

UNIVERSIDADE FEDERAL DE SÃO CARLOS– UFSCAR  
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA– CCET  
DEPARTAMENTO DE COMPUTAÇÃO– DC  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO– PPGCC

**Alex Fabiano Garcia**

**Sistema Baseado em Internet das  
Coisas para o Rastreo Aplicado à  
Logística da Cadeia de Frio**

São Carlos  
2024



**Alex Fabiano Garcia**

**Sistema Baseado em Internet das  
Coisas para o Rastreo Aplicado à  
Logística da Cadeia de Frio**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências Exatas e de Tecnologia da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Sistemas Distribuídos, Arquiteturas e Redes de Computadores (SDARC)

Orientador: Wanderley Lopes de Souza

São Carlos

2024





# UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia  
Programa de Pós-Graduação em Ciência da Computação

---

## Folha de Aprovação

---

Defesa de Dissertação de Mestrado do candidato Alex Fabiano Garcia, realizada em 28/11/2024.

### Comissão Julgadora:

Prof. Dr. Wanderley Lopes de Souza (UFSCar)

Prof. Dr. Helio Crestana Guardia (UFSCar)

Profa. Dra. Rossana Maria de Castro Andrade (UFC)



*Em memória de meu pai.*



---

# Agradecimentos

---

Agradeço primeiramente à minha amada esposa, Amanda, pela parceria, compreensão e incentivo durante toda a nossa jornada compartilhada. Seu apoio foi, mais uma vez, fundamental para a conclusão de mais uma etapa na minha vida.

Ao meu pai, Célio, que nunca me deixou faltar nada diante de todas as adversidades e me mostrou que o trabalho bem-feito e a dignidade são o caminho a ser trilhado. À minha mãe, Dirce, que sempre fez de tudo para que eu pudesse frequentar a escola e sempre me amou incondicionalmente. Aos meus irmãos, Gustavo e Marcelo, meus melhores amigos e que sempre acreditaram em mim e me incentivaram em minhas escolhas.

Ao meu orientador, professor Dr. Wanderley Lopes de Souza, pela oportunidade concedida, compreensão diante do caminho sinuoso e pelos ensinamentos repassados durante o desenvolvimento deste trabalho.

Ao Programa de Pós-Graduação em Ciência da Computação (PPG-CC) da Universidade Federal de São Carlos (UFSCar), por fornecer os recursos humanos e a infraestrutura necessários ao desenvolvimento deste trabalho. À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), através do PPG-CC da UFSCar, e à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), através do Instituto Nacional de Ciência e Tecnologia (INCT) Medicina Assistida por Computação Científica (MACC), por fornecerem parte dos recursos financeiros necessários ao desenvolvimento deste trabalho.



---

# Resumo

---

Esta dissertação descreve o projeto, a implementação e a validação de um sistema baseado em Internet das Coisas para o rastreamento aplicado à logística da cadeia de frio (SBIdC-RALCF). O sistema visa a resolver desafios de logística em áreas remotas usando tecnologias de Internet das Coisas (IdC) para monitorar e gerenciar as condições de produtos perecíveis durante o transporte, garantindo sua qualidade e segurança. A solução proposta abrange vários dispositivos, tipos de sensores e integra-se com múltiplos provedores de rede *Long Range Wide Area Network* (LoRaWAN) para melhorar a cobertura. Ela visa dar suporte aos tomadores de decisão em parcerias público-privadas que se aproveitam da infraestrutura existente e investem em redes de baixo custo para abordar questões sociais em regiões periféricas. O estudo mostra como ferramentas de simulação de cobertura de antenas provêm boa acurácia se comparadas com a distância efetiva dos gateways que recebem o sinal *Long Range* (LoRa). Testes de campo nos Países Baixos demonstram a eficácia do sistema em cenários do mundo real, mostrando recursos como geolocalização sem GPS por multilateração, comunicação de longo alcance e o potencial para aplicar a solução em outros domínios além da logística da cadeia de frio.

**Palavras-chave:** Internet das Coisas. LoRaWAN. Transporte. Independência de GPS. Multilateração. Cadeia de Frio. Produtos Perecíveis.



---

# Abstract

---

This study describes the design, implementation, and validation of the System Based on the Internet of Things for Tracking Cold Chain Logistics (SBIdC-RALCF). The system aims to solve logistics challenges in remote areas by using IoT technologies to monitor and manage the conditions of perishable goods during transportation, ensuring their quality and safety. The proposed solution encompasses various sensor types and integrates multiple Long Range Wide Area Network (LoRaWAN) providers to improve coverage. It aims to support decision-makers in public-private partnerships that leverage existing infrastructure and invest in cost-effective networks to address social issues in outlying regions. The study shows how antenna coverage simulation tools provide good accuracy compared to the effective distance of the gateways that receive the Long Range (LoRa) signal. Field tests in the Netherlands demonstrate the system's effectiveness in real-world scenarios, showing features such as GPS-free geolocation by multilateration, long-range communication, and the potential for applying our solution in other domains beyond cold chain logistics.

**Keywords:** Internet of Things. LoRaWAN. Transport. GPS-free. Multilateration. Cold Chain. Perishable Goods.



---

# Lista de ilustrações

---

Figura 1 – Um <i>framework</i> para a Ciência do Projeto . . . . .	36
Figura 2 – As tarefas do Ciclo de Projeto dentro de um Ciclo de Engenharia . . . . .	38
Figura 3 – NodeMCU V3 conectada aos módulos DHT22 e SX1276 . . . . .	39
Figura 4 – WiFi LoRa 32 conectada ao módulo DHT22 . . . . .	39
Figura 5 – Arquitetura IdC baseada em três camadas: dispositivos, borda e nuvem	50
Figura 6 – Terceira revisão da placa Arduino Uno . . . . .	51
Figura 7 – Tecnologias sem fio de acordo com a sua cobertura geográfica . . . . .	52
Figura 8 – Camadas dos protocolos de rede LoRa e LoRaWAN . . . . .	53
Figura 9 – Camadas da arquitetura LoRaWAN e protocolos relacionados . . . . .	54
Figura 10 – Largura de banda e alcance do LoRaWAN comparado com outras redes sem fio . . . . .	55
Figura 11 – Mapa de calor do sinal de transmissões recebidas por um gateway a 136 metros de altura em Hilversum, Países Baixos . . . . .	56
Figura 12 – Simulação da área de cobertura de uma antena de 902 MHz a 126 metros de altura em Hilversum, Países Baixos . . . . .	57
Figura 13 – Geolocalização por trilateração usando <i>Received Signal Strength Indi- cation</i> (RSSI) . . . . .	58
Figura 14 – <i>Publisher</i> , <i>broker</i> e <i>subscriber</i> envolvidos em uma comunicação via MQTT . . . . .	59
Figura 15 – Funcionalidade de assinatura compartilhada do <i>Message Queue Teleme- try Transport</i> (MQTT) versão 5 . . . . .	60
Figura 16 – Diagrama de arquitetura do Sistema Baseado em Internet das Coisas para o Rastreo Aplicado à Logística da Cadeia de Frio (SBIdC-RALCF)	64
Figura 17 – As principais capacidades do SBIdC-RALCF . . . . .	65
Figura 18 – Diagrama de sequência da funcionalidade de frete . . . . .	67
Figura 19 – Disposição dos pinos da placa NodeMCU v3 e dos módulos DHT22 e SX1276 . . . . .	69

