim evitando prejuízos e trabalhando de forma sustentável.

A agricultura de precisão é conhecida desde o início da Revolução Industrial, porém os fundamentos que conhecemos da agricultura de precisão moderna iniciou no século XX, e somente em 1980 tornou viável para os produtores. Com o avanço dos microcomputadores, sensores e *softwares* era possível enxergar os resultados de forma flexível e eficiente, fazendo com que diminuísse perdas na agricultura, obtendo dados de pequenas áreas e ajudando os produtores a tomarem decisões mais assertivas. Em 1990 iniciou a aplicação no Brasil importando as primeiras colhedoras equipadas com motores de produtividade.

O principal objetivo da AP é otimizar e facilitar aplicações de insumos em local correto, aplicando quantidades adequadas e analisando o momento apropriado. A agricultura de precisão também tem como foco diminuir os custos e o impacto ambiental. Vale ressaltar que a AP não considerada apenas por uso das mais avançadas tecnologias, ela pode ser iniciada com base nos recursos que estão disponíveis que futuramente podem ser aprimoradas.

Segundo Silva e Mann existem três etapas que pode dividir a Agricultura de precisão que estão sendo apresentados na figura 1, eles são: Leitura, Planejamento e Atuação.

Figura 1: Apresentação das etapas da agricultura de Precisão



(Fonte: SILVA e MANN, 2020)

Segundo a mesma autora, o processo é contínuo, após chegar na 3ª etapa o processo reinicia para que novos dados sejam gerados, permitindo um crescimento histórico deles.

1.3. Redes Sem Fio

A redes sem fios tem tornado cada vez mais importante para todos, desde empresas quanto pessoas. A velocidade de transmissões de informações faz com que as pessoas acabem se tornando dependente da tecnologia.

Rede sem fio começou a ser traçada em 1865 por James Clerk Maxwell, quando publicou seu artigo “Uma teoria dinâmica de campo eletromagnético”, estabelecendo uma base de início para redes sem fio entre outros fatores. (MAXWELL, 1865) Em 1886, Heinrich Hertz realizou diversos experimentos a fim de provar a existência das ondas eletromagnéticas. (WU, 2022)

Em 1890 Guglielmo Marconi teve como papel principal de construir a primeira estação de rádio comercialização de telegrafia do mundo, em virtude de enviar sinais por quilômetros. No final da década de 1890 estudiosos da área, viam maneiras de como transmitir qualquer tipo de dado de qualquer distância.

A primeira transmissão de voz e música por ondas de rádio ocorreu em dezembro de 1906 em Massachusetts, Estados Unidos, realizada pelo canadense Reinald Fessenden, que reproduziu conversas e música por uma hora para radioamadores. As primeiras experiências com rádio exigiam dispositivos semelhantes a fones de ouvido e os receptores eram feitos de sulfeto de chumbo e cristais, dificultando a sintonia das estações. A massificação do rádio só ocorreu após 1927, devido às dificuldades técnicas e à Primeira Guerra Mundial (1917), que limitou sua difusão. Após a guerra, o interesse pelo rádio cresceu, levando os governos a monitorarem e utilizarem as transmissões, resultando na abertura de mais emissoras, chegando a 550 em 1922. (ESTES, 2018)

Desde 1927, as redes sem fio evoluíram significativamente. Inicialmente, o rádio AM (amplitude modulada) e FM (Frequência Modulada) popularizou a transmissão de áudio, seguido pela televisão, que trouxe imagens e som aos lares. Nos anos 1980 e 1990, a telefonia móvel surgiu, revolucionando as comunicações pessoais. Em 2000 a introdução do Wi-Fi, permitiu conexões de internet sem fio, transformando a maneira como acessamos informações e nos comunicamos (ESTES, 2018). A evolução do 3G para o 4G e, recentemente, o 5G, trouxe maior velocidade e capacidade de dados, suportando uma ampla gama de dispositivos e serviços conectados, desde smartphones até a Internet das Coisas (IoT). Atualmente, as redes sem fio

são fundamentais para a conectividade global, impulsionando avanços em diversas áreas, como comunicação, entretenimento, saúde e automação industrial.

Redes sem fio pode ser definida como um nó de conexões que não necessita de nenhum tipo de cabeamento, uma conexão entre computadores por meio de várias ondas do espectro eletromagnético.

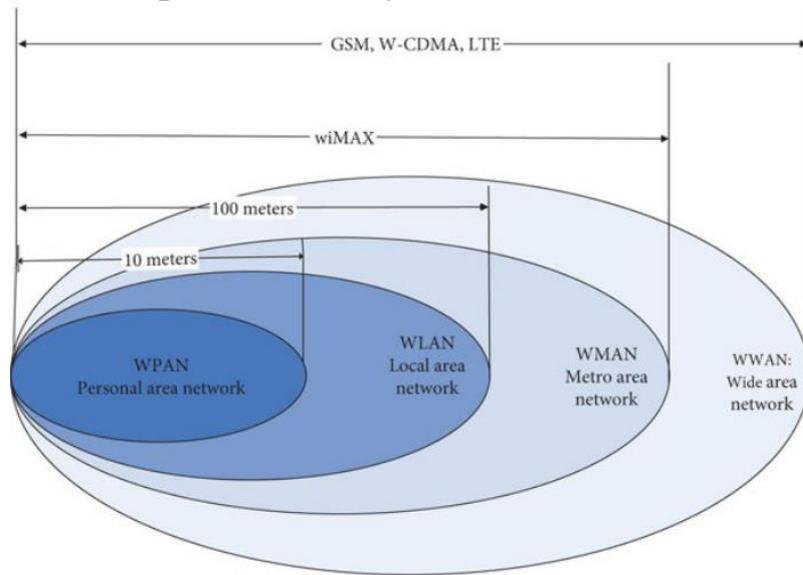
1.3.1. Tipos de redes sem fio

Os tipos de redes podem ser classificados em dois tipos, sendo eles: alcance e frequência.

- Alcance
 - *Wireless Personal Area Network* (WPAN) ou Redes de área pessoal sem fio, permite comunicação sem fio para dispositivos específicos, tem um alcance aproximadamente de 10 metros, podendo ser usado por uma ou duas pessoas, que estão próximas. Utiliza tecnologias como *bluetooth* e infravermelho
 - *Wireless Local Area Network* (WLAN) ou Rede Local sem fio, é uma tecnologia de radiofrequência que facilita a conexão de dispositivos. A capacidade de atingir distâncias grandes, se diferencia em campo, pois possibilita transmissões em tempo real para um terminal central.
 - *Wireless Metropolitan Area Network* (WMAN) ou Rede sem fio em área metropolitana, é uma tecnologia usada para cidades, consideradas como uma conexão de alta velocidade e de longa distância, capazes de cobrir até 20 quilômetros, mesmo assim tende de apresentar uma menor taxa de erros e latência do que outras redes.
 - *Wireless Wide Area Network* (WWAN) ou Rede de área ampla sem fio, é uma tecnologia de rede sem fio com uma velocidade muito maior que as outras, transferindo dados em grandes distâncias. Exemplos desta tecnologia são: GPRS, GSM, 3G, 4G ou 5G.

Na figura 2 é apresentado o comportamento das dimensões dos tipos de redes sem fio

Figura 2: Classificação de sistemas de rádio

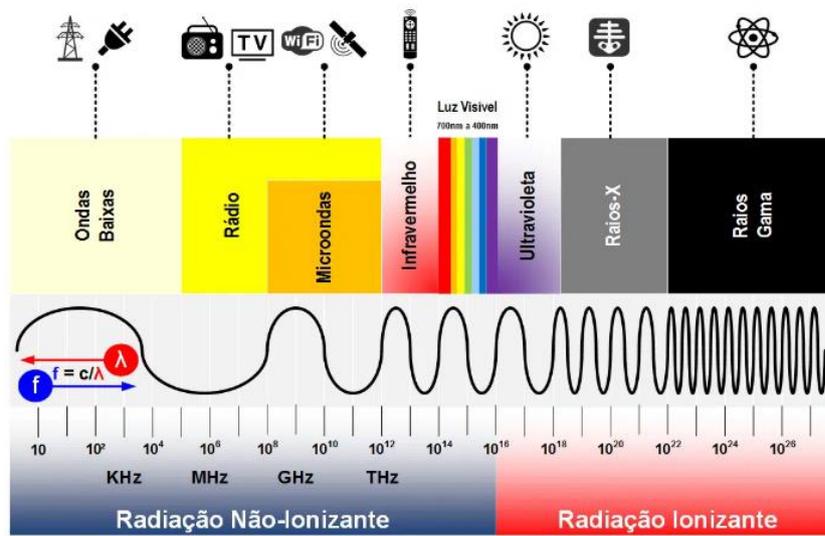


(Fonte: WU, 2022)

- Frequência
 - Microondas Terrestre: São radiações eletromagnéticas de 1 GHz (*Gigahertz*) até 300 GHz que, utilizando antenas parabólicas com cerca de 3 metros de diâmetro, conectam prédios ou cidades sem a necessidade de atravessar cabos.
 - Microondas via satélite: É uma tecnologia que permite uma conexão em diferentes pontos do planeta, opera a partir da ligação entre duas ou várias estações rádios-base e por meio de um satélite em órbita. Os satélites têm suas bandas de frequência específicas, tendo um alcance e velocidade muito grande. Cada sinal enviado de uma estação de rádio base é direcionado ao satélite, com o objetivo de amplificar o sinal e retransmitir para uma outra estação de rádio.
 - Infravermelho: É um tipo de radiação eletromagnética com comprimentos de onda de cerca de 700 nanômetros (nm) a 1 milímetro (mm) e atingindo uma frequência de transmissão de dados de cerca de 300 GHz a 430 THz (*Terahertz*).
 - Ondas de rádio. Essa tecnologia utiliza ondas em várias frequências para emitir e receber sinais de informação, utilizando pouca energia em suas transportações. Possui uma frequência menor, utilizando 10.000 Hz (*Hertz*), e um maior comprimento em ondas, variando 1000 m. Exemplos de ondas de rádio: AM, FM, HF (*High Frequency*), VHF (*Very High Frequency*), UHF (*Ultra High Frequency*) etc.

A figura 3 é possível visualizar os níveis de frequência e para que tipo é utilizado.

Figura 3: Demonstração de tipo de ondas eletromagnéticas



(Fonte: BRITO; 2013)

1.4. Redes MESH

As redes Mesh, redes em malha sem fio ou *Wireless Mesh Network* (WMN), têm sido consideradas uma solução benéfica para fornecer infraestrutura sem fio, em comparação com as arquiteturas de redes sem fio tradicionais baseadas no padrão IEEE (*Institute of Electrical and Electronics Engineers*) 802. 11. WMN é uma rede sem fio de múltiplos saltos, na qual roteadores sem fio encaminham tráfego em benefício de outros roteadores ou de clientes (AKYILDIZ; WANG, 2009).

Rede Mesh foi criada por norte-americanos militares da *Defense Advanced Research Projects Agency* (DARPA) no ano de 1958. O principal propósito, era conceder intercomunicação ponto a ponto entre vários aparelhos, descartando a capacidade para formar uma conversação diferente nova com um ponto no centro da rede.

O conceito de rede Mesh começou a ser utilizada de 1970 e 1980, onde a robustez e a redundância eram essenciais para assegurar a comunicação em situações adversas.

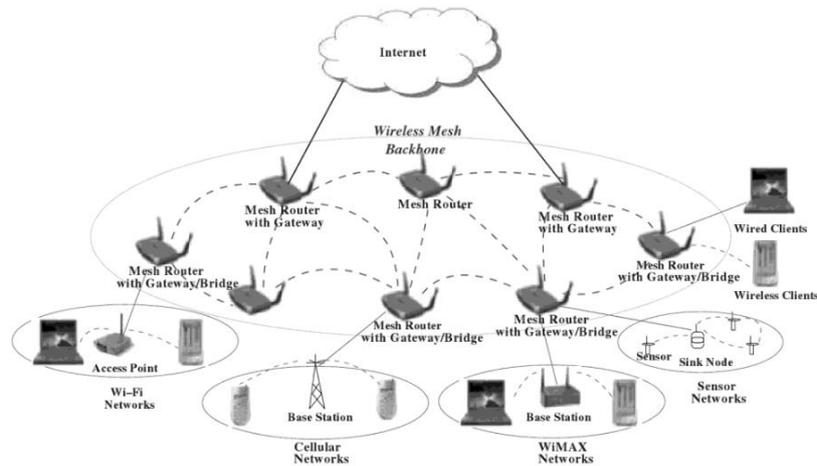
Na década de 1990, com o avanço das tecnologias de comunicação sem fio, o conceito de redes Mesh ganhou mais atenção. Nos anos 2000 o Wi-Fi trouxe novas oportunidades para a aplicação de redes Mesh em ambientes urbanos e domésticos.

A configuração tradicional de redes sem fio utiliza Pontos de Acesso (APs -*Access Points*), que são distribuídos para cobrir uma ampla área com a sobreposição de seus sinais, e cada AP é conectado diretamente à rede de distribuição (*backhaul*) por meio de cabos. Em contraste, as redes em malha sem fio exigem apenas que alguns nós tenham conexão cabeada ao *backhaul*. Os outros nós encaminham pacotes através de múltiplos saltos, criando uma rede mais flexível e escalável.

Segundo Akyildiz e Wang (2009), uma rede Mesh sem fio é composta por nós gateway (MGW - *Mesh Gateway*), roteadores Mesh (MR - *Mesh Router*), pontos de acesso Mesh (MAP - *Mesh Access Point*) e clientes móveis (MC - *Mesh Client*). Como mostrado na figura 4 os nós gateway conectam a WMN a outras redes e à Internet via conexões cabeadas. Os APs Mesh fornecem conectividade aos clientes, geralmente usando o padrão IEEE 802.11, mas outras tecnologias de rede também podem ser usadas para ampliar a aplicabilidade das WMNs. A rede deve suportar clientes não modificados e oferecer conectividade contínua para clientes móveis. Os roteadores Mesh encaminham pacotes entre nós, utilizando padrões como IEEE 802.11,

802.15 e 802.16. Além disso, é crucial manter a separação entre a rede de acesso e o núcleo da WMN através de redes virtuais ou interfaces físicas múltiplas.

Figura 4: Infraestruturas WMNs



(Fonte: AKYILDIZ E WANG, 2009)

Com isso, a principal ideia de uma rede Mesh é que cada nó na rede não só envia seus próprios dados, mas também atua como um repetidor para os dados de outros nós. Criando uma estrutura de rede onde a comunicação pode se redirecionar automaticamente se um caminho estiver indisponível, aumentando a robustez e a flexibilidade da rede.

Vale ressaltar que as redes Mesh são essenciais para a Internet das Coisas, cidades inteligentes e redes de sensores. Devido à sua habilidade, essas redes são fundamentais para ambientes que exigem uma infraestrutura de rede dinâmica e adaptável.

De acordo com uma pesquisa realizada pela empresa de inteligência de mercado, estima-se que redes Mesh tenha um tamanho de mercado de US\$ 9,56 bilhões em 2024, e deve atingir US\$ 14,57 bilhões até 2029, crescendo a um CAGR (*Compound Annual Growth Rate*) de 8,78% durante esse período de. (MORDOR INTELLIGENCE, 2023)

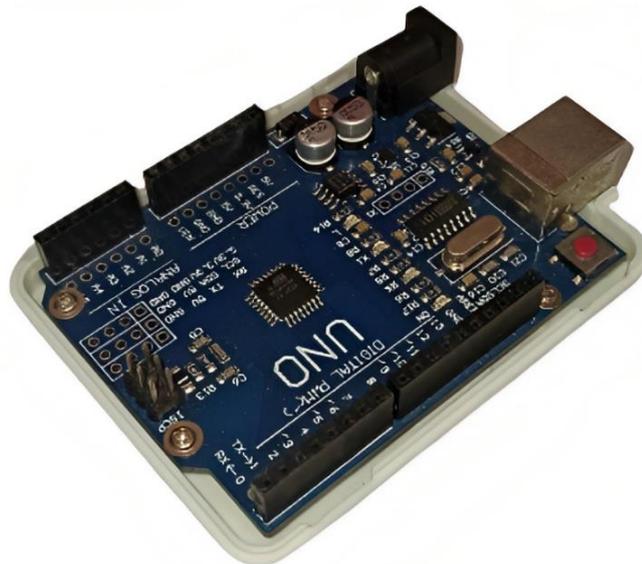
1.5. Arduino

Arduino é um componente voltado para projetos eletrônicos, ou podemos dizer em outro termo, é uma prototipagem eletrônica. A criação desta placa se originou no ano de 2005, por 5 pesquisadores: David CuatIELles, David Mellis, Gianluca Martino, Massimo Banzi e Tom Igoe. Que tiveram como objetivo criar uma ferramenta de baixo custo para projetos educacionais, incentivando a iniciá-los nos fundamentos da robótica (MAKIYAMA,2022).