Figura 20 – Diagrama de sequência da função <i>setup</i> do código-fonte Arduino . . . . .	70
Figura 21 – Diagrama de sequência da função <i>loop</i> do código-fonte Arduino . . . . .	70
Figura 22 – Cobertura da KPN Things nos Países Baixos em ambientes fechados e ao ar livre . . . . .	72
Figura 23 – Dispositivos registrados no portal KPN Things . . . . .	72
Figura 24 – Dispositivos registrados no portal <i>The Things Network</i> (TTN) . . . . .	74
Figura 25 – Integrações da aplicação <i>web</i> do SBIdC-RALCF . . . . .	75
Figura 26 – Formulário de <i>login</i> gerado pela biblioteca Vaadin . . . . .	76
Figura 27 – Itens de menu de um usuário Administrador . . . . .	76
Figura 28 – Listagem de dispositivos registrados na KPN Things e TTN . . . . .	77
Figura 29 – Restrições da vacina “Novavax COVID-19” gerenciadas no SBIdC-RALCF	77
Figura 30 – Frete entre dois centros de vacinação nos Países Baixos . . . . .	78
Figura 31 – Medições de temperatura para um frete da vacina “Novavax COVID-19”	79
Figura 32 – Diagrama de Entidade-Relacionamento do projeto SBIdC-RALCF . . . . .	81
Figura 33 – Mensagens de requisição e aceite de adesão no portal KPN Things . . . . .	84
Figura 34 – Geolocalização recebida da KPN para rota entre Bussum e Soest . . . . .	85
Figura 35 – Aferições de temperatura recebidas de dispositivo conectado à KPN em rota entre Bussum e Soest . . . . .	86
Figura 36 – Gateways e caminho de transmissão mais longo mapeados usando TTN em uma rota de Soest a Bussum . . . . .	87
Figura 37 – Plano de teste JMeter do SBIdC-RALCF . . . . .	88
Figura 38 – Taxa de transferência de medições de dispositivos por servidor de rede	89
Figura 39 – Métrica de condição violada por servidor de rede . . . . .	90
Figura 40 – Diagrama de sequência da funcionalidade “Login” . . . . .	107
Figura 41 – Diagrama de sequência do acrônimo do inglês para <i>Create</i> (Criar), <i>Read</i> (Ler), <i>Update</i> (Atualizar) e <i>Delete</i> (Excluir) (CRUD) de Tipos de Medida	108
Figura 42 – Diagrama de sequência do CRUD de Dispositivos . . . . .	109
Figura 43 – Diagrama de sequência do CRUD de Produtos . . . . .	110
Figura 44 – Diagrama de sequência do CRUD de Entregadores . . . . .	111
Figura 45 – Diagrama de sequência do CRUD de Fretes . . . . .	112
Figura 46 – Placa AzDelivery NodeMCU V3 . . . . .	113
Figura 47 – Placa Heltec WiFi LoRa 32 V2 com antena compatível . . . . .	117
Figura 48 – Módulo DHT22 . . . . .	121
Figura 49 – Módulo SX1276 com antena soldada . . . . .	123
Figura 50 – Configuração das <i>Uniform Resource Locators</i> (URLs) das placas ESP8266 e ESP32 no Arduino <i>Integrated Development Environment</i> (IDE) . . . . .	126
Figura 51 – NodeMCU 1.0 (ESP-12E) listada no Arduino IDE . . . . .	126
Figura 52 – WiFi LoRa 32(V2) listada no Arduino IDE . . . . .	127

Figura 53 – Biblioteca <i>Digital Temperature And Humidity</i> (DHT) da Adafruit instalada no Arduino IDE . . . . .	127
Figura 55 – Caminho que contém os arquivos do Arduino IDE . . . . .	128
Figura 54 – Biblioteca “MCCI LoRaWAN LMIC Library” instalada no Arduino IDE	128
Figura 56 – Adicionar dispositivo no portal KPN Things . . . . .	132
Figura 57 – <i>DevEUI</i> , <i>AppEUI</i> e <i>AppKey</i> gerados pela KPN Things para o dispositivo	132
Figura 58 – Especificação do <i>broker</i> MQTT como destino para as mensagens dos dispositivos KPN . . . . .	133
Figura 59 – Formulário para inclusão de um dispositivo no portal TTN . . . . .	136
Figura 60 – Configuração do <i>broker</i> MQTT no “TTN Mapper” . . . . .	138



---

# Lista de tabelas

---

Tabela 1 – Mapeamento da placa NodeMCU aos módulos SX1276 e DHT22 . . .	68
Tabela 2 – Mapeamento da placa WiFi LoRa 32 ao módulo DHT22 . . . . .	68
Tabela 3 – Os algoritmos de geolocalização LoRa da KPN Things. . . . .	73
Tabela 4 – Desempenho dos algoritmos de geolocalização LoRa da KPN Things. .	73



---

# Lista de siglas

---

**3G** Terceira Geração da telefonia móvel

**4G** Quarta Geração da telefonia móvel

**5G** Quinta Geração da rede móvel

**API** *Application Programming Interface*

**ABP** *Activation By Personalisation*

**AES** *Advanced Encryption Standard*

**ADC** Conversor analógico-digital, em inglês *Analog-to-digital converter*

**BLE** *Bluetooth Low Energy*

**BYOD** *Bring Your Own Device*

**CRUD** acrônimo do inglês para *Create* (Criar), *Read* (Ler), *Update* (Atualizar) e *Delete* (Excluir)

**CSS** *Chirp Spread Spectrum*

**CSV** *comma-separated values*

**CAPES** Coordenação de Aperfeiçoamento de Pessoal de Nível Superior

**CUCPIC** Computação Ubíqua, Computação Pervasiva e Internet das Coisas

**CTC** *Controlled Temperature Chain*

**DSM** *Design Science Methodology*

**DHT** *Digital Temperature And Humidity*

**DER** Diagrama de Entidade-Relacionamento

**DT** *Design Theory*

**FUP** Política de Uso Justo, do inglês *Fair Use Policy*

**GPRS** *General Packet Radio Service*

**GPS** *Global Positioning System*

**GSM** *Global System for Mobile Communications*

**GPIO** pinos de entrada/saída de uso geral, do inglês *general-purpose input/output*

**HTTP** *Hypertext Transfer Protocol*

**HTTPS** *Hypertext Transfer Protocol Secure*

**IdC** Internet das Coisas

**IoT** *Internet of Things*

**ITNG** *International Conference on Information Technology: New Generations*

**IDE** *Integrated Development Environment*

**I2C** *Inter-Integrated Circuit*

**IA** Inteligência Artificial

**JDBC** *Java Database Connectivity*

**JSON** *JavaScript Object Notation*

**JST** *Japan Solderless Terminal*

**LoRaWAN** *Long Range Wide Area Network*

**LoRa** *Long Range*

**LMIC** *LoRaWAN-MAC-in-C*

**LED** *Light-Emitting Diode*

**LPWAN** rede de área ampla de baixa potência, do inglês *Low-power wide-area network*

**LTE-M** *Long-Term Evolution for Machines*

**LoS** linha de visão, do inglês *Line of Sight*

**LiPo** Polímero de Lítio, do inglês *Lithium Polymer*

**MQTT** *Message Queue Telemetry Transport*

**NFC** *Near Field Communication*

**NB-IoT** *Narrowband Internet of Things*

**OMS** Organização Mundial da Saúde

**OTAA** *Over-The-Air-Activation*

**OLED** diodo orgânico emissor de luz, do inglês *organic light-emitting diode*

**OSI** *Open Systems Interconnection*

**ORM** Operador de Rede Móvel

**PPP** Parceria Público-Privada

**PWM** Modulação por largura de pulso, do inglês *Pulse-Width Modulation*

**PPT** Problema de Pesquisa Técnica

**QoS** Qualidade de Serviço, do inglês *Quality of Service*

**QR** *Quick-Response*

**QC** Questão de Conhecimento

**RFID** *Radio-Frequency Identification*

**RSL** Revisão Sistemática da Literatura

**RBAC** Controle de acesso baseado em função, do inglês *Role-based access control*

**RSSF** Rede de sensores sem fio

**REST** *Representational State Transfer*

**RSSI** *Received Signal Strength Indication*

**RoI** retorno sobre o investimento, ou *Return on Investment*

**RSL** Revisão Sistemática da Literatura

**SBIIdC-RALCF** Sistema Baseado em Internet das Coisas para o Rastreamento Aplicado à Logística da Cadeia de Frio

**SD** *Secure Digital*

**SenML** *Sensor Measurement List*

**SF** Fatores de Espalhamento, do inglês *Spreading Factor*

**SPI** *Serial Peripheral Interface*

**SMA** *SubMiniature version A*

**SCM** Gestão da cadeia de suprimentos de negócios, em inglês *Supply chain management*

**TTN** *The Things Network*

**TTS** *The Things Stack*

**TDoA** *Time Distance of Arrival*

**TAR** *Technical action research*

**TCP** *Transmission Control Protocol*

**TLS** *Transport Layer Security*

**TRP** *Technical Research Problem*

**TP** *Teoria de Projeto*

**UART** *Transmissor/Receptor Universal Síncrono e Assíncrono, do inglês *Universal Synchronous Asynchronous Receiver Transmitter**

**URL** *Uniform Resource Locator*

**USB** *Universal Serial Bus*

**UDP** *User Datagram Protocol*

**URI** *Uniform Resource Identifier*

**WSN** *Wireless Sensor Networks*

**WWAN** *Wireless Wide Area Networks*

**WLAN** *Wireless Local-Area Network*

**WPAN** *Wireless Personal Area Network*

**XML** *Extensible Markup Language*



---

# Sumário

---

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>31</b>
1.1	Contexto . . . . .	31
1.2	Motivações e Objetivo . . . . .	33
1.3	Organização da Dissertação . . . . .	33
<b>2</b>	<b>MÉTODOS E MATERIAIS . . . . .</b>	<b>35</b>
2.1	Metodologia . . . . .	35
2.2	Questões e Problema de Pesquisa . . . . .	36
2.3	Ciclo de Projeto e Ferramental . . . . .	37
2.3.1	Investigação . . . . .	38
2.3.2	<i>Design</i> . . . . .	39
2.3.3	Validação . . . . .	40
2.4	Considerações Finais . . . . .	41
<b>3</b>	<b>TRABALHOS RELACIONADOS . . . . .</b>	<b>43</b>
3.1	Domínios de Aplicação da Internet das Coisas . . . . .	44
3.2	Internet das Coisas na Cadeia de Frio . . . . .	45
3.3	LoRaWAN na Logística e Cadeia de Frio . . . . .	47
3.4	Diferencial . . . . .	48
3.5	Considerações Finais . . . . .	48
<b>4</b>	<b>TECNOLOGIAS DA INTERNET DAS COISAS . . . . .</b>	<b>49</b>
4.1	Internet das Coisas . . . . .	49
4.2	Arduino e placas compatíveis . . . . .	51
4.3	Redes sem Fio . . . . .	52
4.4	LoRa e LoRaWAN . . . . .	53
4.4.1	Arquitetura LoRaWAN . . . . .	54

4.4.2	Largura de banda e alcance . . . . .	55
4.4.3	Linha de visão . . . . .	56
4.4.4	Geolocalização . . . . .	57
4.5	MQTT . . . . .	58
4.6	HTTP(S) . . . . .	60
4.7	REST . . . . .	61
4.8	Considerações Finais . . . . .	61
5	SBIDC-RALCF . . . . .	63
5.1	Arquitetura do Sistema . . . . .	64
5.2	Capacidades . . . . .	65
5.3	Dispositivos . . . . .	68
5.3.1	Mapeamento de pinos . . . . .	68
5.3.2	<i>Firmware</i> . . . . .	69
5.4	Servidores de Rede . . . . .	71
5.4.1	KPN Things . . . . .	71
5.4.2	<i>The Things Network</i> . . . . .	74
5.5	Aplicação <b>Web</b> . . . . .	75
5.5.1	Interface do Usuário . . . . .	76
5.5.2	Diagrama de Entidade-Relacionamento . . . . .	79
5.6	Considerações Finais . . . . .	82
6	VALIDAÇÃO E ANÁLISE DOS RESULTADOS . . . . .	83
6.1	Ativação do dispositivo . . . . .	83
6.2	Estudo de Caso . . . . .	85
6.3	Experimento de escala . . . . .	88
6.4	Considerações Finais . . . . .	90
7	DISCUSSÃO . . . . .	91
7.1	Análise das Questões de Pesquisa . . . . .	91
7.2	Limitações . . . . .	93
7.3	Trabalhos Futuros . . . . .	94
7.4	Considerações Finais . . . . .	96
Conclusão . . . . .		97
REFERÊNCIAS . . . . .		99

**APÊNDICES** **105**

APÊNDICE A	–	DIAGRAMAS DE SEQUÊNCIA . . . . .	107
------------	---	----------------------------------	-----

A.1	Login do Usuário . . . . .	107
A.2	CRUD de Tipos de Medida . . . . .	108
A.3	CRUD de Dispositivos . . . . .	109
A.4	CRUD de Produtos . . . . .	110
A.5	CRUD de Entregadores . . . . .	111
A.6	CRUD de Fretes . . . . .	112
APÊNDICE B	– NODEMCU V3 . . . . .	113
APÊNDICE C	– WIFI LORA 32 V2 . . . . .	117
APÊNDICE D	– MÓDULO DHT22 . . . . .	121
APÊNDICE E	– MÓDULO LORA SX1276 . . . . .	123
APÊNDICE F	– ARDUINO IDE . . . . .	125
F.1	Habilitar NodeMCU V3 . . . . .	125
F.2	Habilitar WiFi LoRa 32 V2 . . . . .	126
F.3	Instalar biblioteca DHT . . . . .	127
F.4	Instalar biblioteca LMIC . . . . .	127
F.5	Configuração dos parâmetros da LMIC . . . . .	128
APÊNDICE G	– CONFIGURAÇÃO DO SERVIDOR DE REDE KPN THINGS . . . . .	131
G.1	Registro de dispositivo . . . . .	131
G.2	Configuração do <i>broker</i> MQTT . . . . .	133
G.3	Fluxo de mensagens do <i>broker</i> MQTT . . . . .	133
APÊNDICE H	– CONFIGURAÇÃO DO SERVIDOR DE REDE TTN . . . . .	135
H.1	Criação de aplicação . . . . .	135
H.2	Registro de dispositivo . . . . .	136
H.3	Configuração do TTN Mapper . . . . .	137
APÊNDICE I	– EXECUÇÃO DA APLICAÇÃO . . . . .	139
I.1	Pré-requisitos de software . . . . .	139
I.2	Variáveis de ambiente . . . . .	139
I.3	Comando de execução . . . . .	140
I.4	Execução do teste de carga . . . . .	141

**ANEXOS**

**143**

<b>ANEXO A</b>	<b>–</b>	<b>ARTIGO REGULAR APRESENTADO NA ITNG 2023 . . . . .</b>	<b>145</b>
<b>ANEXO B</b>	<b>–</b>	<b>ARTIGO REGULAR ACEITO PELA ICEIS 2025 .</b>	<b>153</b>

---

# Capítulo 1

## Introdução

---

### 1.1 Contexto

A COVID-19, doença respiratória causada pelo vírus SARS-CoV-2, foi caracterizada como pandemia pela Organização Mundial da Saúde (OMS) em 11 de março de 2020 após o aumento de número de casos em 13 vezes e o triplo de países afetados nas duas semanas anteriores (ORGANIZATION et al., 2020). Desde o surgimento da doença, governos e órgãos sem fins lucrativos somaram esforços essenciais no auxílio às farmacêuticas, resultando no desenvolvimento de vacinas eficazes no combate à doença em tempo recorde (WOUTERS et al., 2021). Segundo levantamento das vacinas candidatas líderes até fevereiro de 2021, realizado por Wouters et al. (2021), a maioria necessita de armazenamento entre 2°C e 8°C, enquanto vacinas de alta eficácia (ZHENG et al., 2022) como Moderna e Pfizer-BioNTech precisam de temperaturas ainda mais baixas: -20°C e -70°C, respectivamente. Esta última, por exemplo, deve ser administrada em até 5 dias após violada sua condição de conservação. Isto posto, o controle do ambiente torna-se vital para a sua utilidade e, conseqüentemente, a contenção da doença, sendo este um desafio em especial para países subdesenvolvidos e emergentes (YAN et al., 2021; WOUTERS et al., 2021).

A gestão da cadeia de frio, expressão que referencia a produção, o armazenamento e a distribuição de produtos sensíveis a variações de condições de temperatura e umidade, é explorada pela OMS desde 2007 com o intuito de facilitar a distribuição de vacinas fora do intervalo de temperatura convencional - entre 2°C e 8°C. Esse empreendimento culminou no desenvolvimento de uma abordagem inovadora, chamada *Controlled Temperature Chain* (CTC), que representa um dos pilares do Plano de Ação Global para Vacinas de 2010 a 2020 (ORGANIZATION et al., 2017). Uma vacina com a etiqueta CTC pode

suportar até 40°C por até 3 dias antes de sua aplicação, fato este que viabiliza a imunização de populações em áreas remotas, além de baratear o gerenciamento ao dispensar refrigeração e facilitar o transporte com a redução do peso do recipiente.

Outros suprimentos vitais também dependem da cadeia de frio para controle da sua efetividade e segurança, como os soros antiofídicos para venenos de cobras comumente aplicados em áreas remotas da região Amazônica, onde a probabilidade de desperdício é maior quando comparada às regiões urbanas, por uso de meios de transporte distintos e suscetibilidade a falhas de equipamentos de refrigeração (FAN; MONTEIRO, 2018). A penetração da Internet nas áreas rurais brasileiras é de cerca de 34%, em comparação com 65% das regiões urbanas, principalmente devido ao alto investimento por habitante coberto e à incerteza de receita do Operador de Rede Móvel (ORMs) (CAVALCANTE et al., 2021). Enfrentar esses desafios requer soluções inovadoras que operem de forma eficiente sob restrições de energia e forneçam comunicação confiável em áreas com infraestrutura de rede limitada.