A plataforma Arduino é *open-source* (código aberto) e pode ser expandida por meio de bibliotecas C/C++. Integra *hardware* e *software* para que as pessoas ou estudantes agreguem os seus conhecimentos de maneira fácil e simples. O ambiente de desenvolvimento ou *Integrated Development Environment* (IDE) que é utilizado para implementação do código e se baseia na linguagem *wiring*.

O Arduino possui um componente microcontrolador sendo capaz de processar os *softwares*, consegue executar programas, analisar e manipular as condições de entrada e saída para que tenha sucesso de comunicação. As placas são capazes de ler variáveis físicas e transformá-las em uma saída. Na figura 5 pode-se analisar os componentes.

Figura 5: Visualização da Placa Arduino UNO



(Fonte: Elaborado pelo autor do projeto)

1.6. Protocolo TCP/IP

Como citado anteriormente, TCP/IP é a sigla de *Transmission Control Protocol/ Internet Protocol*, ou Protocolo de Controle de Transmissão/ Protocolo de Internet, desenvolvido em 1969 pela agência *Defense Advanced Research Projects Agency* (DARPA), por meio de um projeto experimental idealizada em 1967, chamado ARPANET (*Advanced Research Projects Agency Network*), em virtude de criar uma pequena rede para que pesquisadores pudessem compartilhar seus conhecimentos (FOROUZAN, 2010. 2 p.).

No início da década de 1970, algumas redes de dados começaram a ser construída no mundo todo, operando com protocolos de *hardware* e *software* diferentes, dificultando a comunicação entre elas. Em 1972 Vicent Cerf e Bob Kahn diante a estes problemas tiveram a ideia de desenvolver um dispositivo chamado *gateway*, em razão de transferir dados em redes distintas (FOROUZAN, 2010. 2 p.).

Em 1973 Vicent Cerf e Bob Kahn inventaram a rede cruzada TCP, construindo uma rede uma rede internacional de rede de computadores, ou seja, a Internet. O protocolo de controle de comunicação incluía conceito de encapsulamento, datagrama (mensagem enviada sem conexão e sem confirmação) e as funções de um *gateway*. (CERF E KAHN, 1974).

Em 1978 a Agência de Comunicações de Defesa (DCA - *Defense Communications Agency*) e a ARPA (*Advanced Research Projects Agency*), dividiu o TCP em dois protocolos, formando TCP e IP. O protocolo de controle de transmissão ficaria responsável pelo segmento, remontagem e detecção de erros. O protocolo IP cuidaria da parte de roteamento de datagramas. (CERF E KAHN, 1974).

O protocolo TCP é uns dos principais protocolos da camada de transporte no modelo TCP/IP, definido como um protocolo orientado a conexão, utilizando mecanismo de controle de fluxo e de erro, criando uma conexão mais confiável. (TANENBAUM,2011, 28 p.)

1.7. LoRa

Long Range (LoRa), traduzido para o português significa Longo Alcance, refere-se a uma rede *Low-Power Wide-Area Network* (LPWAN), possibilitando uma comunicação de longa distância utilizando uma baixa potência, se tornando uma vantagem para áreas de difícil acesso e com poucas pessoas.

A tecnologia LoRa foi criada pela empresa *Samtech Corporation*, que tiveram como foco desenvolver uma padronização para as conexões de redes de comunicação na indústria. Com o passar dos anos, ganhou reconhecimento pela sua flexibilidade de proporcionar uma comunicação na faixa *Industrial Scientific and medical* (ISM) utilizando um consumo baixo.

Segundo Marques et al (2018). LoRa realiza transmissões de dados via wireless, sem fio, derivada da tecnologia *Chirp Spread Spectrum* (CSS) e por tanto foi projetada para realizar conexões de dispositivos inteligentes em alto alcance diminuindo os níveis de interferência. Podemos definir que a tecnologia LoRa é fundamental para aplicações de IoT.

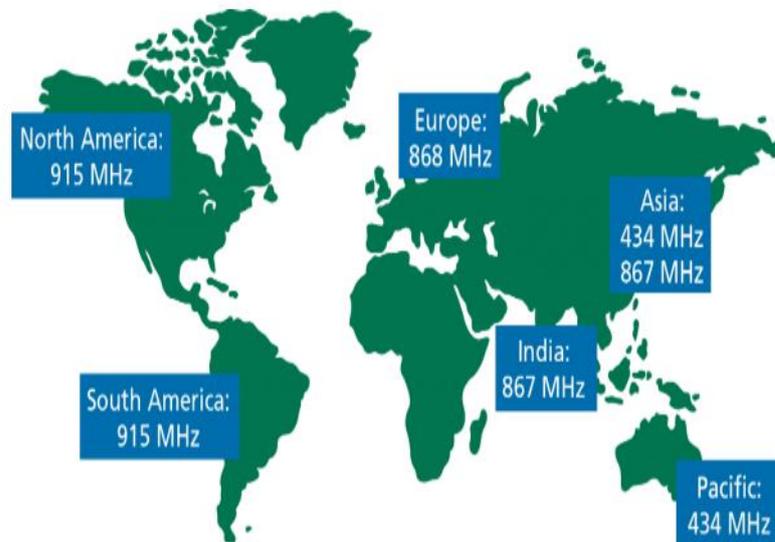
Segundo o mesmo ator, as informações que são transmitidas são codificadas por ondas de rádio usando pulso *chirps*, com isso, o LoRa faz com que os problemas que envolvem a comunicação sem fio sejam solucionados, na qual as transmissões possam ter alto alcance, obter um consumo menor de energia dos sistemas e que seus dados tenha uma transmissão íntegra. Para que isso ocorra de forma eficiente, os dispositivos devem estar localizados em um alcance adequado. A distância do *gateway* pode variar dependendo do dispositivo.

1.7.1. Topologia Básica

De acordo com Agência Nacional de Telecomunicações (ANATEL), no Brasil a frequência regulamentada para ISM fica entre 915 e 928 MHz, pois na camada física (PHY) do LoRa modula sinais em sub-bandas de rádios não licenciadas (ANATEL, 2018).

Na figura 7, podemos observar que tem uma grande variação de frequências dependendo da região.

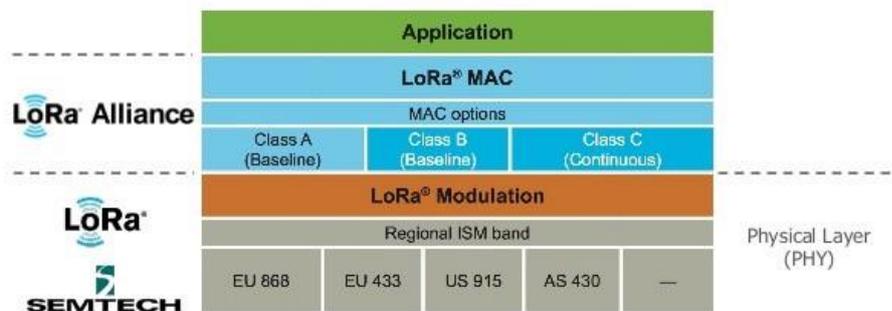
Figura 7: Frequência de operação LoRa



(Fonte: MARQUES e BOCHIE, 2018)

Na figura 8 pode-se observar as informações das transmissões, por meio dessas frequências, os dados enviados para os *gateways* e assim que recebem, os dados são enviados para os servidores utilizando o protocolo TCP/IP.

Figura 8: Camadas do LoRa Informações das Transmissões



(Fonte: MARQUES e BOCHIE, 2018)

1.7.2. Parâmetros de transmissão

A tecnologia LoRa possui alguns parâmetros que personaliza a modulação LoRa, ou seja, essas variáveis podem ser ajustadas, podendo determinar variação de transmissão, alcance do sinal e até mesmo o consumo de energia que está sendo utilizada. Portanto, os parâmetros que estamos nos retratando são: frequência de operação, largura de banda, potência de transmissão, Fator de Espalhamento e taxa de código.

1.7.2.1. Frequência de Operação

As comunicações via tecnologia LoRa devem aderir a padrões de frequência que variam de acordo com a localidade, situando-se globalmente entre 300 e 1000 MHz. Para facilitar a criação de redes amplas e sem grandes entraves burocráticos, essas transmissões frequentemente ocorrem em faixas do espectro que não necessitam de licenciamento. No Brasil, as faixas alocadas para LoRa, identificadas pela LoRa *Alliance* em 2019, são 433 a 435 MHz, 902 a 907,5 MHz, e 915 a 928 MHz, todas dentro do espectro não licenciado. Essas faixas são escolhidas por sua robustez contra atenuações causadas tanto por elementos naturais como chuva e névoa, quanto por barreiras físicas, incluindo prédios e árvores. No entanto, a alta demanda por estas faixas pode resultar em considerável interferência. Para combater esse desafio, a LoRa emprega técnicas de modulação sofisticadas, destinadas a fortalecer a comunicação contra interferências, com mais detalhes fornecidos em uma seção específica do documento.

1.7.2.2. Largura de banda

A largura de banda traduzido do inglês, *Bandwidth* (BW), é o parâmetro que determina a medida de capacidade de dados de transmissão, ou seja, qual o valor de frequência ao domínio, poderá ser utilizado para transmitir uma determinada quantidade de dados. Existe três valores que podem ser assumidos para transmissões LoRa: 125 kHz, 250 kHz e 500 kHz.

1.7.2.3. Potência de Transmissão

A transmissão de dispositivos LoRa geralmente opera com potências bastante reduzidas, principalmente porque muitos desses dispositivos são alimentados por baterias. Os níveis de potência de transmissão para essa tecnologia variam de -4 a 20 dBm (*Decibel-milliwatt*), ajustáveis em incrementos de 1 dBm. Contudo, restrições de *hardware* frequentemente condicionam esse intervalo para entre 2 e 20 dBm. Adicionalmente, existem regulações específicas em diferentes países que limitam a potência máxima de transmissão em determinadas faixas de frequência. Na União Europeia, a transmissão na faixa de 863 a 870 MHz (*Megahertz*) para comunicações *uplink* é limitada a 25 mW (ou 14 dBm). As especificações sobre o ganho das antenas, que são cruciais em aplicações de baixa potência,

também enfrentam restrições; nesse contexto, o ganho máximo de uma antena não pode ultrapassar +2,15 dBi (*Decibel-isotropic*).

1.7.2.4. Fator de Espalhamento

Fator de espalhamento, tendo referências de artigos, *Spreading Factor* (SF) em inglês, é um dos parâmetros importantes para modulação LoRa, pois tem a responsabilidade de melhorar a transmissão de comunicação entre os dispositivos. Este fator faz com que a mensagem seja dividida em vários *chirps*, maximizando a sensibilidade do receptor, isto pode variar para cada dispositivo diante a sua finalidade.

O *spreading factor* refere-se à proporção entre a taxa de chips da sequência empregada para codificar os dados e a taxa de símbolos transmitidos. Na tecnologia LoRa, esse fator pode variar e é especificado nos valores discretos de 6, 7, 8, 9, 10, 11 ou 12, conforme definido pelas normas estabelecidas para essa tecnologia de comunicação sem fio de longo alcance.

1.7.2.5. Taxa de Código

A Taxa de Código, ou *Code Rate* (CR) em inglês, é definida pelo número de bits destinados à repetição da mensagem. Em outras palavras, representa a proporção do fluxo de bits transmitidos que são úteis em relação aos redundantes. A sua representação é através de um número fracionário k/n .

K = bits de informação

N =bits de informação que são gerados

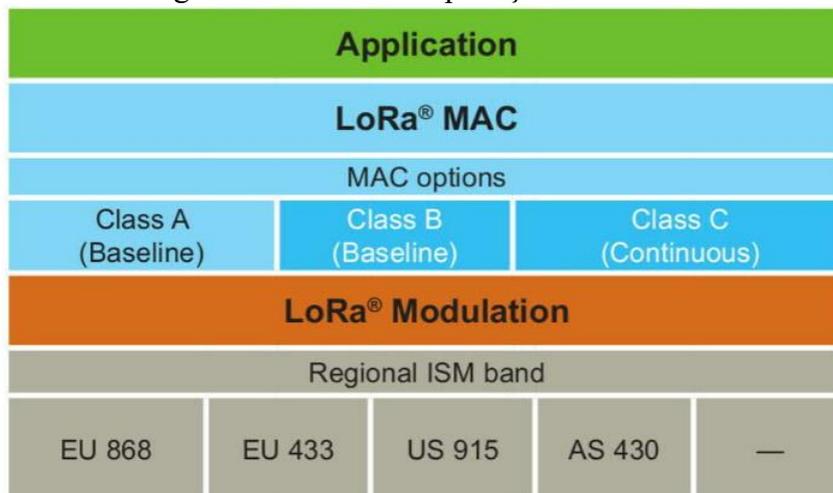
Quanto maior o número de CR, mais segurança teremos em não haver perdas e maior o tempo de transmissão. O número do CR pode variar de 1 a 4 (DELLALIBERA, 2018). Os valores mais utilizados para a CR de em comunicação LoRa são 4/5, 4/6, 4/7 e 4/8, com esses valores, o sinal fica mais robusto, trazendo uma proteção maior contra as interferências e uma grande possibilidade de correção de bit com redundância (BUENO, 2021).

1.7.3. LoRaWAN

LoRaWAN (*Long Range Wide-Area Network*) é um protocolo de controle de acesso à mídia ou em seu termo inglês *Media Access Control* (MAC) situado sobre a modulação LoRa. Desenvolvido e tratado pela *LoRa Alliance*, tendo seu primeiro lançamento em 2015. Trata-se de uma camada de *software* que estabelece os procedimentos pelos quais os dispositivos aproveitam as capacidades do *hardware* LoRa, incluindo o momento das transmissões e a estrutura das mensagens. No entanto os dispositivos configurados têm a comunicação direta com a internet.

Segundo Saari (2018), os dispositivos LoRaWAN são mais que módulos radiofrequência, pois possuem microcontroladores que são dedicados a outros componentes, em virtude de conectar os dispositivos à rede e favorecer uma conexão segura. Na figura 9, pode-se analisar a camada de aplicação do LoRaWAN.

Figura 9: Camada de Aplicação LoRaWAN



(Fonte: SAARI,2018)

1.7.3.1. Classes

Na figura 9 é identificado três classes, classe A classe B e classe C. Estas classes são itens que fazem parte da estrutura da camada MAC LoRaWAN, e são responsáveis por definirem o tipo de modos de operação dos dispositivos desta tecnologia.

- Classe A

A classe A é sigla que representa “All”, ou seja, todos ou tudo. Todas as outras classes deste protocolo, B e C, são capazes de responder a essa classe. A classe A pode ser assumido como o modo mais simples de transmissão, podendo enviar dados a qualquer momento, com isso ficam no aguardo por qualquer resposta que venha do servidor, apresentando um baixo consumo de energia.

A classe A tem capacidade de criar duas janelas de recepção de resposta do servidor. Quando o transmissor enviar um pacote (*uplink*), o receptor (Classe A) tem duas formas de analisar e responder se os pacotes chegaram (*downlink*) de forma íntegra. A primeira forma de identificação de *downlink* é aberto na primeira janela, representada como Rx1 na figura 10. Conforme a padronização de tempo de espera de *downlink*, que podendo ser alterado nas configurações, estimasse uma espera de 1 segundo.

Na figura 10, é mostrado a transmissão do pacote, o tempo de resposta e a identificação do recebimento do pacote de forma íntegra na primeira janela de recepção.

Figura 10: Recebimento do pacote na primeira janela Classe A



(Fonte: LORA DEVELOPS SEMTECH)

Caso não tenha identificado nenhum *downlink* na primeira janela, se inicia a segunda forma de identificação, abrindo a segunda janela de recepção. Neste caso, observaremos um tempo de resposta maior, totalizando 2 segundos, conforme apresentado na figura 11. E ele pode também receber esta mensagem na segunda janela, pode ocorrer de não receber nenhuma resposta.