Tornar a cadeia de frio dispensável no gerenciamento desses complexos biológicos seria a solução mais efetiva para lidar com as dificuldades impostas, conforme sugerido por (YU et al., 2021), no entanto, isto ainda significa um desafio enorme de pesquisa farmacêutica, segundo os próprios autores. No ramo alimentício, analogamente, estima-se que 30% dos alimentos produzidos para consumo humano são desperdiçados por falhas no gerenciamento da cadeia de distribuição, em grande parte devido a interrupções causadas pela má refrigeração (BADIA-MELIS et al., 2018).

Diversas tecnologias são aplicadas com o intuito de monitorar alterações nas condições na cadeia de frio, destacando-se a identificação por radiofrequência e a rede de sensores sem fio, do inglês *Radio-Frequency Identification* (RFID) e *Wireless Sensor Networks* (WSN), respectivamente (BADIA-MELIS et al., 2018). Enquanto a primeira possibilita a identificação de objetos por proximidade, por exemplo por meio de leituras de etiquetas, a segunda abrange as redes sem fio que suportam a comunicação de dados coletados por sensores. Ambas remetem à IdC, ou *Internet of Things* (IoT) em inglês, termo cunhado em 1999 por Kevin Ashton, onde objetos - ou coisas - comuns ao nosso dia a dia são dotados de tecnologias embarcadas capazes de identificar, observar, intervir e compreender o mundo, comunicando-se por meio da Internet (ASHTON et al., 2009).

Um protocolo de rede de área ampla de baixa potência, do inglês *Low-power wide-area network* (LPWAN) como LoRaWAN emerge como um candidato ideal para aplicações de IdC em locais com acesso limitado a energia e conectividade (CENTENARO et al., 2016). Sua arquitetura de rede robusta operando em uma banda livre de licença permite implantação escalável, acomodando milhares de dispositivos conectados e facilitando o monitoramento ambiental abrangente em logística de cadeia de frio (VANGELISTA; ZANELLA; ZORZI, 2015). Além disso, mostra-se como uma alternativa custo-efetiva, pois possibilita a geolocalização de dispositivos sem a necessidade de módulos *Global Position-*

*ing System* (GPS)(FARGAS; PETERSEN, 2017) e o custo diário total para um número crescente de dispositivos é menor do que o das alternativas de redes celulares, devido aos seus custos de assinatura e infraestrutura privada (FRANGOUDIS; TSIGKANOS; DUSTDAR, 2021).

As características do LoRaWAN são amplamente exploradas na literatura, incluindo alguns estudos no contexto de distribuição de vacinas e outros produtos biológicos da cadeia de frio. No entanto, há oportunidade do emprego de atributos do protocolo, como redes híbridas e geolocalização, especialmente para enfrentar os desafios da cadeia de frio impostos em países emergentes e em desenvolvimento, onde a infraestrutura de telecomunicações é limitada, com falhas no armazenamento e transporte que podem comprometer a imunização da população.

## 1.2 Motivações e Objetivo

Esta dissertação de Mestrado teve como motivação inicial a temática da distribuição e conservação de vacinas dada a relevância do tema durante a pandemia da COVID-19, pano de fundo no início deste trabalho. As revisões exploratórias e sistemáticas realizadas permitiram, de forma mais abrangente, identificar os desafios enfrentados na logística da cadeia de frio e como as tecnologias da IdC podem colaborar para enfrentar ou mitigar tais adversidades.

O objetivo principal desta pesquisa foi projetar, desenvolver e validar o Sistema Baseado em Internet das Coisas para o Rastreo Aplicado à Logística da Cadeia de Frio (SBIdC-RALCF), um sistema com protótipos e aplicação *web* LoRaWAN que melhora a comunicação de rede em áreas remotas ao suportar múltiplos servidores de rede, possibilita o rastreo dos produtos independentemente de módulos GPS e permite detectar e notificar violações de condições cruciais, mesmo sem conectividade com a Internet. Desta forma, espera-se mitigar o desperdício de produtos perecíveis vitais no contexto da logística da cadeia de frio, além de apoiar governos e tomadores de decisões a estabelecer Parceria Público-Privada (PPP) de forma custo-efetiva, uma vez que esta abordagem de rede híbrida, com múltiplas redes LoRaWAN, pode reaproveitar a conectividade suportada por empresas de telecomunicação enquanto um novo investimento público pode ser empregado em infraestrutura de rede afastada dos centros urbanos.

## 1.3 Organização da Dissertação

A sequência desta dissertação de mestrado é estruturada da seguinte forma: o Capítulo 2 expõe a metodologia e o ferramental utilizados no desenho e implementação do sistema proposto; o Capítulo 3 discute os mais relevantes trabalhos de aplicações IdC e LoRaWAN no contexto de logística e da cadeia de frio; o Capítulo 4 fornece uma visão geral do

estado da arte das principais tecnologias empregadas neste projeto; o Capítulo 5 apresenta o projeto e desenvolvimento do SBIdC-RALCF, incluindo sua arquitetura, capacidades, protótipos, servidores de rede e a aplicação web; o Capítulo 6 discorre sobre os mecanismos e experimentos utilizados na validação do sistema de acordo com a metodologia adotada; o Capítulo 7 trata dos desafios enfrentados, as limitações e potenciais trabalhos futuros sobre a proposta, seguido da Conclusão com as considerações finais deste trabalho.

---

## Capítulo 2

# Métodos e Materiais

---

Para atingir o objetivo de projetar um sistema que aprimore o rastreamento de produtos da cadeia de frio e validar as hipóteses de forma experimental, este trabalho utilizou uma metodologia de pesquisa que formaliza o projeto e desenvolvimento de uma solução para um problema em um contexto definido, além de empregar ferramentas e tecnologias de código-fonte aberto que viabilizam a reprodutibilidade dos resultados obtidos em cada etapa.

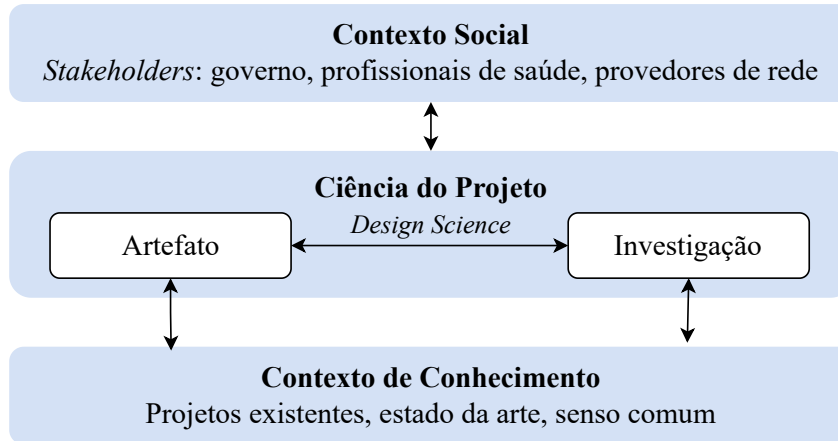
Este capítulo apresenta a metodologia seguida, o problema e questões de pesquisa, e o ferramental empregado no desenvolvimento desta dissertação. Isto inclui os métodos de pesquisa que norteiam a investigação da problemática da logística da cadeia de frio, o levantamento de tecnologias de IdC que podem ser empregadas neste contexto e a ciência por trás do projeto e validação do SBIdC-RALCF.

### 2.1 Metodologia

Este projeto de mestrado foi desenvolvido seguindo a metodologia da “Ciência do Projeto”, ou *Design Science Methodology* (DSM) (WIERINGA, 2014). O *framework* utilizado é ilustrado na Figura 1 nos moldes sugeridos por Wieringa (2014, 1.3), constituído pelo Contexto Social, o Contexto do Conhecimento e a Ciência do Projeto. O primeiro define as partes interessadas ou *stakeholders*, que fornecem o orçamento e estabelecem metas a serem alcançadas pelo projeto. O Contexto do Conhecimento consiste em teorias existentes da ciência e engenharia, lições aprendidas com a experiência de pesquisadores em projetos anteriores da Ciência do Projeto e senso comum. A Ciência do Projeto ou *Design Science*, centrada no diagrama, itera sobre duas atividades principais: projetar um artefato que melhora algo para as partes envolvidas e investigar empiricamente seus

efeitos em um contexto de problema. Ela produz um novo projeto ou *design* de resolução de problemas e responde a perguntas de conhecimento.

Figura 1 – Um *framework* para a Ciência do Projeto



Fonte: Adaptado de (WIERINGA, 2014)

No caso deste estudo, o contexto é o rastreamento de produtos da cadeia de frio em áreas remotas com ausência ou intermitência de infraestrutura de rede, enquanto o artefato é um sistema LoRaWAN com suporte a múltiplos servidores de rede que visa remover o problema. As partes interessadas são o governo, profissionais de saúde e provedores de rede. O primeiro atua como patrocinador do projeto de pesquisa e requer que ele seja relevante e útil para a sociedade no contexto da logística da cadeia de frio. Os profissionais de saúde são atores que interagem diretamente com os componentes do artefato construído, enquanto os últimos, os provedores de rede, desempenham um papel essencial como detentores da infraestrutura de rede que deve ser usufruída e aprimorada.

## 2.2 Questões e Problema de Pesquisa

De acordo com Wieringa (2014, 2.2), um problema de projeto, ou *design problem*, é um instrumento para alcançar um objetivo de pesquisa ao projetar um artefato alinhado com os requisitos dos *stakeholders* que propicie mudanças no contexto investigado. O autor define como Problema de Pesquisa Técnica (PPT) ou *Technical Research Problem* (TRP) o termo para representar este problema como uma questão de pesquisa. Adicionalmente, a definição e resolução de um conjunto de Questão de Conhecimento (QC) (WIERINGA, 2014, 2.3) embasam a formulação do PPT por meio do entendimento do contexto de conhecimento através de dados, sem causar uma melhoria direta no contexto de estudo.

As QCs a seguir foram levantadas e respondidas encadeadamente ao longo deste projeto de mestrado para que o PPT pudesse ser formulado.

**QC1:** *Quais são os domínios de aplicação mais relevantes da IdC?*

**QC2:** *Quais são as principais tecnologias de IdC utilizadas na logística da cadeia de frio?*

**QC3:** *Como estas tecnologias de IdC são aplicadas em áreas remotas?*

**QC4:** *Como aplicações LoRaWAN são utilizadas na logística e cadeia de frio?*

A partir da obtenção de respostas das QCs durante o ciclo de projeto, apresentado na seção a seguir, a questão de pesquisa no formato de PPT deste projeto de mestrado foi definida como:

*Como projetar um sistema baseado em IdC*

*que melhore a comunicação de rede em áreas remotas, possibilite o rastreamento dos produtos com baixo custo e permita detectar e notificar violações de condições na ausência de conectividade com a Internet*

*para apoiar governos e tomadores de decisões a estabelecer Parcerias Público-Privadas (PPP) custo-efetivas enquanto mitiga o desperdício de produtos perecíveis*

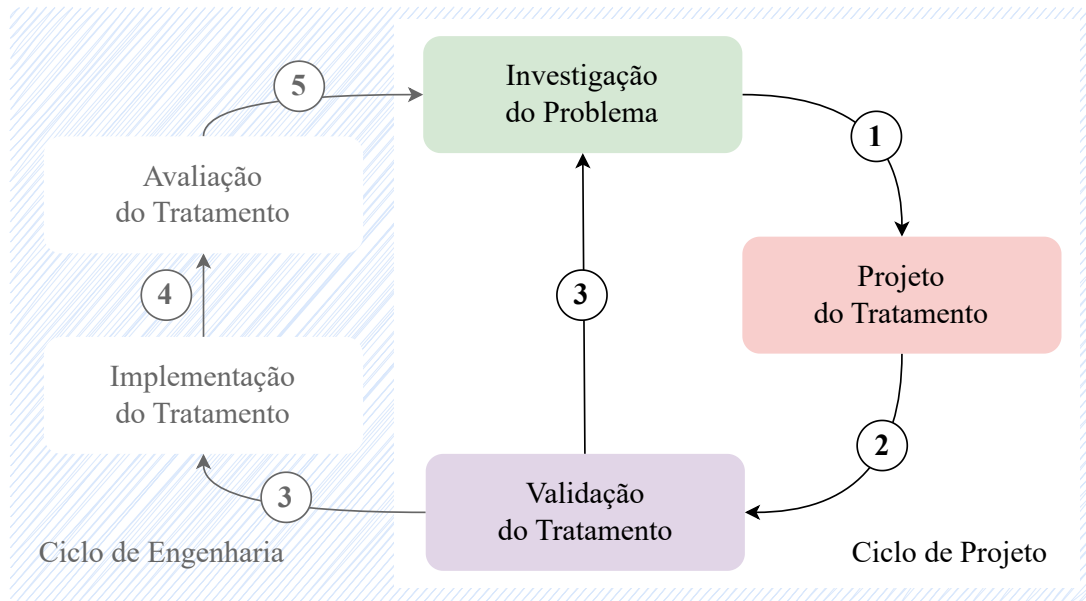
*na logística da cadeia de frio?*

A Teoria de Projeto (TP) ou *Design Theory* (DT) (WIERINGA, 2014, 5.2) deste estudo para responder o PPT é que ao construir uma aplicação LoRaWAN que suporte múltiplos provedores de rede, a confiabilidade e redundância da comunicação de longo alcance neste contexto será aperfeiçoada, permitindo detectar e notificar violações das condições ideais dos produtos perecíveis, além de habilitar o rastreamento destas mercadorias com o recurso de geolocalização de baixo custo do protocolo.

## 2.3 Ciclo de Projeto e Ferramental

A tarefa de *design* do artefato da DSM é realizada dentro do Ciclo de Projeto ou *Design Cycle* Wieringa (2014, 3.1). O artefato apresentado nesse projeto de mestrado foi concebido seguindo este ciclo que consiste em três tarefas: 1. Investigação do Problema: descobrir os objetivos das partes interessadas e compreender os desafios enfrentados no rastreamento da cadeia de frio em regiões remotas; 2. Projeto do Tratamento: projetar o aplicativo LoRaWAN para melhorar o rastreamento da cadeia de frio em áreas remotas com cobertura de rede intermitente; e 3. Validação do Tratamento: avaliar os efeitos causados pela aplicação proposta no contexto do rastreamento da cadeia de frio. Uma validação pode desencadear uma nova iteração da investigação do problema, conforme mostrado na Figura 2.

Figura 2 – As tarefas do Ciclo de Projeto dentro de um Ciclo de Engenharia



Fonte: Elaborado pelo autor

De acordo com Wieringa (2014, 3.1), o Ciclo de Projeto faz parte de um fluxo mais abrangente chamado Ciclo de Engenharia, que também abrange as tarefas adicionais de Implementação do Tratamento e Avaliação de Tratamento para validar a aplicação do artefato em um cenário do mundo real e usá-lo e avaliá-lo com as partes interessadas. Os protótipos e aplicação aqui apresentados foram validados apenas em ambientes experimentais, portanto essas tarefas estão fora do escopo deste projeto de pesquisa.

Ainda de acordo com a DSM, o termo “solução” deve ser evitado ao se descrever o projeto, pois o artefato pode solucionar apenas parcialmente determinado problema, enquanto “tratamento” sugere uma contribuição ou melhoria para um problema do mundo real, como um medicamento aplicado no corpo humano contribuindo para a cura.

### 2.3.1 Investigação

Na tarefa de Investigação do Problema, foi conduzida uma Revisão Sistemática da Literatura (RSL) (GARCIA; SOUZA, 2023) de aplicações e tecnologias de IdC para rastreamento da cadeia de frio com foco na distribuição de vacinas, detalhada no Anexo A. Nesta pesquisa, ficou evidente a falta de infraestrutura de rede que suporte adequadamente a distribuição de produtos da cadeia de frio, especialmente devido ao baixo retorno sobre o investimento, ou *Return on Investment* (RoI) para empresas de telecomunicações, considerando a densidade populacional em áreas remotas. Como contrapartida, governos costumam estabelecer metas de cobertura também em áreas de baixo retorno.