Na figura 11, é realizado o início da transmissão, o tempo de resposta de 2 segundos e o recebimento do pacote na segunda janela de recepção de pacotes.

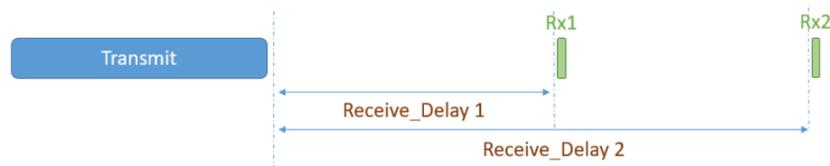
Figura 11: Recebimento do pacote na segunda janela Classe A



(Fonte: LORA DEVELOPS SEMTECH)

Uma terceira identificação que é considerado também que, quando não receber nenhum pacote em ambas as janelas, assim como mostrado na figura 12.

Figura 12: Nenhum pacote recebido Classe A



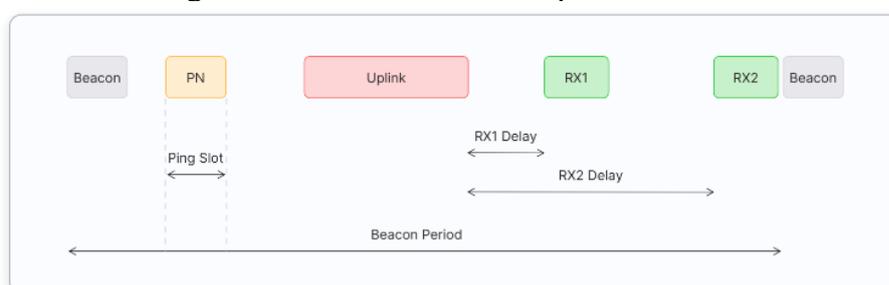
(Fonte: LORA DEVELOPS SEMTECH)

- Classe B

A classe B é a sigla que representa a palavra “*Beacon*” ou “*Beaconing*”, são definidos como dispositivos finais e fornecem janelas com *slots* de *ping* de recebimento agendados regularmente, ou seja, além das janelas de recebimento da Classe A há também as janelas fornecidas da classe B. Para que esse processo que acabamos de citar funcione adequadamente, um *beacon* deve ser configurado em um tempo sincronizado, e assim configurado, os *beacons* passam pela rede por meio dos *gateways* e em tempos em tempos são sincronizando com o relógio interno com a rede. Entretanto, permite que o servidor entenda se o dispositivo final está “ouvindo”.

Na figura 13 é realizado uma transmissão de um pacote de um servidor para um dispositivo final (Classe B).

Figura 13: Transmissão de um pacote classe B



(Fonte: THE THING NETWORK)

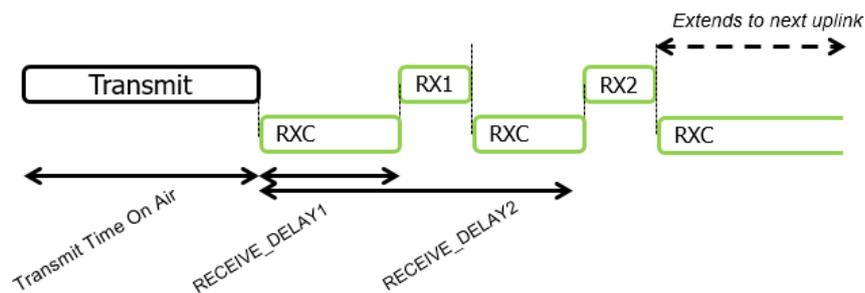
- Classe C

A Classe C é a sigla que representa a palavra “*Continuous*”, são definidos também como dispositivos finais, esta classe faz com que o dispositivo fique à disposição de uma resposta do servidor, mantendo as janelas de recepção quase sempre abertas, ou seja, ele fica ouvindo até

receber alguma mensagem. Quando o dispositivo realiza alguma transmissão, estas portas são fechadas. Vale ressaltar que por conta dessa operação, os dispositivos finais Classe C, têm um consumo maior de energia e oferecem uma menor latência para comunicação do servidor, referente as outras classes que demonstramos acima.

De início, o dispositivo Classe C funciona com as mesmas janelas da Classe A, Rx1 e Rx2, como podemos observar na figura 14 (Recebimento do pacote na segunda janela). Porém, a segunda janela (Rx2) não é fechada até enviarem a próxima transmissão de *uplink* de volta ao servidor, pois nela pode receber um *downlink* a qualquer momento. Segundo a própria documentação “uma pequena janela na frequência e taxa de dados do RX2 também é aberta entre o final da transmissão e o início da janela de recepção do RX1.”

Figura 14: Janela de recebimento Classe C

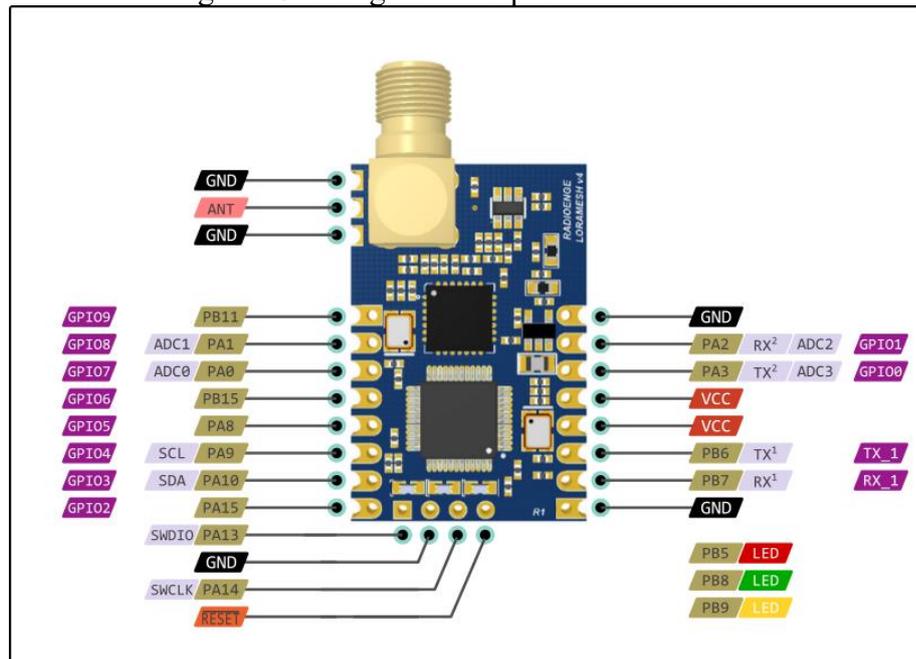


(Fonte: LORA DEVELOPS SEMTECH)

1.7.3.2. Pinout

Na figura 15, são determinados os *pinouts* numerados de 1 a 19 do LoRaMESH, a interface física que permite conectar dispositivos ao módulo LoRaMESH para comunicação sem fio de longo alcance em redes Mesh.

Figura 15: Pinagem do dispositivo LoRaWAN



(Fonte: CASSETARI,2019)

Para compreensão melhor dos *pinouts* do LoRaMESH, a tabela 1 apresenta o número dos pinos, o nome cada pino, o tipo de porta e uma breve descrição sobre ele.

Quadro 1: Lista de definição dos pinos LoRaWAN

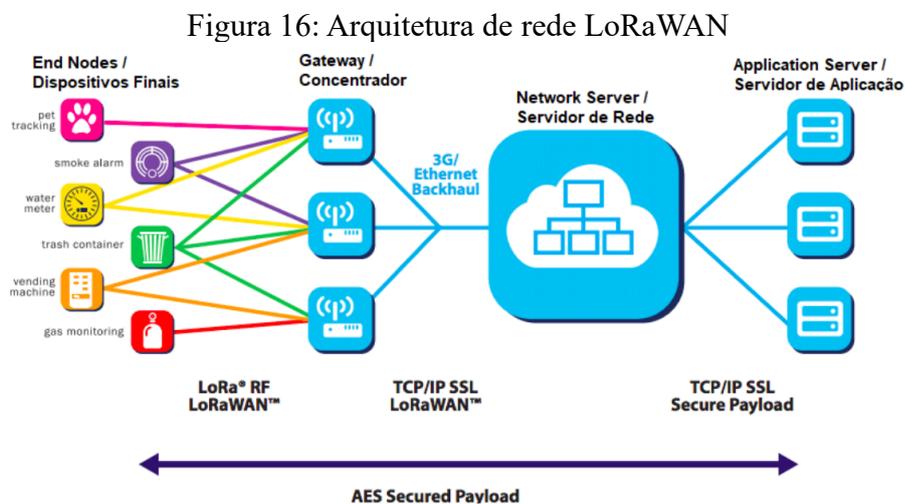
Pino	Nome	Tipo	Descrição
1	GND	Alimentação	Conectado ao <i>ground</i>
2	RX_1	Entrada	RX da interface UART de comando
3	TX_1	Saída	TX da interface UART de comando
4	VCC	Alimentação	Conectado à alimentação
5	VCC	Alimentação	Conectado à alimentação
6	GPIO0/AD	I/O ou Analógico	Pino de uso geral GPIO0 ou entrada analógica
7	GPIO1/AD	I/O ou Analógico	Pino de uso geral GPIO1 ou entrada analógica
8	GND	Alimentação	Conectado ao <i>ground</i>
9	GPIO2	I/O	Pino de uso geral
10	GPIO3	I/O	Pino de uso geral
11	GPIO4	I/O	Pino de uso geral
12	GPIO5	I/O	Pino de uso geral
13	GPIO6	I/O	Pino de uso geral
14	GPIO7/AD	I/O ou Analógico	Pino de uso geral GPIO7 ou entrada analógica
15	GPIO8/AD	I/O ou Analógico	Pino de uso geral GPIO8 ou entrada analógica
16	GPIO9	I/O	Pino de uso geral
17	GND	Alimentação	Conectado ao <i>ground</i>

18	ANT	Saída RF	Saída de RF para antena externa
19	GND	Alimentação	Conectado ao <i>ground</i>

(Fonte: ADAPTADO MANUAL LORAWAN, 2023)

1.7.3.3. Arquitetura de rede LoRaWAN

Os Dispositivos Finais são pequenos dispositivos com sensores, como exemplo sensor de Umidade, além de interruptores que tem a função de ligar e desliga. Eles usam a tecnologia LoRa para se comunicar com os *gateways*, que recebem os dados brutos dos sensores ou de algum outro dispositivo final, e os enviam para um servidor de rede. Cada *gateway* pode cobrir uma área considerável, recebendo dados de muitos sensores, dependendo da localização. Os Servidores de Rede gerenciam as informações dos *gateways*, e os servidores de aplicações executam ações específicas com base nesses dados. É importante destacar que a estrutura do LoRaWAN divide claramente em duas camadas: a camada física (PHY), baseada na modulação LoRa, e a camada, definida pelo protocolo LoRaWAN (CUNHA et al, 2020). A figura 16 representa a arquitetura de uma rede LoRaWAN.



(Fonte: CUNHA et al, 2020)

1.7.3.4. Comando AT

Os comandos AT são responsáveis por nos enviar e receber informações da placa LoRa. Todos os comandos serão iniciados com as duas letras ‘AT’, e o único diferencial para buscar uma determinada informação, está representada na frente do comando, na tabela 2 podemos visualizar melhor. Estes comandos não diferem utilizando o dispositivo LoRaMESH.

Quadro 2: Lista de Comando AT

Comando	Descrição
+DEUI	<i>Get</i> o DevEui
+DADDR	<i>Get/Set</i> o DevAddr
+APPKEY	<i>Get/Set</i> o AppKey
+APPSKEY	<i>Get/Set</i> o AppSKey
+NWKSKEY	<i>Get/Set</i> o NwkSKey
+APPEU	<i>Get/Set</i> o AppEui/JoinEui
+DR	<i>Get/Set</i> o <i>Datarate</i>
+RX2FQ	<i>Get/Set</i> a janela de frequência Rx2 em Hz
+RX2DR	<i>Get/Set</i> o <i>datarate</i> da janela Rx2
+RX1DL	<i>Get/Set</i> o <i>delay</i> da janela Rx1 em milissegundos
+RX2DL	<i>Get/Set</i> o <i>delay</i> da janela Rx2 em milissegundos
+JN1DL	<i>Get/Set</i> o <i>delay</i> do <i>Join</i> janela 1 em milissegundos
+JN2DL	<i>Get/Set</i> o <i>delay</i> do <i>Join</i> janela 2 em milissegundos
+NJM	<i>Get/Set</i> o modo <i>Join</i>
+CLASS	<i>Get/Set</i> a classe do dispositivo
+JOIN	Executa o procedimento <i>join</i>
+NJS	Informa o status do <i>join</i>
+SENDER	Envia dados hexadecimais junto com a porta do aplicativo
+SEND	Envia dados de texto junto com a porta do aplicativo
+VER	Informa a versão do <i>firmware</i>
+CFM	<i>Get/Set</i> o modo de confirmação
+SNR	Informa a SNR do último pacote recebido em dB
+RSSI	Informa o RSSI do último pacote recebido em dBm
+BAT	Informa o nível da bateria
+BAUDRATE	<i>Get/Set</i> o <i>baudrate</i> da interface UART
+NBTRIALS	<i>Get/Set</i> o número de retransmissões
+KEEPALIVE	<i>Get/Set</i> os pacotes de <i>keepalive</i> do dispositivo
+TXCFM	Envia dados de texto junto com a porta do aplicativo, indicando se é confirmado ou não

+TXBCFM	Envia dados hexadecimais junto com a porta do aplicativo, indicando se é confirmado ou não.
+CHMASK	<i>Get/Set</i> a máscara de canal
+GPIOC	Configura os pinos I/O
+WPIN	Realiza a escrita dos pinos
+RPIN	Realiza a leitura dos pinos

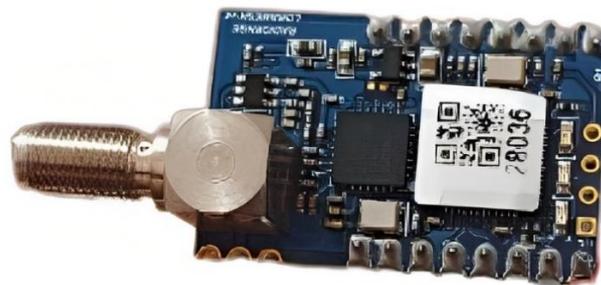
(Fonte: MANUAL LORAWAN – *Radioenge* 2024)

Conforme demonstrado acima, podemos observar a quantidade de comando que podemos utilizar para interagir com a placa LoRa.

1.7.4. LoRaMESH

O sistema LoRaMESH é caracterizado pela sua capacidade de estabelecer uma rede de comunicação Mesh, conforme informa no seu nome. Tal rede é composta por dispositivos que desempenham funções tanto de usuários finais, responsáveis pela emissão e recepção de dados, quanto de roteadores, encarregados de encaminhar pacotes de dados provenientes de outros dispositivos na rede. O dispositivo LoRaMESH da *Radioenge* possui a tecnologia LoRa® na camada física. Na figura 17 pode-se observar dispositivo real.

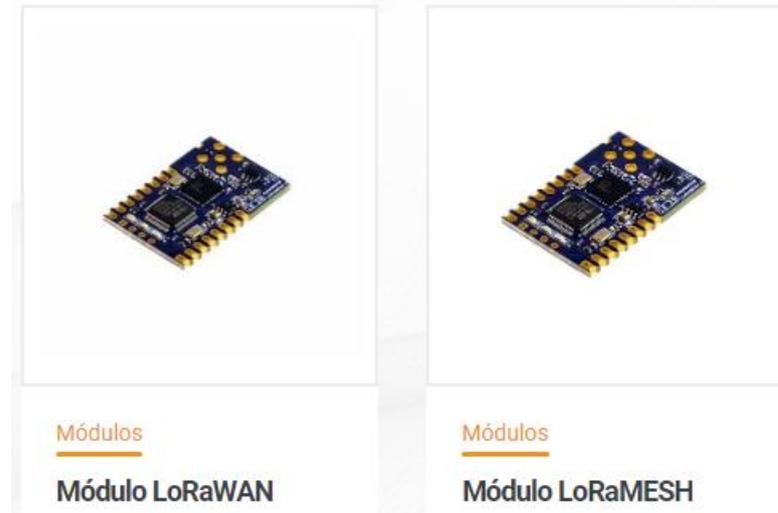
Figura 17: Visualização do dispositivo LoRaMESH Radioenge - RD24C



(Fonte: Elaborado pelo autor do projeto)

Ao analisar os dispositivos LoRa no site da *Radioenge*, deve-se observar se o dispositivo é LoRaMESH ou LoRaWAN, a similaridade física entre eles é perceptível como mostrado na figura 18. O diferencial entre ambos os dispositivos é o seu *firmware*.