Outro aprendizado importante desta RSL é que tecnologias de LPWAN oferecem recursos apropriados em contextos com recursos limitados. Dentre elas, LoRaWAN destacou-se

por apresentar características como baixo custo, baixo consumo de energia e operação em bandas de frequência livres de licença. Devido a isso, uma revisão abrangente da literatura dessa tecnologia e suas aplicações foi conduzida para identificar seu papel potencial como parte do artefato. Embora muitas aplicações empreguem essa tecnologia bem conhecida, elas não exploram alguns aspectos, especialmente a capacidade de rede híbrida em um contexto como a cadeia de frio.

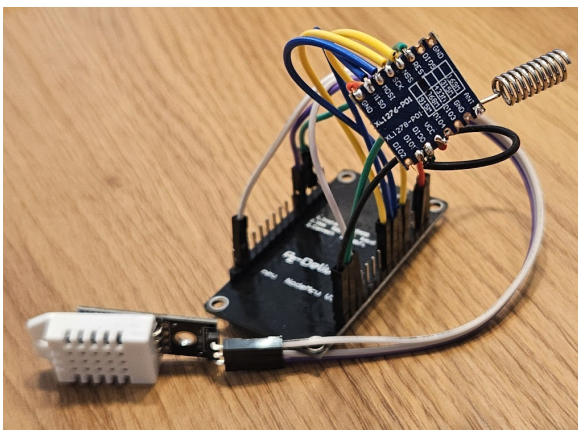
Os trabalhos relacionados e principais tecnologias aplicadas no tratamento aqui proposto são apresentados no capítulo 3.

### 2.3.2 Design

Na tarefa de Projeto do Tratamento, detalhada no capítulo 5, uma aplicação LoRaWAN com suporte a múltiplos servidores de rede foi projetada, incluindo protótipos de hardware, a camada de comunicação e uma aplicação *web*. O código-fonte resultante deste *design* é publicamente acessível através do repositório Git<sup>1</sup> em URL <<https://github.com/alexfabgarcia/cold-chain-tracking>>.

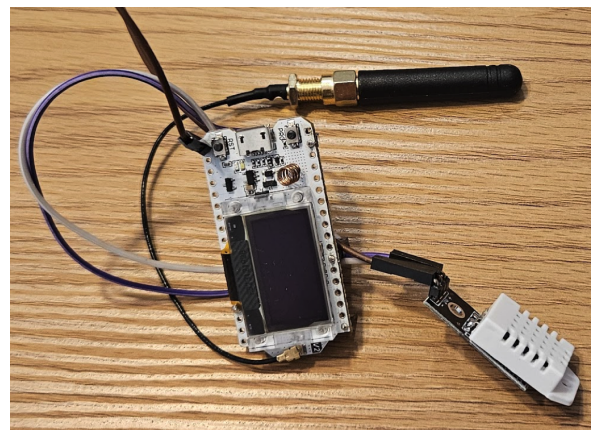
Dois protótipos representam os dispositivos LoRaWAN deste estudo, exibidos nas Figuras 3 e 4. O primeiro é uma placa de desenvolvimento NodeMCU V3 conectada a um módulo de sensor DHT22 e a um transceptor SX1276 com uma antena soldada. O segundo é uma placa WiFi LoRa 32 V2 conectada somente a um módulo de sensor DHT22, sendo a sua antena removível. Os Apêndices B, C, D e E detalham os pinos e funções destas placas, sensor e transceptor. O mapeamento necessário para a sua reprodução é esmiuçado na seção 5.3, para ambas as placas.

Figura 3 – NodeMCU V3 conectada aos módulos DHT22 e SX1276



Fonte: Produzido pelo autor

Figura 4 – WiFi LoRa 32 conectada ao módulo DHT22



Fonte: Produzido pelo autor

O Arduino IDE foi a ferramenta empregada para embarcar o código-fonte nos protótipos. Para que isso seja possível, é requerido que este IDE gerencie os modelos das placas

<sup>1</sup> <<https://git-scm.com>>

e as bibliotecas que abstraem a comunicação com o sensor e transceptor utilizados. Além disso, há configurações específicas do protocolo LoRaWAN como a região e credenciais de ativação que devem ser parametrizadas. O Apêndice F concentra todas as configurações requeridas para que os dispositivos se tornem operáveis neste IDE.

A camada de comunicação do SBIdC-RALCF prevê múltiplos servidores de rede LoRaWAN, que permitem o registro de dispositivos próprios e provêm integrações para envio e recebimento de mensagens. Os Apêndices G e H descrevem como KPN Things e TTN, respectivamente, são configurados e utilizados neste Ciclo de Projeto. A seção 5.5 detalha como o SBIdC-RALCF pode ser estendido para embarcar outro servidor de rede que oferece capacidades similares.

O SBIdC-RALCF contempla uma aplicação *web* desenvolvida em Java versão 21<sup>2</sup> multiplataforma, que se integra com os servidores de rede, *Application Programming Interface* (API) de geolocalização para resolução de coordenadas e com o seu banco de dados. A seção 5.5 detalha como os parâmetros dos servidores de rede e credenciais para acesso às APIs devem ser configurados na aplicação. Para a sua inicialização, além do Java na versão condizente, é necessária a instalação do software Docker<sup>3</sup> pois ele auxilia na inicialização do banco de dados e componentes de monitoramento da aplicação. O arquivo `README.md` do código-fonte também contém o detalhamento destas variáveis e como a aplicação pode ser inicializada.

### 2.3.3 Validação

A Validação do Tratamento compreende um mecanismo de estudo de caso e um experimento de escalabilidade (WIERINGA, 2014, 7.4.2; 7.5) como métodos de validação. Conforme detalhado no capítulo 5, primeiro o SBIdC-RALCF foi testado em campo com diferentes rotas e modos de transporte, validando o cumprimento dos requisitos definidos. Posteriormente, ele foi submetido a maior escalabilidade com condições mais realistas por meio de testes de carga para avaliar a sua robustez.

A reprodução do estudo de caso é possível desde que todos os componentes do sistema estejam operantes, conforme detalhado na seção anterior. O único requisito adicional é a instalação e configuração do aplicativo “TTN Mapper” em um aparelho celular Android ou iPhone com a capacidade de GPS, conforme detalhado no Apêndice H.3. Isto é necessário para registrar a distância do protótipo e do gateway da TTN que recebe a mensagem, pois este servidor não suporta geolocalização por padrão.

O experimento de escalabilidade também pode ser reproduzido com o sistema em operação. A seção 6.3 detalha como a massa de teste deve ser preparada antes da execução do teste de carga com a ferramenta JMeter<sup>4</sup>. O comando

---

<sup>2</sup> <<https://jdk.java.net>>

<sup>3</sup> <<https://www.docker.com>>

<sup>4</sup> <<https://jmeter.apache.org/>>

```
jmeter -t jmeter/cct-load-test.jmx
```

é suficiente para iniciar o plano de teste (em relação à raiz do código-fonte), considerando a execução da aplicação em um ambiente local conforme descrito no Apêndice I.

## 2.4 Considerações Finais

Os métodos e materiais utilizados para o projeto e desenvolvimento do SBIdC-RALCF foram apresentados neste capítulo, destacando a adoção da DSM para a definição do problema de pesquisa, além do ferramental de código-fonte aberto empregado em sua construção. Métodos de pesquisa como RSLs e revisões bibliográficas foram empregados para a investigação do problema e obtenção de respostas para as questões de conhecimento. O entendimento do contexto dos desafios enfrentados na logística da cadeia de frio norteou a execução das tarefas do Ciclo de Projeto, que permitiram projetar uma solução baseada na tecnologia LoRaWAN e validar seu impacto com um mecanismo de estudo de caso e um experimento de escala.



---

## Capítulo 3

# Trabalhos Relacionados

---

No início desse projeto de mestrado, uma revisão sistemática sobre estudos secundários foi realizada para identificar os principais domínios de aplicação da IdC. Duas das áreas mais relevantes, logística e saúde, apresentavam uma intersecção com a principal motivação deste estudo no contexto da pandemia: a distribuição das vacinas. A ferramenta Parsifal<sup>1</sup> foi utilizada nesta e nas subsequentes revisões para planejar, conduzir e consolidar os resultados.

A cadeia de frio ou *cold chain*, que abrange a cadeia de suprimentos de produtos perecíveis como vacinas e alimentos, foi um termo chave encontrado em pesquisas exploratórias. Este termo guiou uma nova revisão nessa temática para identificar quais tecnologias da IdC eram aplicadas neste contexto e como os manuscritos lidavam com um dos desafios mais citados: a intermitência de infraestrutura de rede e energia em áreas remotas. As respostas obtidas sugeriram que protocolos LPWAN, em especial o LoRaWAN, oferecem capacidades que poderiam ser exploradas nestas condições. Assim, uma RSL final foi executada para identificar as aplicações que utilizam este protocolo e as lacunas dos estudos encontrados, que contribuiriam para a elaboração do PPT deste projeto.

Este capítulo aborda os trabalhos mais relevantes desta série de revisões bibliográficas abrangentes e sistemáticas realizadas para responder às QCs. Após explorar, compreender e resumir os principais resultados, este estudo foi encaminhado à problemática do rastreamento de produtos da logística da cadeia de frio.

---

<sup>1</sup> <<https://parsif.al>>

### 3.1 Domínios de Aplicação da Internet das Coisas

A primeira RSL serviu como um aprendizado para a organização e condução de pesquisas de forma sistemática para responder as QCs. As fontes utilizadas, “ACM Digital Library”<sup>2</sup>, “IEEE Digital Library”<sup>3</sup>, “Science@Direct”<sup>4</sup> e “Scopus”<sup>5</sup>, foram submetidas à seguinte *string* de pesquisa

(*“internet of things” OR “IoT” OR “pervasive computing” OR “ubiquitous computing”*) AND (*“survey”*) AND (*“domain” OR “application”*)

com o objetivo de prospectar estudos secundários da área de Computação Ubíqua, Computação Pervasiva e Internet das Coisas (CUCPIC) referentes aos domínios de aplicação. Dentre os estudos selecionados na condução da RSL, quatro atenderam ao critério de seleção no âmbito dos principais domínios de aplicação.

Choi et al. (2021) realizaram uma análise bibliométrica, considerada uma técnica inovadora capaz de ressaltar tendências atuais e melhorias de um domínio específico de conhecimento, para destacar os principais domínios considerando o número de produções científicas, incluindo publicações, citações, dentre outros aspectos. A partir do levantamento de pesquisas bibliométricas anteriores relacionadas à IdC, destacam-se os domínios de Saúde ou *Smart Health*, Cidades Inteligentes e Sustentáveis ou *Smart and Sustainable City*, Energia ou *Smart Grid*, Indústria 4.0 ou *Industry 4.0*, Identidade ou *Authentication/Access Control*, Logística ou Gestão da cadeia de suprimentos de negócios, em inglês *Supply chain management* (SCM), *Network Science* (redes complexas de telecomunicações, computadores, biológicas, sociais, dentre outras), *Edge Computing* (camada de computação próxima aos dispositivos finais que intermedia o contato com a nuvem) e Segurança Alimentar. Ressalta-se ainda neste mesmo estudo o crescimento de interesse no conceito de Automação Residencial ou *Smart Home*, sobreposto em vários destes domínios, de 18,43% em 2015 a 43,70% em 2019.

Já em (DING et al., 2020), além do levantamento das tecnologias recentes da área, propõe-se uma classificação adicional das aplicações IdC considerando os tipos de usuários finais das aplicações e requisitos específicos: taxa de dados, latência, cobertura, energia, confiabilidade e mobilidade. O tipo de usuário final pode ser “orientado à máquina”, no qual há interação majoritariamente entre dispositivos e sensores, ou “orientado a humano”, com maior intervenção humana. Aplicações orientadas a humanos usualmente envolvem interface para visualização e/ou aceitam interações baseadas em linguagem natural, como comando de voz, além de que usualmente apresentam altas taxas de dados. Por outro lado,

---

<sup>2</sup> <<http://portal.acm.org>>

<sup>3</sup> <<http://ieeexplore.ieee.org>>

<sup>4</sup> <<http://www.sciencedirect.com>>

<sup>5</sup> <<http://www.scopus.com>>

aplicações orientadas às máquinas envolvem pouco envolvimento humano, apresentando maior concentração e baixas taxas de dados em sua maioria.

Kassab e Darabkh (2020), por sua vez, apresenta um *survey* com o intuito de cobrir lacunas de trabalhos relacionados recentes da área de IdC, dando cobertura ampla a paradigmas, arquiteturas, protocolos, middlewares, ferramentas de simulação, aplicações, dentre outros aspectos. Com o extenso levantamento bibliográfico realizado, os domínios de aplicações recentes também são sumarizados, destacando-se Cidades Inteligentes, Energia, Saúde, Agricultura, Automação Residencial e Indústria 4.0. Nele, tais domínios são detalhados, como é o caso de Cidades Inteligentes, no qual temos subdomínios como o gerenciamento de desperdício, monitoramento de ruído e da qualidade do ar, controle de tráfego, consumo energético, estacionamentos e semáforos inteligentes, dentre outros.

Por fim, Asghari, Rahmani e Javadi (2019) apresenta uma RSL focada no levantamento de estudos primários da área de IdC entre os anos de 2011 e 2018. Coincidentemente, umas das questões de pesquisa a serem respondidas nesta RSL é “Quais domínios são categorizados em aplicações IoT?”. Por meio de 72 artigos finais coletados é criada uma taxonomia de aplicações IdC a partir dos temas tratados em cada estudo, onde destacam-se os domínios de Saúde, Meio Ambiente, Cidades Inteligentes, Comercial/Negócios e Indústria. Na conclusão deste último estudo, em alinhamento com a taxonomia proposta, a questão de pesquisa que trata dos domínios de aplicação IdC é respondida de acordo com quantificação dos artigos coletados na RSL e classificados de acordo com o tema tratado, destacando-se Cidades Inteligentes com 29% e Saúde com 20%.

## 3.2 Internet das Coisas na Cadeia de Frio

Com a pandemia da COVID-19 no plano de fundo deste projeto de mestrado, uma RSL foi realizada para prospectar aplicações de IdC para o rastreamento de vacinas na cadeia de frio, que resultou no artigo “Internet of Things Applications for Cold Chain Vaccine Tracking: A Systematic Literature Review”, apresentado na *International Conference on Information Technology: New Generations (ITNG) 2023* (GARCIA; SOUZA, 2023). A sua íntegra está disponível no Anexo A.

Alkhoori et al. (2021) detalham CryptoCargo, uma solução de containers inteligentes criada para rastrear remessas e, principalmente, identificar ameaças à saúde das embalagens, como alterações de temperatura, humidade e a sua abertura. Duas placas são utilizadas no container para a prova de conceito: a principal é uma Arduino MKR GSM 1400 com módulo GPS, enquanto a secundária é uma ESPWROOM32. A aplicação proposta possibilita a comunicação em tempo real com o protocolo MQTT via conexão Quarta Geração da telefonia móvel (4G) entre o container e os serviços da plataforma em nuvem Azure utilizados, como o “Azure IoT Hub”. Uma aplicação *web* fornece aos usuários finais a possibilidade de acompanhar o rastreamento e realizar consultas analíticas. Além

de melhorias de usabilidade da interface de usuário, são sugeridos como melhorias para trabalhos futuros a criação de uma placa única que atenda as necessidades da solução e o armazenamento local de informações para cenários *offline* por falta de cobertura.

Urbano et al. (2020) apresentam uma abordagem de baixo custo para rastreabilidade de alimentos por meio da criação de uma “smart-tag”, que contém um microcontrolador LPC811M, sensor de temperatura e humidade, bateria e um chip RFID. Para a leitura de múltiplas mercadorias simultaneamente é proposto um gateway baseado no dispositivo ThingMagic M6e RFID, que provê capacidade de comunicação via Wi-Fi, Bluetooth, dentre outros protocolos. É considerada pelos autores uma solução viável em um cenário onde o custo-benefício do monitoramento de itens individuais é importante, sendo o seu custo de produção em massa estimado em menos de dois euros, todavia, o hardware apresentado limita-se ao monitoramento de temperatura.

Sergi et al. (2021), por sua vez, criam uma solução totalmente baseada nas tecnologias da “Microsoft Azure Sphere” com a placa MT3620 MCU e serviços em nuvem como o “Azure IoT Hub”, também acoplando sensores de temperatura, humidade, luminosidade e pressão para monitoramento da logística de alimentos. Entretanto, apesar de apresentar uma aplicação na cadeia de frio, o estudo tem como foco principal a análise de segurança e privacidade oferecidas pelas tecnologias ofertadas pela Microsoft, como a autenticação baseada em certificados, comparando-as a outras soluções apresentadas em estudos relacionados. Como recomendações de continuidade do projeto, são citadas a substituição do hardware utilizado por outro modelo e o teste da solução em campo, uma vez que os resultados apresentados, considerados satisfatórios pelos autores no âmbito de segurança, são baseados em simulações feitas em laboratório com conexão Wi-Fi.