Figura 18: Visualização dos dispositivos LoRa Radioenge - RD29C e RD42C

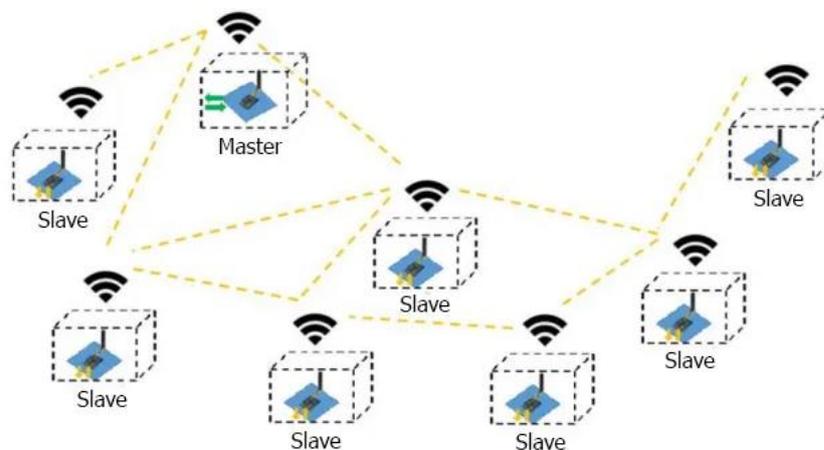


(Fonte: Elaborado pelo autor do projeto)

1.7.4.1. Gerenciamento do LoRaMESH

Em um gerenciamento de rede LoRaMESH *Radioenge*, pode-se observar dois tipos de dispositivos, o *master* e o *slave* como apresentado na figura 19. O dispositivo *master* é único em uma rede, ele pode enviar comandos ou pacotes com informação para qualquer *slave* em uma rede, porém o *slave* tem duas funções, realizar o roteamento de pacotes e o envio informações apenas para o *master*.

Figura 19: Diagrama de rede LoRaMESH



(Fonte: ADAPTADO MANUAL LORAMESH, 2024)

1.7.4.2. Tensão dos dispositivos

Na tabela 3 pode-se visualizar a variedade de tensão suportadas pelo dispositivo LoRaMESH de acordo com seus respectivos objetivos.

Quadro 3: Valores Máximos e Absolutos

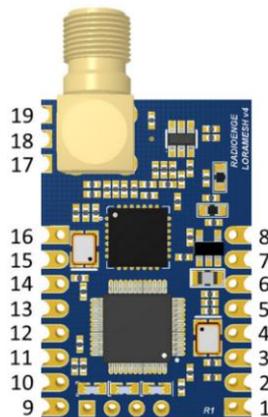
Parâmetro	Mínimo	Máximo	Un.
Tensão entre VCC e GND (com regulador)	-0,3	16,0	V
Tensão entre VCC e GND (sem regulador)	-0,3	4,0	V
Tensão nos pinos de UART (classe A)	-0,3	3,6	V
Tensão nos pinos de UART (classe C)	-0,3	5,0	V
Tensão nos pinos GPIO e Analógicos	-0,3	4,0	V
Corrente máxima drenada por uma GPIO	-	16,0	mA
Corrente máxima fornecida por uma GPIO	-	16,0	mA
Corrente máxima drenada por todas as GPIOs	-	90	mA
Temperatura de armazenamento	-10	+70	°C
Temperatura de operação	-5	+55	°C
Potência máxima na entrada RF	-	0	dBm

(Fonte: MANUAL LORAMESH, 2024)

1.7.4.3. Pinout da placa RD42C

Na figura 20 podemos observar a arquitetura no LoRaMESH e onde está localizado alguns pinos que precisaremos no decorrer do projeto.

Figura 20: Pinout numerada do LoRaMESH



(Fonte: MANUAL LORAMESH, 2024)

Na tabela 4 define com mais detalhes sobre os *pinout*, com isso podemos analisar a figura 20 e ter uma definição de qual é a responsabilidade de cada pino. O LoRaMESH possui 8 portas GPIO (*General Purpose Input Output*) para entrada e saída de dados digitais, duas portas GPIO são portas analógicas, possibilitando leitura de sensores e controle de atuadores.

Além de ter duas portas UART (*Universal Asynchronous Receiver-Transmitter*), uma definida para os comandos e a outra definida para a comunicação transparente.

Quadro 4: Definição dos pinos LoRaMESH

Pino	Nome	Tipo	Descrição
1	GND	Alimentação	Conectado ao ground
2	RX_1	Entrada	RX da interface UART de comando
3	TX_1	Saída	TX da interface UART de comando
4	VCC	Alimentação	Conectado à alimentação
5	VCC	Alimentação	Conectado à alimentação
6	TX_2	Saída	TX da interface UART transparente
7	RX_2	Entrada	RX da interface UART transparente
8	GND	Alimentação	Conectado ao ground
9	GPIO0	I/O	Pino de uso geral
10	GPIO1	I/O	Pino de uso geral
11	GPIO2	I/O	Pino de uso geral
12	GPIO3	I/O	Pino de uso geral
13	GPIO4	I/O	Pino de uso geral
14	GPIO5	I/O ou Analógico	Pino de uso geral ou entrada analógica
15	GPIO6	I/O ou Analógico	Pino de uso geral ou entrada analógica
16	GPIO7	I/O	Pino de uso geral
17	GND	Saída RF	Conectado ao ground
18	ANT	Saída RF	Saída de RF para antena externa
19	GND	Saída RF	Conectado ao ground

(Fonte: MANUAL LORAMESH, 2024)

1.7.4.4. Taxa de dados

Na tabela 5 é possível visualizar a taxa de dados que varia de acordo com alguns principais parâmetros que citamos na seção 1.7.2, são eles: *Bandwidth*, *Spreading Factor* e *Coding Rate*.

Quadro 5: Taxa de dados LoRaMESH

CODING RATE 4/5							
Taxa De Dados (Bps)		Spreading Factor					
		7	8	9	10	11	12
Largura De Banda (kHz)	125	5469	3125	1758	977	537	293
	250	10938	6250	3516	1953	1074	586
	500	21875	12500	7031	3906	2148	1172
CODING RATE 4/6							
Taxa De Dados (Bps)		Spreading Factor					
		7	8	9	10	11	12
Largura De	125	4557	2604	1465	814	448	244
	250	9115	5208	2930	1628	895	488

Banda (kHz)	500	18229	10417	5859	3255	1790	977
CODING RATE 4/7							
Taxa De Dados (Bps)		Spreading Factor					
		7	8	9	10	11	12
Largura De Banda (kHz)	125	3906	2232	1256	698	384	209
	250	7813	4464	2511	1395	767	419
	500	15625	8829	5022	2790	1535	837
CODING RATE 4/8							
Taxa De Dados (Bps)		Spreading Factor					
		7	8	9	10	11	12
Largura De Banda (kHz)	125	3418	1953	1098	610	336	183
	250	6836	3906	2197	1220	671	366
	500	13672	7813	4395	2441	1343	732

(Fonte: ADAPTADO MANUAL LORAMESH, 2024)

1.7.4.5. Modo de Operação

Existe dois tipos de configurações que pode ser implementado no dispositivo finais, no entanto precisa definir qual objetivo que cada dispositivo final terá no projeto. Classe A e Classe C são os dois tipos que pode ser aplicado nas configurações. Às definições mais detalhadas estão na sessão 1.7.3.1. A figura 21 representa a diferença entre ambos.

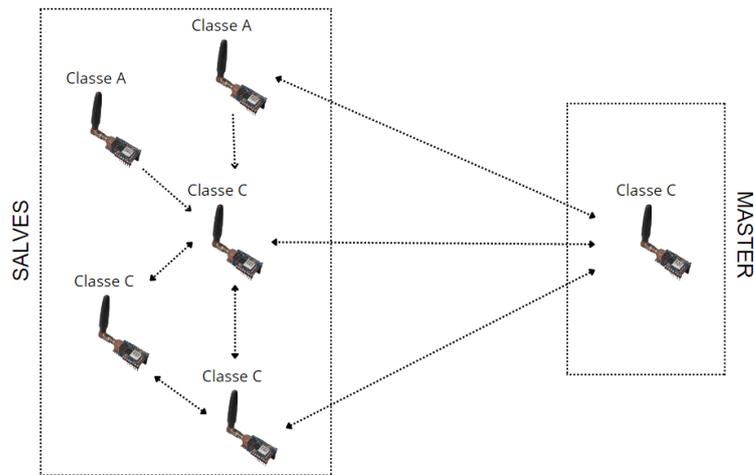
- Classe A

Na classe A não é possível fazer roteamento de pacotes dos outros *EndDevices*, portanto tende a ser uma configuração destinada a quem quer um baixo consumo de energia e queria somente enviar os dados coletados para o *master*. A classe

- Classe C

Na classe C é totalmente o oposto que podemos ter na classe A, pois o dispositivo sempre ficar ativo e realizando o roteamento dos dados dos *slaves*. A classe C tem um consumo um pouco maior que o da classe A, porque está sempre ativo.

Figura 21: Representação Classe A e Classe C



(Fonte: Elaborado pelo autor do projeto)

1.7.4.6. Protocolo Modbus

O protocolo Modbus desenvolvida em 1979 pela *Modicon Industrial Automation Systems*, denominada como *Schneider*. Este protocolo é uma estrutura de mensagem composta por *bytes*, em virtude de estabelecer comunicação entre diversos dispositivos, sendo eles mestre e escravos. (ALFA INSTRUMENTO, 2020).

A comunicação mestre e escravo permite que somente um dispositivo, sendo ele o mestre, possa iniciar as transações, enquanto os dispositivos escravos respondem de acordo com o que foi solicitado pelo mestre. Os dispositivos escravos são definidos por processar as informações e enviá-las para o dispositivo mestre (ALFA INSTRUMENTO, 2020).

1.8. Trabalhos Relacionado

Segundo Douglas em seu projeto de pós-graduação “Implementação e análise de protocolos de roteamento para redes Mesh sem fio LoRa”, é implementado e analisado diferentes protocolos de roteamento para Cidades Inteligentes, desenvolvendo uma ferramenta no simulador *Cupcarbon* para calcular o alcance dos nós e propondo um algoritmo de ajuste de potência de transmissão para economizar energia. Usando a tecnologia LoRa, o estudo simula uma rede em João Pessoa-PB, avaliando critérios como entrega de pacotes, atraso, jitter e taxa de transferência. Os resultados mostram que o protocolo DSR (*Dynamic Source Routing*), ajustado para economia de energia, foi o mais eficiente, economizando 11,32% de energia. O protocolo AODV (*Ad Hoc On-Demand Distance Vector*) teve o melhor desempenho geral, enquanto o DVR (*distance-vector routing*) consumiu mais energia e teve alta latência, mas perdeu mais pacotes. Testes experimentais foram realizados com o DSR e a aplicação no servidor The Things Network (TTN).

Segundo Édson em seu trabalho de graduação “Implementação de Rede Mesh Utilizando Comunicação de Longo Alcance para Sensoriamento Remoto de Produção Agrícola.”, desenvolveu um dispositivo eletrônico utilizando o microcontrolador ESP32 e tecnologia LoRa para criar uma Rede Mesh dinâmica, confiável e de baixo consumo de energia, adequada para ambientes com pouca infraestrutura. A rede foi testada com cinco nós em diferentes cenários, incluindo uma área industrial, e mostrou-se eficaz na comunicação com até dois saltos. Foram analisados a formação de tabelas de vizinhos, rotas, saltos de comunicação e leituras de sensores, verificando-se a eficácia da rede através do cálculo de perdas de pacotes. O resultado mostrou que a Rede Mesh conseguiu estabelecer uma comunicação confiável com uma taxa de perda de pacotes mínima.

CAPÍTULO II

Metodologia

A metodologia teve como base a característica da pesquisa, os materiais para implementação do projeto e quais serão os procedimentos para a coleta de dados

2.1. Caracterização Da Pesquisa

Este trabalho envolverá uma pesquisa detalhada do tipo descritiva e exploratória, com o objetivo de identificar e analisar variáveis que possam impactar o objeto de estudo. Além disso, serão propostas estratégias para controlar essas variáveis, a fim de examinar os efeitos resultantes de sua influência.

2.2. Materiais para implantação do projeto

- **Arduino:** a função deste componente se dá a implementação da primeira fase, possibilitando meios de configuração via programação, dando estabilidade ao rádio LoRaMESH e aos sensores que serão adaptados ao projeto e assim possibilitando também programar e analisar as informações que estão sendo processadas.
- **Radioenge LoRaMESH RD42C:** LoRaMESH terá como função estar implementando tanto na primeira quanto na segunda fase, fazendo com que ambas possam se comunicar.
- **Conversor FTDI232:** Este dispositivo tem como principal função de converter USB (*Universal Serial Bus*) para serial, ou seja, permite que a nossa placa RD42C se comunique com o computador.
- **Cabos Jumpers:** são cabos que utilizaremos para interligar nossos equipamentos, como exemplo, interligar os sensores com o Arduino.

- **Sensor de Umidade do solo:** este é o sensor principal do nosso projeto, tem como função coletar as informações sobre a umidade do solo.
- **Led:** O led terá como função de sinalizar o resultado, ou seja, se o solo está seco ou úmido.
- **Potenciômetro:** tem como objetivo para realizarmos teste de portas analógicas, e assim analisar os valores coletados através dele
- **Configurador LoRaMESH – v2r7p0:** é um *software* que possibilita realizar as configurações do dispositivo LoRaMESH, possibilitando, teste de envio de transmissão, controle remoto, visualização dos dados das portas analógicas, configurações das portas de entradas digitais (GPIO), visualização de ida e volta de uma transmissão, atualização de *firmware* entre outros.

Na figura 22 pode-se visualizar melhor os dispositivos físicos que serão utilizados para implementação do projeto.

Figura 22: Dispositivos físicos do projeto



(Fonte: Elaborado pelo autor do projeto)

2.3. Procedimentos para coleta e análise de dados

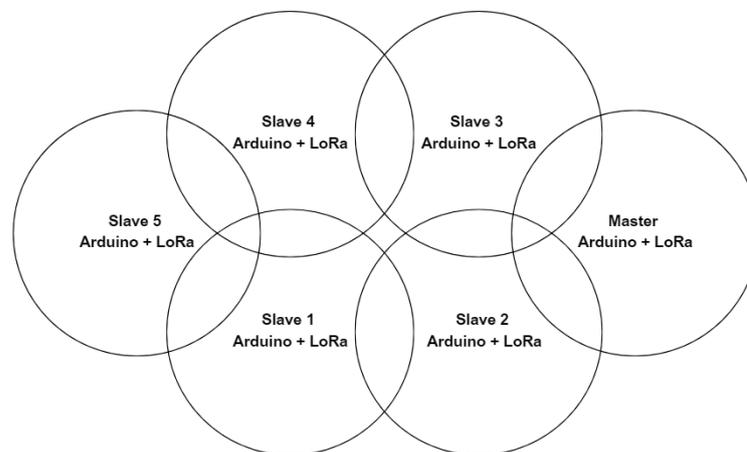
Nos tópicos seguintes será apresentado todo o processo que foi realizado diante o desenvolvimento do projeto, explicando detalhadamente cada passo para melhor compreensão.

2.3.1. Técnicas para coleta de dados

Na coleta de dados foi realizado duas fases. A primeira fase será a exploração do *software* Configurador LoRaMESH – v2r7p0, na qual teve a capacidade de aplicar configurações para o dispositivo RD42C. Na primeira fase foi utilizado um Arduino Uno, um conversor FTDI, cabos jumpers, um dispositivo RD42C e um potenciômetro para simulação das informações geradas pelas portas GPIO, em específico a porta analógica GPIO 5 do dispositivo LoRaMESH. Na segunda fase, utilizou-se a IDE do Arduino para programar as configurações do LoRaMESH, bem como desenvolver uma lógica para a coleta de dados da porta analógica do dispositivo *slave*. Nesta mesma fase os dispositivos utilizados foram: um LoRaMESH, um Arduino, cabos jumpers e um led para sinalização, descrito na sessão 2.3.3.6.

O fluxo de transmissão de dados ocorre dos dispositivos *slaves* para o dispositivo *master*. Independentemente da localização do *slave*, é possível transmitir seus dados através de outros dispositivos próximos. Na figura 23 é apresentado uma possibilidade de implementação de comunicação entre os dispositivos mestre e escravos.

Figura 23: Demonstração de comunicação entre os dispositivos



(Fonte: Elaborado pelo autor do projeto)

Os dados coletados podem ser provenientes de uma variedade de sensores, os quais os agricultores podem optar por implementar, dependendo de seus objetivos específicos.

2.3.2. Natureza da análise de dados

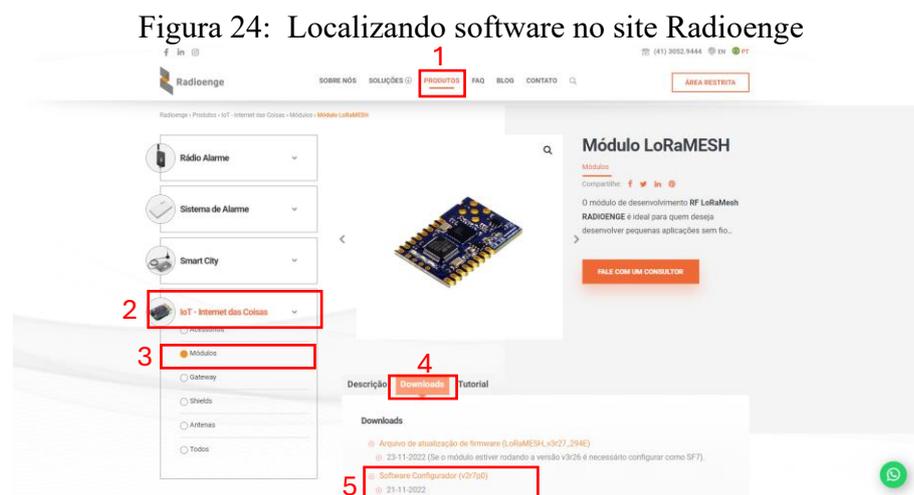
Conforme os dados foram coletados e transmitidos ao dispositivo *master* de forma eficiente e íntegro, possibilitou a compreensão do campo, observando os principais pontos que podem ser melhorados e assim definir estratégias de negócio para empresa, direcionando seus esforços para o crescimento e desenvolvimento do agronegócio.

2.3.3. Testes e implementação do código no projeto

Algumas figuras serão representadas com os processos que foram realizados, para ter uma melhor visualização e compreensão deste projeto.

2.3.3.1. Baixando *software* Configurador LoRaMESH – v2r7p0

O *software* é disponibilizado pela própria empresa do dispositivo *Radioenge*, podendo ser encontrado no site oficial. Após localizar o dispositivo modulo LoRaMESH, na aba ‘Downloads’, localize por ‘*Software* Configurador (v2r7p0)’ para baixá-lo no computador, como mostra na figura 24.

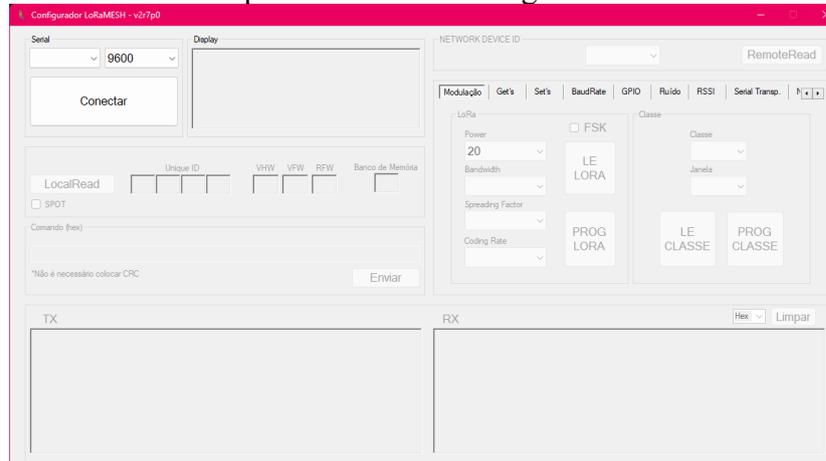


(Fonte: Elaborado pelo autor do projeto)

Depois que baixar, basta abrir o explorador de arquivos, ir à pasta “downloads” e procurar o arquivo baixando. Por fim, extrai-lo para que o configurador possa ser utilizado.

Ao inicializar o *software*, não conseguimos ter permissão de explorar as funções sem um dispositivo LoRaMESH conectado, conforme mostrado na figura 25. Portanto é necessário que um dispositivo RD42C seja conectado para observar os recursos disponíveis.

Figura 25: Primeiras impressões com o Configurador LoRaMESH – v2r7p0



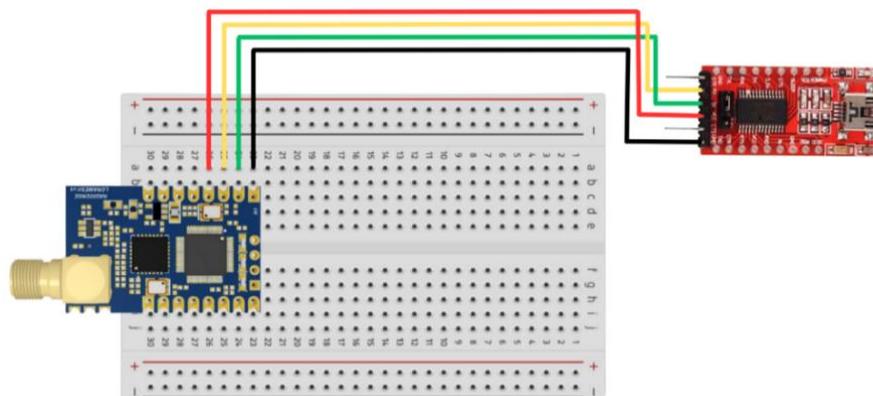
(Fonte: Elaborado pelo autor do projeto)

2.3.3.2. Montagens FTDI232 e RD42C

Para possibilitar a comunicação com o *software* de configuração do LoRaMESH, foi necessário estabelecer uma conexão entre a placa RD42C e o computador. Para isso, deve-se conectar a placa ao conversor de tensão, utilizando um regulador de tensão ajustado para 5V, conforme ilustrado na figura 26.

As entradas RX (Receber) e TX (Transmitir) devem estar alternadas. O RX da placa RD42C deve estar na entrada TX do conversor, e o TX no RX.

Figura 26: Montagem circuitos RD42C + FTDI232



(Fonte: Elaborado pelo autor do projeto)

2.3.3.3. Configuração o dispositivo *Master*

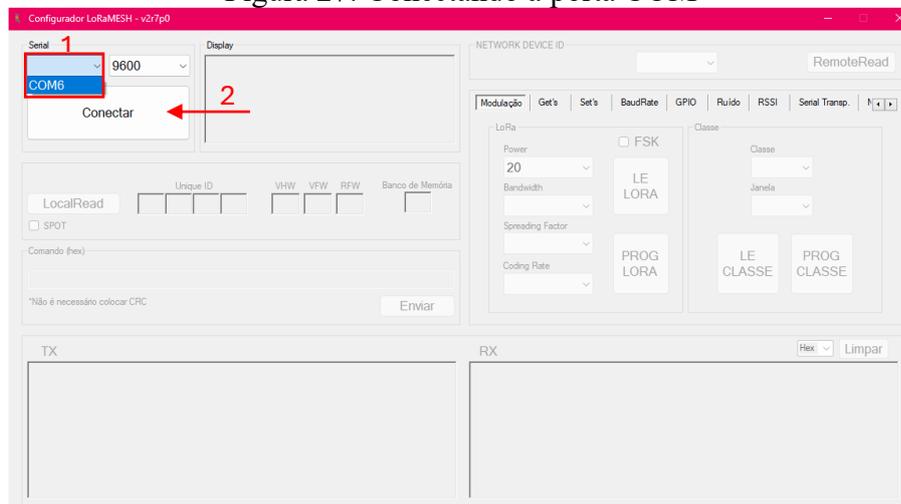
Nesta fase, foram realizados os ajustes específicos do RD42C, estabelecendo-o como o dispositivo principal, o '*MASTER*'. Para isso, utilizou-se o *software* 'Configurador LoRaMESH – v2r7p0', essencial para a configuração detalhada do dispositivo, facilitando a programação do RD42C e implementando as configurações desejadas. É importante destacar novamente que, para fazer essas configurações, foi utilizado um módulo conversor FTDI232, conforme mostrado na figura 26.

- Selecionar a Serial COM

Recomenda-se utilizar apenas um dispositivo por vez ao selecionar a porta serial. Caso contrário, o sistema pode gerar respostas inesperadas e apresentar alguns bugs."

Na figura 27 pode-se observar que só existe uma porta COM que o computador identificou, caso tenha mais certifique-se qual se refere a placa RD42C.

Figura 27: Conectando a porta COM



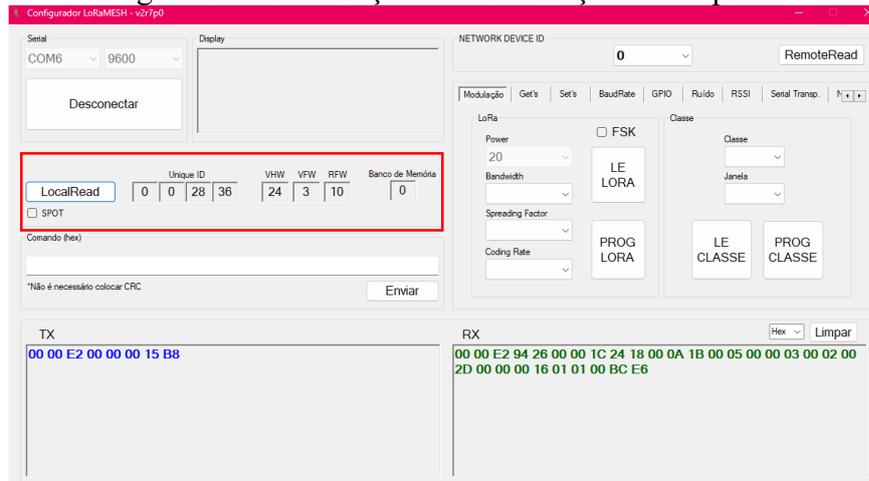
(Fonte: Elaborado pelo autor do projeto)

Após selecionar a interface de comunicação serial e clicar no botão 'Conectar', a ausência de qualquer mensagem no display indicará que a conexão foi estabelecida com sucesso. No entanto, se a mensagem 'porta COM já conectada em outro lugar' for exibida no display, será necessário reiniciar o sistema operacional para liberar a porta serial.

- Informações da placa RD42C

Na Figura 28 o botão ‘*LocalRead*’, localizado no retângulo destacado do lado esquerdo, carrega todas as informações de fábrica do dispositivo LoRaMESH. Com isso podemos visualizar qual é o *Unique ID* e qual versão do *hardware* e *firmware* integrado.

Figura 28: Visualização das informações da 1ª placa

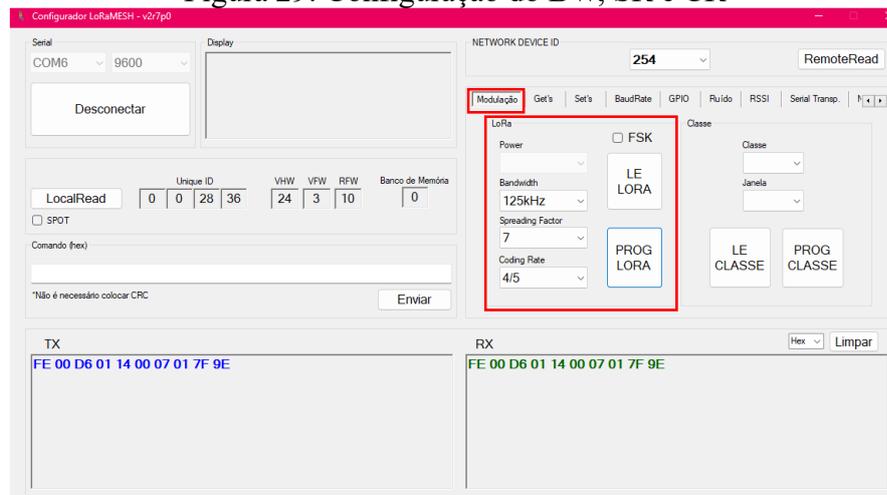


(Fonte: Elaborado pelo autor do projeto)

- Configuração do BW - SF - CR

Na figura 29 é possível observar algumas abas. Na aba ‘Modulação’, é definido as configurações adequadas dos BW, SF e o CR, identificados na seção 1.7.2. e na tabela 5. Após o preenchimento clicar em ‘PROG LORA’. Vale ressaltar que o FSK deve estar desabilitado. Com isso, pode-se observar o envio e o recebimento da informação gravadas localizada na caixa TX e RX.

Figura 29: Configuração do BW, SR e CR



(Fonte: Elaborado pelo autor do projeto)

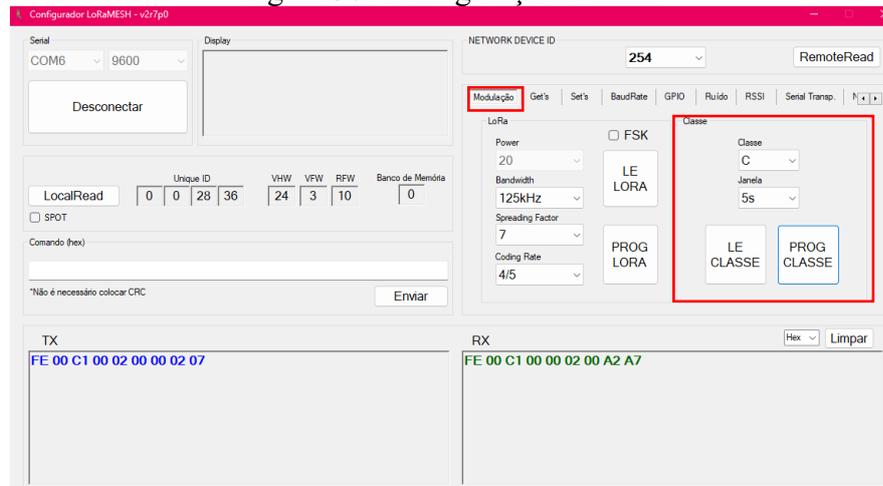
- Configuração da Classe

A configuração da classe é determinada de acordo com a funcionalidade do dispositivo. O objetivo deste processo é tornar esse LoRaMESH como o principal, ou seja, *master*.

Deve-se tomar alguns cuidados com esse processo da configuração. Na caixa ‘Classe’, existe duas opções, A e C presente na seção 1.7.3.1. A classe A, tem o objetivo de enviar informações dos escravos para o mestre e responder comandos via UART. Quando se utiliza a classe A o dispositivo não funciona como um repetidor e a tensão UART utilizada é 3.3V. A classe C, tem como objetivo de enviar e receber informações, o dispositivo funciona como um repetidor e a tensão de comunicação UART utilizada é 5V.

Como o dispositivo configurado é o *master*, ele deve estar sempre ativo para receber continuamente informações dos dispositivos *slaves*, e para quando precisar mandar algum comando específico. A figura 30 ilustra o processo de configuração da classe.

Figura 30: Configuração a classe



(Fonte: Elaborado pelo autor do projeto)

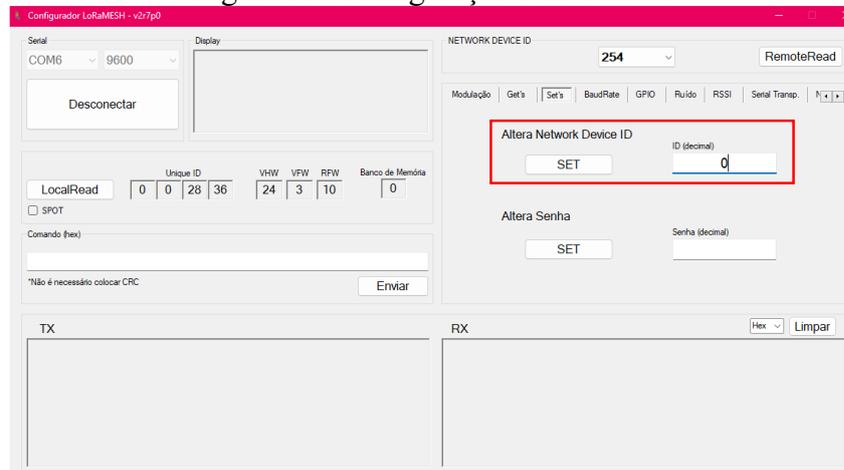
Após definir o tipo de dispositivo que foi programado, clicar em ‘PROG CLASSE’.

- Configurando Network Device ID

De acordo com o próprio datasheet do LoRaMESH, os dispositivos *masters* deve ter o ID definido como 0, e os *slaves* com os números seguintes, ou de uma forma que seja padronizada, para organização da rede. É destacado que o ID 2047 permite transmitir um pacote para todos os rádios da rede mesh, e só pode ser utilizado em um dispositivo mestre.

Na figura 31 pode-se observar que o ID do dispositivo que estamos configurando é 0, portanto ele é o mestre. Após definir o ID do dispositivo, clicar em ‘SET’.

Figura 31: Configuração o ID Master



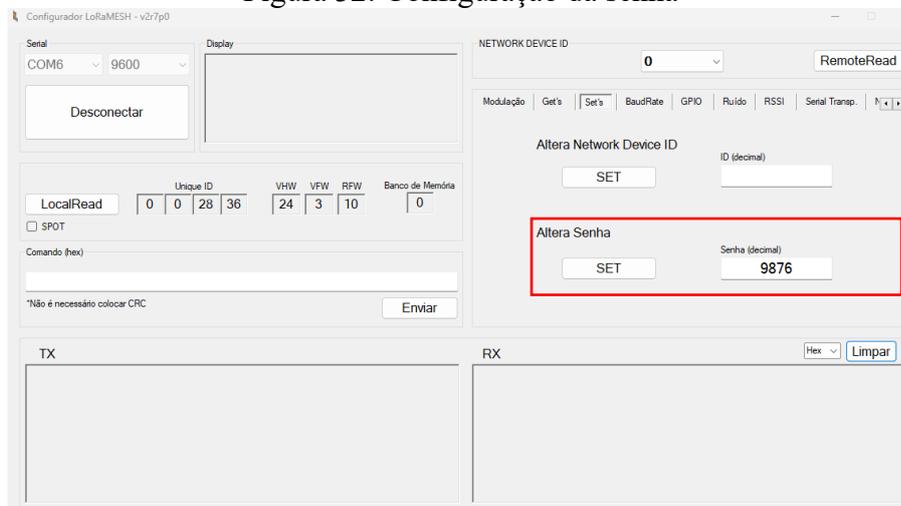
(Fonte: Próprio Autor)

- Configurando Senha

A definição da senha é de suma importância, pois define qual rede o dispositivo está sendo direcionado. Se a senha de um dispositivo for diferente de um outro, o dispositivo entende que ambos estão em redes distintas. Na figura 32 mostra a senha que configuramos para o nosso dispositivo, 9876.

Após definir a senha do dispositivo, clicar em ‘SET’.

Figura 32: Configuração da senha



(Fonte: Elaborado pelo autor do projeto)

2.3.3.4. Configuração do dispositivo *Slave*

A configuração do *slave* deve-se prestar muita atenção nas configurações, pois devem estar configurados com algumas informações que foi adicionada no dispositivo *master*. Com as configurações definidas, obteve êxito na comunicação entre ambos os dispositivos e o acesso remoto do *master* com o dispositivo *slave*.

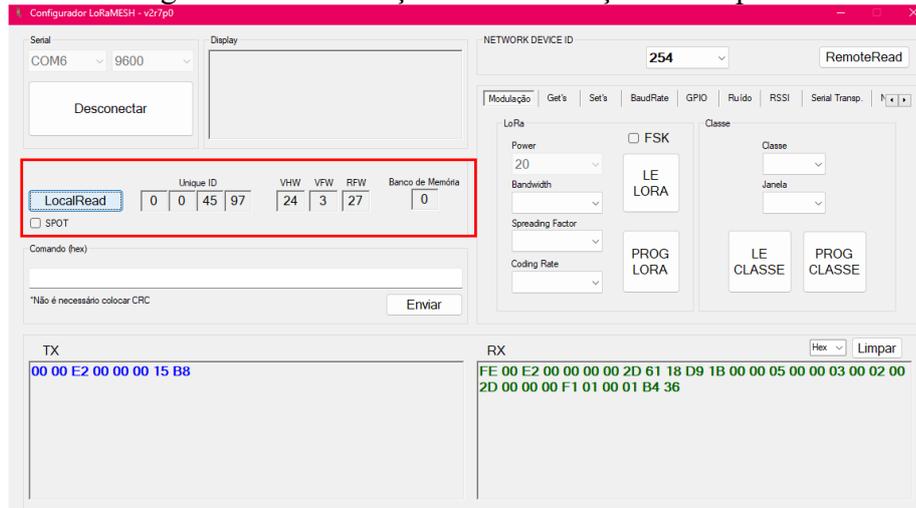
Neste tópico aborda o processo de configuração que foi realizado no dispositivo ‘*SLAVE*’. Com um outro dispositivo RD42C, deve-se novamente conectá-lo com as mesmas instruções realizadas no dispositivo (*master*) anterior, visualizada na figura 26, sessão 2.3.3.3.

Os dois dispositivos testados, o parâmetro ‘*Network Device ID*’ é configurado com o número 254.

Após conectado, clique em ‘*LocalRead*’, nota-se na figura 33 que o UNIQUE ID do dispositivo mudou, pois, como abordado anteriormente, o UNIQUE ID é único para cada dispositivo LoRaMESH.

- Analisando parâmetros do UniqueID

Figura 33: Visualização das informações da 2ª placa

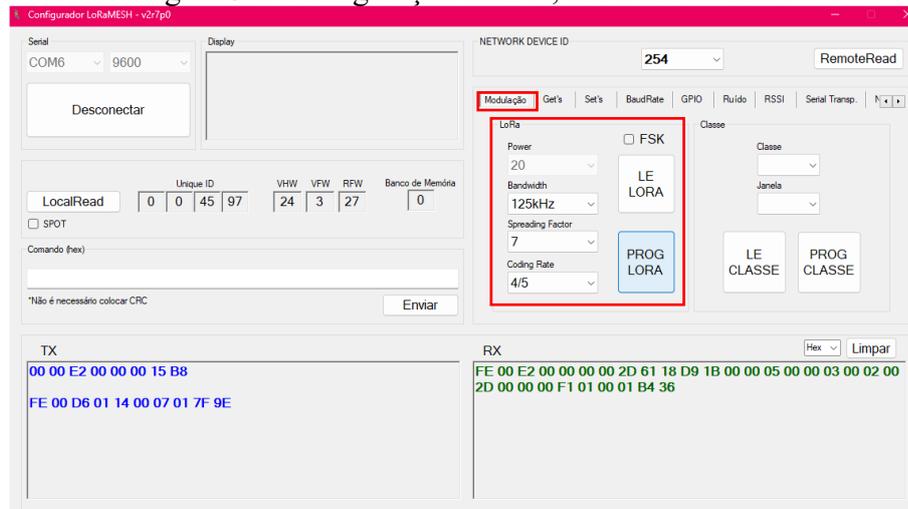


(Fonte: Elaborado pelo autor do projeto)

- Configurando do BW - SF - CR

A comunicação entre ambos os dispositivos só é possível se for definido as configurações iguais, assim como mostrado na figura 34. No entanto a frequência de comunicação e a taxa de dados ficaram da mesma forma que anteriormente pois, os dispositivos precisam conversar na mesma frequência.

Figura 34: Configuração do BW, SR e CR do slave

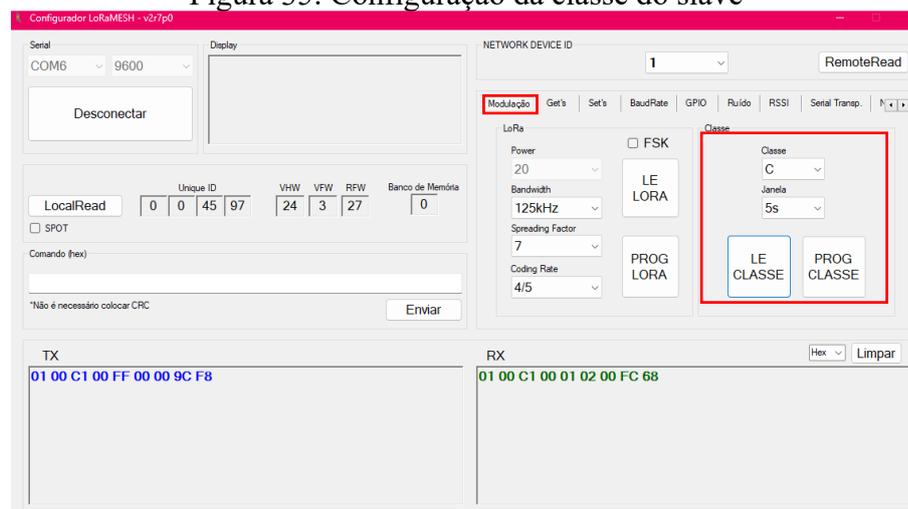


(Fonte: Elaborado pelo autor do projeto)

- Configurando da classe

Na figura 35 a configuração da Classe foi definida como classe C. Este trabalho permitiu que múltiplos dispositivos em um campo enviem os dados para o dispositivo *master*. Para alcançar isso, é necessário implementar o roteamento de dados entre os dispositivos *slaves* até o dispositivo *master*.

Figura 35: Configuração da classe do slave



(Fonte: Elaborado pelo autor do projeto)

Alguns dispositivos podem estar distantes do *master*, exigindo que outros dispositivos *slaves* atuem como roteadores. A classe C é ideal para essa finalidade, pois permite o roteamento

de dados, ao contrário da classe A, que apenas envia informações diretamente ao *master* sem a capacidade de roteamento.

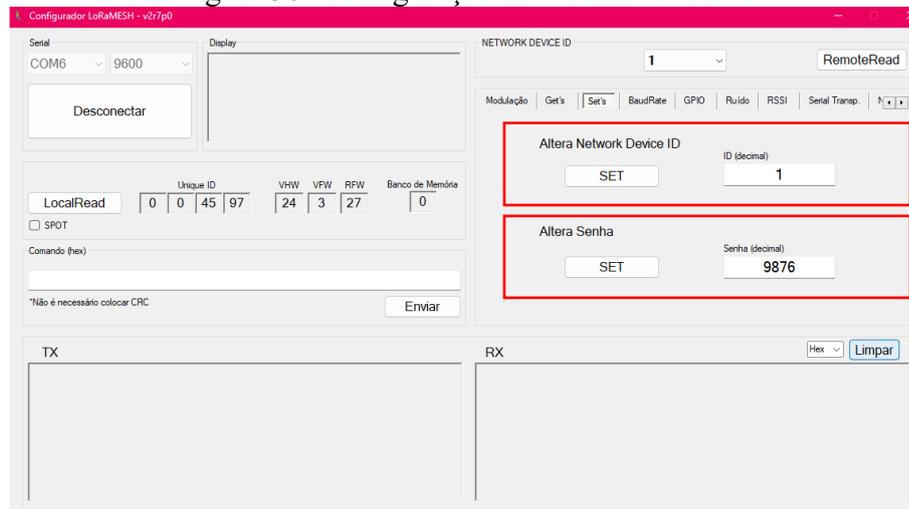
Na figura 23, pode-se observar que o dispositivo ‘*SLAVE 5*’ longe do *master*. Conseqüentemente, é necessário que outros dispositivos escravos realizem o roteamento dos dados fornecidos por ele até o recebimento pelo dispositivo mestre. É importante destacar que esta figura serve como uma analogia, e a disposição espacial dos dispositivos pode ser ajustada conforme as necessidades específicas do agricultor.

- Configuração do Network Device ID e Senha

Na figura 36 foi definido o ‘*Network Device ID*’ como 1, para manter uma organização e um padrão aos dispositivos em campo. É importante que a senha seja a mesma que a do dispositivo *master*, pois entende-se que estão na mesma rede. Se os dispositivos não estiverem na mesma rede não é possível a comunicação entre ambos e o acesso remoto.

Após a definição do número do ‘*Network Device ID*’ e a ‘*Senha*’, clique em ‘*SET*’ em ambas as configurações, de acordo com a figura 36

Figura 36: Configuração ID e senha do slave



(Fonte: Elaborado pelo autor do projeto)

2.3.3.5. Testes no *software* do Configurador LoRaMESH – v2r7p0

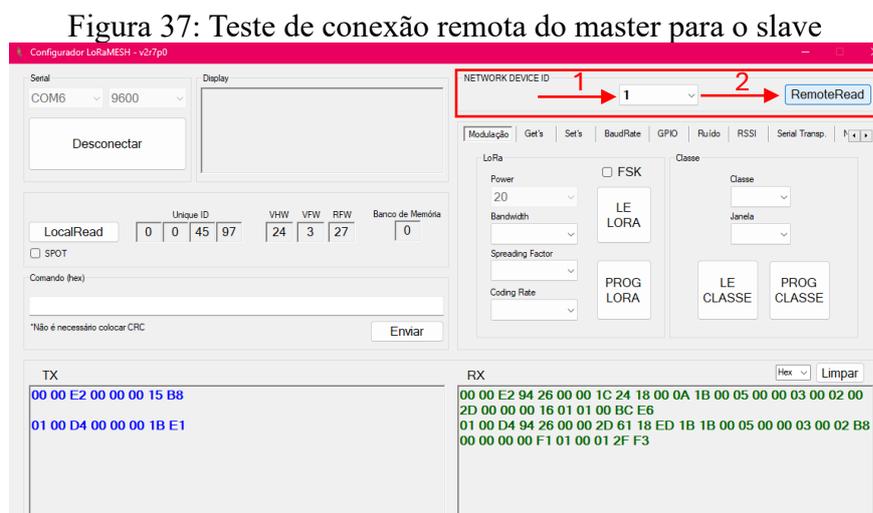
Durante os testes no *software* do Configurador LoRaMESH – v2r7p0, foram realizadas avaliações abrangentes para verificar a funcionalidade e eficiência do sistema. O objetivo

principal foi validar a capacidade do *software* em configurar e gerenciar redes LoRaMESH de maneira eficaz e intuitiva.

- Teste de Comunicação Remota

De acordo com o datasheet, a comunicação remota é sempre realizada do dispositivo Mestre para o dispositivo Escravo, nunca ao contrário. Para realizar o teste foi necessário a configuração de ambos, conforme realizado nos tópicos anteriores. Com isso conectamos o dispositivo RD42C *Master* ao computador que possui o *software* de configuração LoRaMESH instalado. Certifique-se de ligar o dispositivo *slave*, foi utilizada como fonte de alimentação o microcomputador Raspberry PI, mas qualquer fonte de alimentação adequada pode ser usada. Não é necessário abrir o *software* no escravo, apenas ligá-lo, pois as configurações já estão gravadas na placa.

No dispositivo mestre, foi estabelecida a conexão com o *software* de configuração LoRaMESH. Após a conexão, dentro do campo denominado 'Network Device ID', selecione o ID do dispositivo ao qual deseja conectar, conforme demonstrado no processo da figura 37. Como o dispositivo escravo está configurado com este ID, clique em '*RemoteRead*'.



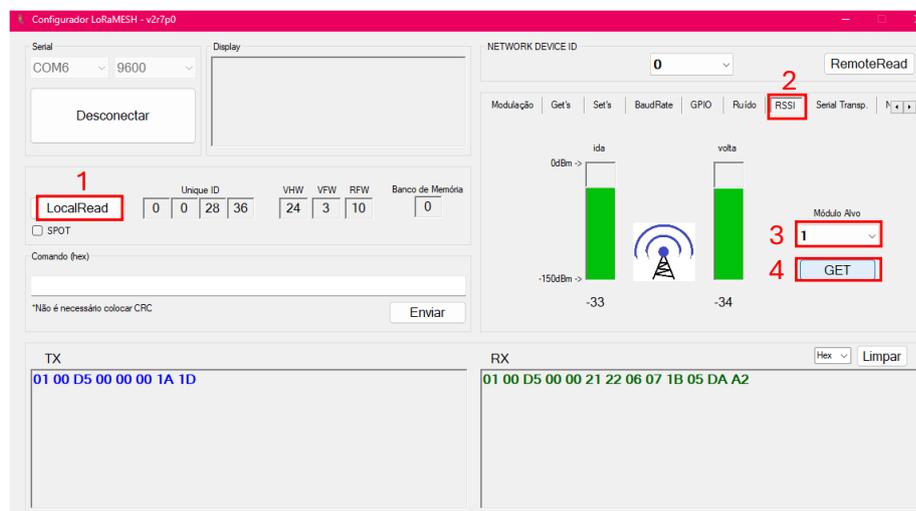
(Fonte: Elaborado pelo autor do projeto)

- Teste de análise RSSI

A aba ‘RSSI’ (*Received Signal Strength Indication*) mostra a distância que os dispositivos *slaves* e *master* estão. Como mostrado na figura 38, observa-se o primeiro gráfico que é a de potência de comunicação de ida, já o segundo a potência da volta. Os dispositivos estavam aproximadamente 1 metro de distância, quando mais longe maior o valor.

Para análise da distância RSSI, clicar em ‘LocalRead’, ir na aba “RSSI”, selecionar o dispositivo e clicar em ‘GET’.

Figura 38: Teste do RSSI do master até o slave



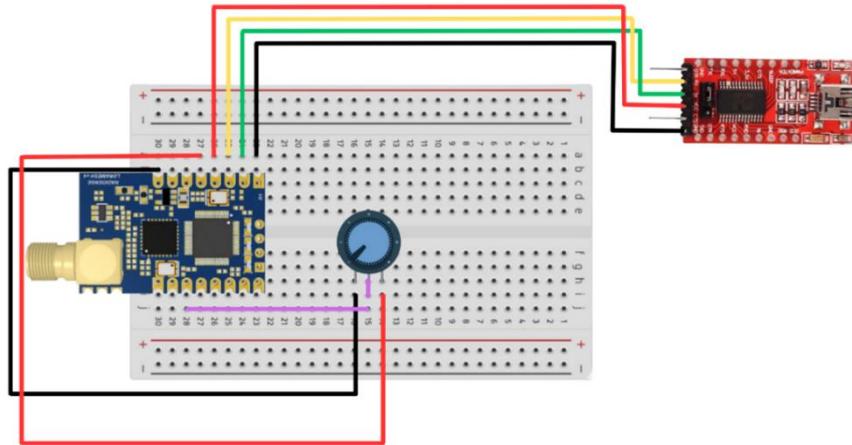
(Fonte: Elaborado pelo autor do projeto)

- Teste de Leitura Analogia

A leitura analogia foi observada no display, sendo o valor da conversão de 12 bits de AD para a tensão de 3,3V. Com a utilização de um potenciômetro pode-se ter os valores gerados de 0 a 3,3V.

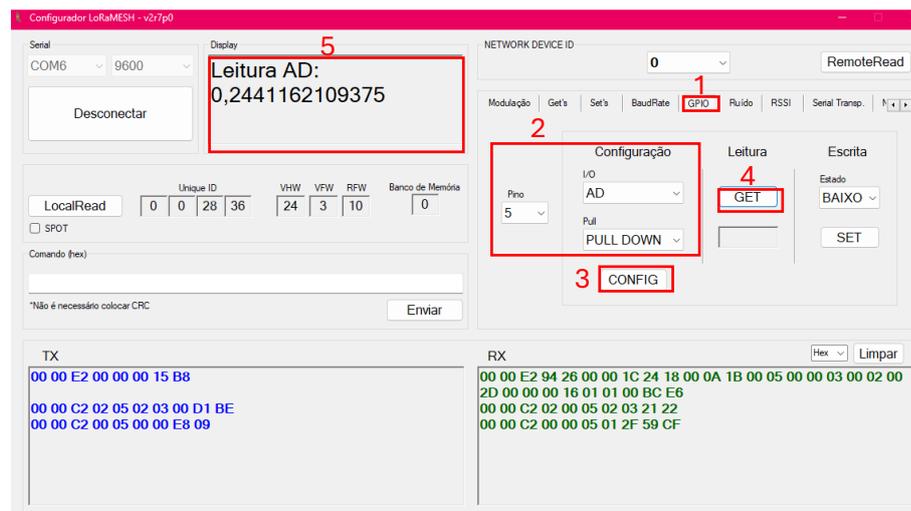
Na figura 39 mostra o desenvolvimento do circuito do teste, a visualização das informações que foram geradas no display do AD, podem ser analisadas na figura 40. Foi utilizado os dispositivos LoRaMESH RD42C, conversor FTDI232 e um potenciômetro para variação do valor de tensão do AD.

Figura 39: Montagem circuito RD42C + FTDI232 + Potenciômetro



(Fonte: Elaborado pelo autor do projeto)

Figura 40: Leitura da porta AD do Configurador LoRaMESH



(Fonte: Elaborado pelo autor do projeto)

- Teste de envio do comando 0xC2 com repetição de 5 segundos

Com o *software*, possibilitou o envio de comando para coletar informações do dispositivo 1 a cada segundo programado, porém não foi possível visualizar os valores no display.

Cada comando no manual do LoRaMESH é específico e está em formato hexadecimal. No teste realizado, utilizou-se o comando para identificar o valor gerado no GPIO 5 do *slave* 1, repetindo-o por 5 segundos. Na figura 41 e 42 é possível visualizar a formato de envio e resposta

do pacote com o determinado comando, como representado na figura 43. A figura 41 está mostrando o formato do pacote que foi enviado no quadrado do TX.

Figura 41: Visualização do formato do pacote enviado com o comando 0XC2

Byte	0	1	2	3	4	5	6	7	8
	01	00	C2	00	05	00	00	F8	C9
Dado	ID (LSB)	ID (MSB)	0XC2	0X02	PINO	PULL	INOUT	CRC (LSB)	CRC (MSB)

(Fonte: Elaborado pelo autor do projeto)

A figura 42 está mostrando o formato do pacote que foi recebido no quadrado do RX. Os detalhes de cada byte são encontrados no *datasheet*.

Figura 42: Visualização do formato do pacote recebido com o comando 0XC2

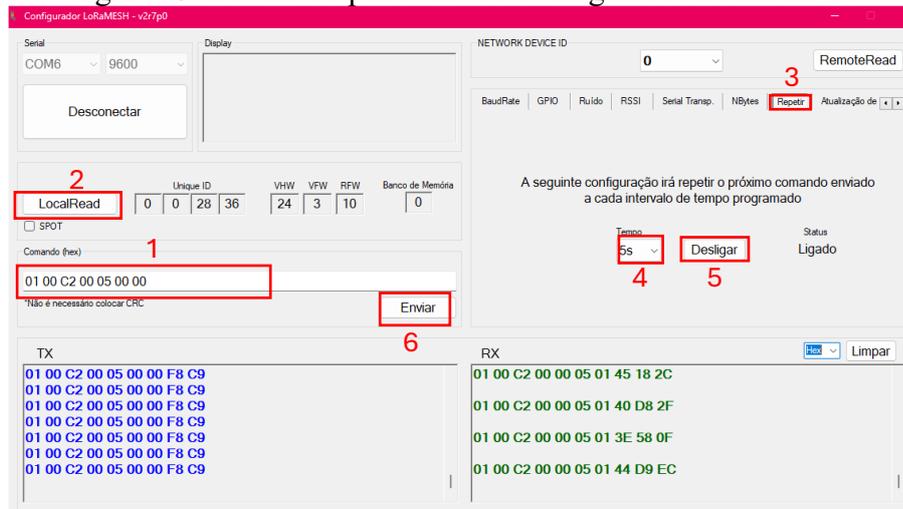
Byte	0	1	2	3	4	5	6	7	8	9
	01	00	C2	00	00	05	01	44	D9	EC
Dado	ID (LSB)	ID (MSB)	0XC2	0X02	Erro	PINO	PULL	INOUT	CRC (LSB)	CRC (MSB)

(Fonte: Elaborado pelo autor do projeto)

Com isso é possível ter uma visualização melhor dos seus determinados parâmetros. Abaixo segue o passo a passo que foi realizado para o teste de envio de comando repetido, na figura 43 também mostra o processo realização, porém algumas informações abaixo não são mostradas na figura.

- 1º Passo: Clicar em LocalRead;
- 2º Passo: Realizar o 'Teste comunicação remota';
- 3º Passo: Clicar na aba GPIO;
- 4º Passo: Realizar a 2ª etapa do 'Teste de leitura analógica';
- 6º Passo: copiar o comando apresentado no TX após a leitura da porta AD, e colar em 'Comando (hex)'. Não necessita do CRC como informado no sistema;
- 6º Passo: Clicar em LocalRead, verifique se está no ID 0;
- 7º Passo: Ir na aba 'Repetir', selecione 5 segundos e clicar em 'ligar';
- 8º Passo: Clicar em enviar abaixo do comando;

Figura 43: Leitura da porta AD do Configurador LoRaMESH



(Fonte: Elaborado pelo autor do projeto)

2.3.3.6. Aplicação de configurações do Arduino do dispositivo *Master*

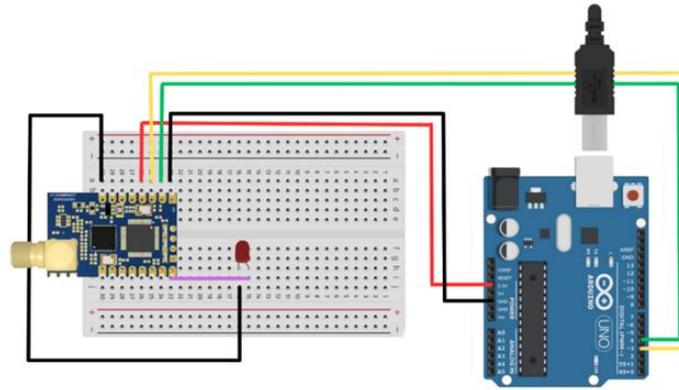
Neste tópico foi iniciado a montagem do projeto em questão. Para este experimento utilizou-se um Arduino, um LED, o dispositivo RD42C e os cabos jumpers.

O objetivo do *master* é coletar as informações que o *slave* 1, que retorna o valor gerado do sensor de umidade do solo, no caso do projeto foi simulado um sensor com um potenciômetro, e depois testado com o sensor real que está representado na montagem do *slave*, figura 47. O AD tem um valor de 8bit que varia a leitura de 0 a 255. Quando o dispositivo identificasse que o valor recebido da porta analógica do *slave* fosse menor que 150 (valor de simulação para realização do teste), ele informava que a área estava úmida. Caso o valor recebido fosse maior ou igual a 150 o mestre identificava que o solo estaria seco e acendia o LED para poder sinalizar.

- **Montagem do circuito do projeto**

A Figura 44 mostra a montagem do circuito do mestre, projetado para sinalizar quando recebesse dados indicando que a área estava seca.

Figura 44: Montagem circuito RD42C + Arduino + LED



(Fonte: Elaborado pelo autor do projeto)

- **Configurações iniciais da placa RD42C na IDE do Arduino**

Para o funcionamento adequadamente do programa realizado na IDE do Arduino, precisou utilizar a biblioteca do *LoRaMESH-Master*, pois temos que chamar algumas funções que estão dentro desta biblioteca.

Na figura 45, foi realizado a programação das configurações. O dispositivo *master* consegue configurar os *slaves*, o que possibilita uma visualização melhor dos dispositivos *slaves* configurados.

Figura 45: Programação da configuração no IDE Arduino para o dispositivo Master

```
#include "LoRaMESH.h"
#include <SoftwareSerial.h>

SoftwareSerial SerialCommand(3, 4);
LoRaMESH lora(&SerialCommand);

uint8_t ID = 0;

void setup() {
  Serial.begin(115200);
  delay(2000);
  SerialCommand.begin(9600);
  lora.begin(true);

  lora.setnetworkId(ID);
  lora.config_bps(BW125, SF_LoRa_7, CR4_5);
  lora.config_class(LoRa_CLASS_C, LoRa_WINDOW_15s);
  lora.setPassword(9876);

  if (ID == 0)
    lora.config_digital_gpio(LoRa_GPIO0, LoRa_NOT_PULL, LoRa_INOUT_DIGITAL_OUTPUT, LoRa_LOGICAL_LEVEL_LOW);
  if (ID == 1)
    lora.config_analog_gpio(LoRa_GPIO5);

  Serial.println("      Informações da Placa ");
  Serial.println("-----");
  Serial.println("LocalID: " + String(lora.localId));
  Serial.println("UniqueID: " + String(lora.localUniqueId));
  Serial.println("Password: " + String(lora.registered_password));
  Serial.println("-----");
}
```

(Fonte: Elaborado pelo autor do projeto)

- **Configurações do processo para coletar as informações do *Slave 1***

Na Figura 46, foi realizada a programação para visualização dos dados coletados pelo mestre, permitindo a visualização dos dados gerados sobre a leitura da porta analógica (GPIO

5) do escravo. Também foi realizado a conversão para uma tensão, onde o valor gerado pela porta de entrada do AD é multiplicado por 3.3V e dividido por 255, na qual permitiu uma variação de tensão.

Figura 46: Programação na IDE Arduino para o dispositivo master

```
void loop() {
  if(ID == 0 && lora.read_gpio(1, 5) > 150){
    static bool sensorUmi = false;
    if(sensorUmi)
      Serial.println("Valor Convertido: " + String(lora.read_gpio(1, 5) * 3.3/255.0));
      Serial.println("GPIO 5 - 1: " + String(lora.read_gpio(1, 5)));
      Serial.println("SECO ");
    }
  else{
    if(ID == 0 && lora.read_gpio(1, 5) < 150)
      Serial.println("Valor Convertido: " + String(lora.read_gpio(1, 5) * 3.3/255.0));
      Serial.println("GPIO 5 - 2 : " + String(lora.read_gpio(1, 5)));
      Serial.println("UMIDO ");
    }
  }
  if(ID == 0 && lora.read_gpio(1, 5) > 150){
    lora.write_gpio(0, LoRa_GPIO00, LoRa_LOGICAL_LEVEL_HIGH);
  }
  else{
    lora.write_gpio(0, LoRa_GPIO00, LoRa_LOGICAL_LEVEL_LOW);
    delay(1000);
  }
  delay(1000);
}
```

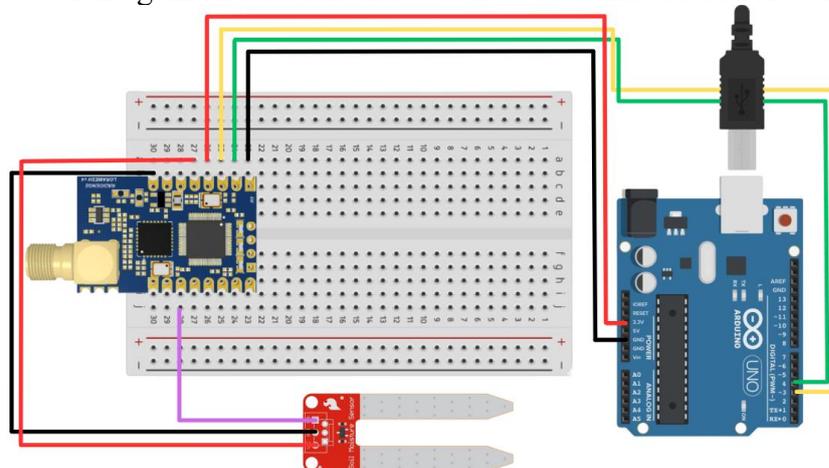
(Fonte: Elaborado pelo autor do projeto)

2.3.3.7. Aplicação de configurações do Arduino do dispositivo *Slave*

- **Montagem do circuito do projeto**

A figura 47 representa a montagem do circuito do projeto realizado com o dispositivo *slave*. Os principais dispositivos utilizados nesta etapa foi: Um Arduino, um sensor de umidade e um dispositivo LoRaMESH –RD42C.

Figura 47: Montagem circuito RD42C + Sensor de Umidade do solo + Arduino



(Fonte: Elaborado pelo autor do projeto)

- **Configurações da placa RD42C *Slave* na IDE do Arduino**

Conforme mostrado na figura 48, a configuração do *slave* precisou somente das configurações iniciais, pois quem administra as configurações que ele precisa ter é o *master*.

Figura 48: Programação da configuração no IDE Arduino para o dispositivo slave

```
#include "LoRaMESH.h"
#include <SoftwareSerial.h>

SoftwareSerial SerialCommand(3, 4);
LoRaMESH lora(&SerialCommand);

uint8_t ID = 1;

void setup() {
  Serial.begin(115200);
  delay(2000);
  SerialCommand.begin(9600);
  lora.begin(true);

  lora.setnetworkid(ID);
  lora.config_bps(BW125, SF_LoRa_7, CR4_5);
  lora.config_class(LoRa_CLASS_C, LoRa_WINDOW_15s);
  lora.setpassword(9876);

  Serial.println("          Informações da Placa" );
  Serial.println("-----" );
  Serial.println("LocalID: " + String(lora.localId));
  Serial.println("UniqueID: " + String(lora.localUniqueId));
  Serial.println("Password: " + String(lora.registered_password));
  Serial.println("-----" );
}

void loop() {
}
```

(Fonte: Elaborado pelo autor do projeto)

2.3.4. Resultados e Discussão

Os principais objetivos iniciais centraram-se na concretização dos conceitos abordados no projeto, na exploração dos dispositivos demonstrados, na montagem do sistema Arduino com LoRaMESH e na execução de configurações utilizando o configurador LoRaMESH – v2r7p0 e na plataforma Arduino IDE.

Com base nos conteúdos abordados neste projeto, adquirimos um amplo conhecimento e compreensão de alguns dispositivos IoT, representativos de uma vasta gama existente globalmente, explorando suas funcionalidades e características distintivas. Abordamos também os conceitos dos termos e plataformas que vamos utilizar diante o desenvolvimento do projeto.

Portanto, os resultados obtivemos, mostraram que existe uma possibilidade de trabalhar com o dispositivo RD42C *Radioenge* criando uma rede LoRaMESH e utilizando suas portas digitais e analógicas em campo, trazendo uma boa comunicação entre os dispositivos, garantindo uma estabilidade na conexão e um bom fluxo de transferência de dados.

A porta AD gera um valor de 8 bits, que varia de 0 a 255. Este valor permitiu a conversão para uma tensão, onde o valor gerado pela porta de entrada do AD é multiplicado por 3.3V e dividido por 255, resultando na variação do valor de tensão correspondente.

Figura 49: Recebimento dos dados do slave para o master

```
TX: 0100C200050000F8C9
RX: 0100C20000050047987D
TX: 0000C201000000F9F4
TX: 0100C200050000F8C9
RX: 0000C201000000034420100C20000050047987D
TX: 0100C200050000F8C9
RX: 0100C20000050047987D
TX: 0100C200050000F8C9
RX: 0100C20000050048D879
TX: 0100C200050000F8C9
RX: 0100C20000050048D879
Valor Convertido: 0.93
TX: 0100C200050000F8C9
RX: 0100C20000050047987D
GPIO 5 - 2 : 71.00
UMIDO
TX: 0100C200050000F8C9
RX: 0100C20000050098D9E5
TX: 0000C201000100F864
TX: 0100C200050000F8C9
RX: 0000C20100000001F5820100C200000500965821
TX: 0100C200050000F8C9
RX: 0100C2000005009A5824
TX: 0100C200050000F8C9
RX: 0100C200000500965821
GPIO 5 - 1: 150.00
SECO
```

(Fonte: Elaborado pelo autor do projeto)

Com a programação, podemos adaptar funções para operações em campo. Por exemplo, neste projeto, utilizar sensores de umidade do solo para coletar dados. Quando o solo está seco,

essa informação é enviada ao controlador principal, que, por sua vez, aciona os irrigadores automaticamente até que o solo atinja o nível de umidade desejado.

CAPÍTULO III

3. Conclusão

Os resultados obtidos demonstraram que a rede LoRaMESH com *Radioenge* RD42C apresentou um desempenho estável e confiável na transmissão de dados entre os dispositivos.

A capacidade de enviar informações das portas digitais e analógicas permitiu um monitoramento preciso das condições agrícolas em tempo real, facilitando decisões rápidas e informadas pelos agricultores. A integração bem-sucedida de sensores e equipamentos à rede Mesh proporcionou uma visão abrangente e detalhada das operações agrícolas, contribuindo para a otimização dos recursos e aumento da produtividade.

Este estudo demonstrou que a implementação de uma rede LoRaMESH utilizando o *Radioenge* RD42C é uma solução viável e eficaz para aplicações de monitoramento e controle na Agricultura 4.0. A tecnologia LoRa® mostrou-se adequada para operar em ambientes rurais, oferecendo uma comunicação robusta e econômica que atende às necessidades específicas do setor agrícola moderno. Os resultados positivos indicam um potencial significativo para melhorar a eficiência operacional, reduzir custos e promover práticas agrícolas mais sustentáveis e responsáveis. Futuras pesquisas podem explorar ainda mais as capacidades da rede LoRaMESH em diferentes contextos agrícolas e expandir seu uso para outras aplicações industriais e comerciais.

BIBLIOGRAFIA

ANATEL (2018). **Faixa de Radiofrequência utilizáveis por Equipamentos de Radiação Restrita**. Acessado em 01 de janeiro de 2024.

ALFA INSTRUMENTO (2020). **Protocolo de Comunicação Serial Modbus RTU / ASCII – Versão 1.0**. Acessado em 12 de março de 2024.

ARMINGOL, J. M. *Master in Internet of Things: Applied Technologies*. Acessado em 20 de setembro de 2023.

AKYILDIZ, I.; WANG, X.(2009) *Wireless Mesh Networks*. [S.l.]: John Wiley & Sons. Acessado em 16 de fevereiro de 2024.

BANCO NACIONAL DE DESENVOLVIMENTO ECONÔMICO E SOCIAL (BNDES); MINISTÉRIO DA CULTURA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES (MCTIC). **Relatório sobre a implementação da Internet das Coisas (IoT) na agricultura**. Rio de Janeiro: BNDES, 2020.

BASTOS, C. **Matéria Prima**. Acessado em 26 de outubro de 2023.

BUENO, H. O. **Otimização de seleção de parâmetros de transmissão de redes LORA MESH de gateway single channel em ambientes rurais**. Acessado dia 10 de janeiro de 2023.

CELF, V. KAHN, R. *A Protocol for Packet Network Intercommunication*. Acessado em 10 de maio de 2024.

CUNHA, L. J.; VALIM, P. R. O. **Rede de Sensores Sem Fio para Monitoramento de Variáveis de Ambiente**. Acessado em 18 de outubro de 2023.

DELLALIBERA, L. A. (2018) **Técnicas de múltiplo acesso para redes LoRaWAN**. 2018. 48 f. trabalho de conclusão de curso (bacharelado - engenharia de telecomunicações) - Universidade Estadual Paulista Julio de Mesquita Filho, campus experimental de são joão da boa vista. p. 21. Acessado em 10 de janeiro de 2024.

DHANALAKSHMI, R.; SINHA, B. B. (2022) *Recent advancements and challenges of Internet of Things in smart agriculture: A survey*. Acessado em 18 de agosto de 2023.

ESTES, A. C. (2018) **Como começou essa história de transmitir informações sem fio**. Acessado em 18 de junho de 2024.

FAROOQ, M. S.; RIAZ, S.; ABID, A.; UMER, T.; ZIKRIA, Y. B. (2020) *Role of IoT Technology in Agriculture: A Systematic Literature Review*. Acessado em 25 de novembro de 2023.
do Rio dos Sinos – UNISINOS.

FÉLIX, R. P; MANUEL, T. (2022) **IoT (*Internet of Things*) Podemos Confiar?** Acessado em 13 de setembro de 2023.

FERREIRA, A. M.; NASCIMENTO, G. C.; Santos, L. C. (2023). **Tecnologia da Internet das Coisas na Agricultura 4.0.** Acessado em 20 de janeiro de 2024.

FLORES, É. (2021) **Implementação de Rede Mesh Utilizando Comunicação de Longo Alcance para Sensoriamento Remoto de Produção Agrícola.** Trabalho de Conclusão de Curso Graduado em Engenharia, pelo Curso de Engenharia Elétrica da Universidade do Vale

FOROUZAN, B. A; FEGAN, S. C. **Protocolo TCP/IP.** 3.ed. Porto Alegre: AMGH,2010.

GALLEGOS, R. (2023). ***What are IoT devices and what should you know about them?*** Acessado em 2 de agosto de 2023.

GIL, A. C. **Como elabora projeto de pesquisa.** 4. ed. São Paulo: Atlas, 2002. Acessado em 20 de junho de 2024.

KRAMP, T.; KRANENBURG, R. V.; LANGE, S. (2013) ***Introduction to the Internet of Things.*** Acessado 13 de setembro de 2023

LORA DEVELOPS SEMTECH. Acessado em 2 de janeiro de 2024.

MANTOVANI, E.C.; QUEIROZ, D. M; DIAS, G.P (1988). **Máquinas e operações utilizadas na agricultura de precisão.** Disponível em: Congresso brasileiro de engenharia,27. Acessado em 10 de outubro de 2023.

MANUAL LORAMESH (2024). Acessado em 02 de janeiro de 2024

MAKIYAMA, M. **O que é Arduino, para que serve, benefícios e projetos.** Acessado em 23 de outubro de 2023.

MARQUES, J. J. B.; BOCHIE, K. **LoRa.** Acessado em 09 de outubro de 2023.

MASSRUHÁ, S. M. F. S.; LEITE, M. A. A. (2017). **Agro 4.0 – Rumo à agricultura Digital.** Acessado em 18 agosto de 2023.

MAXWELL, J. C. (1865) **Uma Teoria Dinâmica do Campo Eletromagnético.** Acessado em 15 de março de 2024.

MEDEIROS, D. F. (2021) **Implementação e análise de protocolos de roteamento para redes mesh sem fio LoRa.** Pós Graduação em Engenharia Elétrica. Universidade Federal da Paraíba- UFPB.

MORDOR INTELLIGENCE (2023) **Tamanho do mercado de redes de malha sem fio e análise de participação – Tendências e previsões de crescimento (2024 – 2029).** Acessado em 11 de maio de 2024.

NASCIMENTO, J. M. A.; LUCENA, P. B. (2003) **Protocolo Modbus.** Acessado em 20 de abril de 2024.

NETO, E. S. (2017) **LoRa Arquitetura e Especificações**. Acessado em 20 de fevereiro de 2024.

NIARAKI, A. S. (2023) ***Internet of Things (IoT) review of review: Bibliometric overview since its foundation***. Acessado em 10 de agosto de 2023.

ORTIZ, F. M.; ALMEIDA, T. T.; Ferreira A. E.; Costa, L. H. M. K. **Caracterização de Desempenho de uma Rede LoRa em Ambientes Urbanos**. Acessado em 6 de outubro de 2023.

SAN, L. M. (2019) **O ABC do LoRa**. Acessado em 19 de outubro de 2023.

SAARI, M. (2018) ***LoRa-A Survey of Recent Research Trends***. Acessado em 15 de janeiro de 2024.

SILVA, C.E.; ESPEJO, B.S.M.M. (2020) ***Internet of Things (IoT) no Agronegócio***. Acessado em 30 de agosto de 2023.

SILVA, W. V. R.; MANN, R. S. (2020) **Agricultura de Precisão no Brasil: conjuntura atual, desafios e perspectivas *Precision Agriculture in Brazil: current situation, challenges and perspectives* Agricultura de Precisión en Brasil: situación actual, desafíos y perspectivas**. Acessado em 10 de janeiro de 2024.

TANENBAUM, A. S. **Redes de Computadores**. 5. ed. Person Education,2011.

THE THINKGS NETWORK. Acessado em: 01 de janeiro de 2024.

VIOLA, E.; MENDES, V. **Agricultura 4.0 e mudanças climáticas no Brasil**. Acessado em 09 de novembro de 2023.

WOLFERT, S.; VERDOUW, C.G.L; BOGAARDT, J.M. (2017) ***Big Data na Agricultura Inteligente***. Acessado em 17 de agosto de 2023.

WU, T. (2022) ***The Structural Features and Translation Skills of English in the Era of Rádio Communication Networks***. Acessado em 11 de março de 2023

ZAIT, A. (2022). ***Introduction to Arduino uno pinout***. Acessado em 24 de outubro de 2023.

ZHANG, J.; TAO, D (2020) ***Empowering Things with Intelligence: A Survey of the Progress, Challenges, and Opportunities in Artificial Intelligence of Things***. Acessado em 24 de março de 2024.

```

/*CRIADO POR: Bruna Strazza Quintero
DATA 09/06/2024
FATEC INDAIATUBA

```

```

-----
CONFIGURAÇÃO MASTER
-----

```

```

*/

```

```

#include "LoRaMESH.h"
#include <SoftwareSerial.h>

```

```

SoftwareSerial SerialCommand(3, 4);
LoRaMESH lora(&SerialCommand);

```

```

uint8_t ID = 0;

```

```

void setup() {
  Serial.begin(115200);
  delay(2000);
  SerialCommand.begin(9600);
  lora.begin(true);

```

```

  lora.setnetworkId(ID);
  lora.config_bps(BW125, SF_LoRa_7, CR4_5);
  lora.config_class(LoRa_CLASS_C, LoRa_WINDOW_15s);
  lora.setpassword(9876);

```

```

  if(ID == 0)
    lora.config_digital_gpio(LoRa_GPIO0, LoRa_NOT_PULL,
LoRa_INOUT_DIGITAL_OUTPUT, LoRa_LOGICAL_LEVEL_LOW);
  if(ID == 1)
    lora.config_analog_gpio(LoRa_GPIO5);

```

```

  Serial.println("      Informações da Placa" );
  Serial.println("-----" );
  Serial.println("LocalID: " + String(lora.localId));
  Serial.println("UniqueID: " + String(lora.localUniqueId));
  Serial.println("Password: " + String(lora.registered_password));
  Serial.println("-----" );

```

```

}

```

```

void loop() {

  if(ID == 0 && lora.read_gpio(1, 5) > 150){
    static bool sensorUmi = false;
    if(sensorUmi)

```

```
Serial.println("Valor Convertido: "+ String(lora.read_gpio(1, 5) *3.3/255.0));
Serial.println("GPIO 5 - 1: " + String(lora.read_gpio(1, 5)));
Serial.println("SECO ");
}

else{
  if(ID == 0 && lora.read_gpio(1, 5) < 150)
    Serial.println("Valor Convertido: " + String(lora.read_gpio(1, 5) *3.3/255.0));
    Serial.println("GPIO 5 - 2 : " + String(lora.read_gpio(1, 5)));
    Serial.println("UMIDO ");
  }

if(ID == 0 && lora.read_gpio(1, 5) > 150){
  lora.write_gpio(0, LoRa_GPIO0, LoRa_LOGICAL_LEVEL_HIGH);
}
else{
  lora.write_gpio(0, LoRa_GPIO0, LoRa_LOGICAL_LEVEL_LOW);
  delay(1000);
}
delay(1000);
}
```

```

/*CRIADO POR: Bruna Strazza Quintero
DATA 09/06/2024
FATEC INDAIATUBA

```

```

-----
CONFIGURAÇÃO SLAVE
-----

```

```

*/
#include "LoRaMESH.h"
#include <SoftwareSerial.h>

SoftwareSerial SerialCommand(3, 4);
LoRaMESH lora(&SerialCommand);

uint8_t ID = 1;

void setup() {
  Serial.begin(115200);
  delay(2000);
  SerialCommand.begin(9600);
  lora.begin(true);

  lora.setnetworkId(ID);
  lora.config_bps(BW125, SF_LoRa_7, CR4_5);
  lora.config_class(LoRa_CLASS_C, LoRa_WINDOW_15s);
  lora.setpassword(9876);

  Serial.println("      Informações da Placa" );
  Serial.println("-----" );
  Serial.println("LocalID: " + String(lora.localId));
  Serial.println("UniqueID: " + String(lora.localUniqueId));
  Serial.println("Password: " + String(lora.registered_password));
  Serial.println("-----" );

```

```
}
```

```
void loop() {
```

```
}
```