Por fim, Hasanat et al. (2021) propõem um sistema para monitoramento da entrega de vacinas com autonomia de 17,5 horas para utilização em áreas remotas, além de compará-lo com estudos correlatos. A implementação é baseada em Arduino Uno, com sensores de temperatura, humidade e luminosidade, além do módulo de comunicação GSM-GPS-GPRS<sup>6</sup>, responsável pela captura da localização e comunicação com o servidor. Na falta de cobertura da rede celular, a aplicação utiliza um cartão *Secure Digital* (SD) para armazenamento local. Na continuidade do estudo (HASANAT et al., 2020), os autores focam nas especificações das funcionalidades, atores envolvidos, arquitetura e interfaces de interação - via *web* e *mobile*. No entanto, as tecnologias empregadas nas camadas descritas não são detalhadas, bem como resultados do estudo não são apresentados.

(LAWRENCE et al., 2018) aplica tecnologias LoRaWAN e IoT para dar suporte ao setor de saúde em Kikwit, República Democrática do Congo. Os autores implementam uma rede *mesh* em uma aplicação do mundo real, demonstrando a adequação do protocolo em contextos de recursos limitados e áreas com infraestrutura precária. Eles mencionam as

---

<sup>6</sup> Módulo de comunicação para Arduino com *Global System for Mobile Communications* (GSM), GPS e *General Packet Radio Service* (GPRS)

dificuldades de validação dos resultados, especialmente devido à estratégia *ad-hoc* adotada para encontrar locais adequados para instalar gateways e a falta de suporte financeiro em níveis nacionais ou regionais para agilizar sua implantação.

### 3.3 LoRaWAN na Logística e Cadeia de Frio

O potencial dos protocolos de LPWAN, descoberto durante a RSL supracitada, e as limitações mencionadas por Lawrence et al. (2018) na validação dos resultados da sua aplicação criaram a QC4 para este projeto de mestrado. Novas pesquisas exploratória e sistemática foram realizadas, respectivamente, para identificar as características do protocolo LoRaWAN e prospectar as aplicações da literatura que são utilizadas na logística e cadeia de frio.

Em (ZINAS et al., 2017), os autores relatam um protocolo e aplicação chamado “Cow-Track” com base na tecnologia LoRaWAN para rastreamento de gado, implementado e testado em Pogoniani, Grécia. O protótipo do dispositivo inclui sensores, um módulo GPS, um transceptor LoRa e um microcontrolador fechado em uma caixa e colocado na coleira do sino das vacas. Durante seus experimentos, os autores puderam enviar dados por LoRaWAN a uma distância de 6 km e validar a funcionalidade de alerta desenvolvida.

Com foco na manutenção de condições ideais de temperatura, (ENRIKO; ALEMUDA; ADRIANTO, 2022) apresenta uma abordagem baseada em IdC para monitorar a qualidade da vacina durante a distribuição na Indonésia. A abordagem integra vários dispositivos e sensores de IdC, garantindo monitoramento contínuo e alertas imediatos em caso de desvios das condições necessárias. Devido à cobertura de rede, a arquitetura proposta emprega LoRaWAN apenas para nós em repouso, enquanto usa tecnologias celulares para dispositivos em trânsito.

(BOSE; AITHAL; RAJESHWARI, 2022) propõe um sistema de monitoramento de armazenamento a frio de vacinas baseado em LoRaWAN que compreende um nó final com um módulo GPS, gateway LoRaWAN, servidor de rede e uma interface de aplicativo de usuário baseada na *web*. Os autores testaram seu *design* com um gateway estacionário colocado em alta altitude em Bangalore, Índia. Os resultados apresentados indicam que o protocolo é resistente a ruídos para comunicações de longo alcance em áreas urbanas.

Focado no recurso de geolocalização do protocolo, Fargas e Petersen (2017) propõe um sistema com potencial uso em casos como assistência a idosos e rastreamento de animais, priorizando baixo consumo de energia. O sistema utiliza a tecnologia LoRaWAN para rastrear dispositivos IdC sem depender de GPS ou GSM, combinando módulos LoRa, gateways e algoritmos baseados em diferenças de tempo de chegada ou *Time Distance of Arrival* (TDoA) para calcular posições com precisão de cerca de 100 metros em cenários estáticos. Os resultados destacam a viabilidade dessa abordagem, embora melhorias sejam necessárias para aplicações em tempo real, como o aprimoramento de algoritmos e maior

número de gateways.

### 3.4 Diferencial

Este projeto de mestrado aborda algumas das lacunas dos trabalhos relacionados, com foco principal na Teoria de Projeto em que uma abordagem com múltiplos servidores de rede LoRaWAN pode melhorar a cobertura de rede da solução e possibilitar a reutilização de infraestrutura existente. Adicionalmente, apresenta um método de posicionamento de antenas com uma ferramenta de simulação como alternativa a tentativas *ad-hoc* de Lawrence et al. (2018). Não obstante, o estudo de caso para validação do projeto prevê cenários em movimento em áreas urbana e rural, diferentemente de Bose, Aithal e Rajeshwari (2022), Enriko, Alemuda e Adrianto (2022). Há ainda o emprego do recurso de geocalização do protocolo LoRaWAN aproveitando-se da capacidade oferecida por um dos servidores de rede adotado com alta cobertura, fator que não era disponível para Fargas e Petersen (2017) validar o desempenho desta funcionalidade em tempo real. Há ainda fatores secundários, mas ainda importantes, como a extensibilidade do SBIdC-RALCF para outros casos de uso através de abstrações e desacoplamento dos componentes da arquitetura.

O emprego de métodos robustos de validação como estudo de caso e experimento de escalabilidade asseguram o funcionamento do SBIdC-RALCF em cenários próximos do mundo real. No entanto, este estudo não foca no emprego de gateways e criação de rede privada próprios como os trabalhos relacionados, sendo este um fator a ser considerado no eventual avanço do sistema em um Ciclo de Engenharia (WIERINGA, 2014, 3.1).

### 3.5 Considerações Finais

Este capítulo apresentou os principais trabalhos relacionados e os aprendizados extraídos deles, que nortearam esta dissertação de mestrado até a criação do SBIdC-RALCF. A trajetória começou por uma RSL para evidenciar a relevância dos domínios de logística e saúde para aplicações IdC, seguindo para uma RSL dedicada aos obstáculos enfrentados na logística da cadeia de frio, incluindo quais tecnologias de IdC são aplicadas como tratamentos desses problemas, especialmente em áreas remotas com falta de infraestrutura adequada. Com o protocolo LoRaWAN como uma possibilidade no radar, novas pesquisas exploratória e sistemática foram realizadas para identificar as lacunas dos artefatos existentes na literatura, fatores determinantes para estabelecer o PPT deste projeto.

---

## Capítulo 4

# Tecnologias da Internet das Coisas

---

Este capítulo apresenta as tecnologias mais relevantes aplicadas no desenvolvimento do projeto do SBIdC-RALCF após levantamento do estado da arte por meio de RSLs. Isto inclui conceitos de IdC, de redes de sensores sem fio e de longa distância, da plataforma Arduino e placas compatíveis, dos protocolos de rede LoRa e LoRaWAN, dos protocolos da camada de aplicação MQTT e *Hypertext Transfer Protocol Secure* (HTTPS), além do padrão *Representational State Transfer* (REST), que foram levantados nas RSLs e revisões bibliográficas que formam o embasamento teórico deste texto.

### 4.1 Internet das Coisas

*Internet of Things* (IoT), Internet das Coisas (IdC) em português, é um termo cunhado em 1999 por Kevin Ashton, onde objetos - ou coisas - comuns ao nosso dia a dia são dotados de tecnologias embarcadas capazes de identificar, observar, intervir e compreender o mundo, comunicando-se por meio da Internet e redes próprias (ASHTON et al., 2009). A área de IdC ganhou maior relevância na última década, com média de crescimento anual de publicações acima de 85% entre 2015 e 2019, como apontado no estudo bibliométrico<sup>1</sup> de Wang et al. (2021).

Wang et al. (2021) mostra que tecnologias de IdC estão cada vez mais presentes e imperceptíveis no dia a dia das pessoas, ganhando destaque na comunidade científica. É possível identificar tendências de pesquisas, entre 2015 e 2019 o crescimento anual de publicações relacionadas à área superou 85%. Isso ocorre devido a incentivos governamentais e às possibilidades de aplicações em diferentes domínios do nosso dia a dia, como

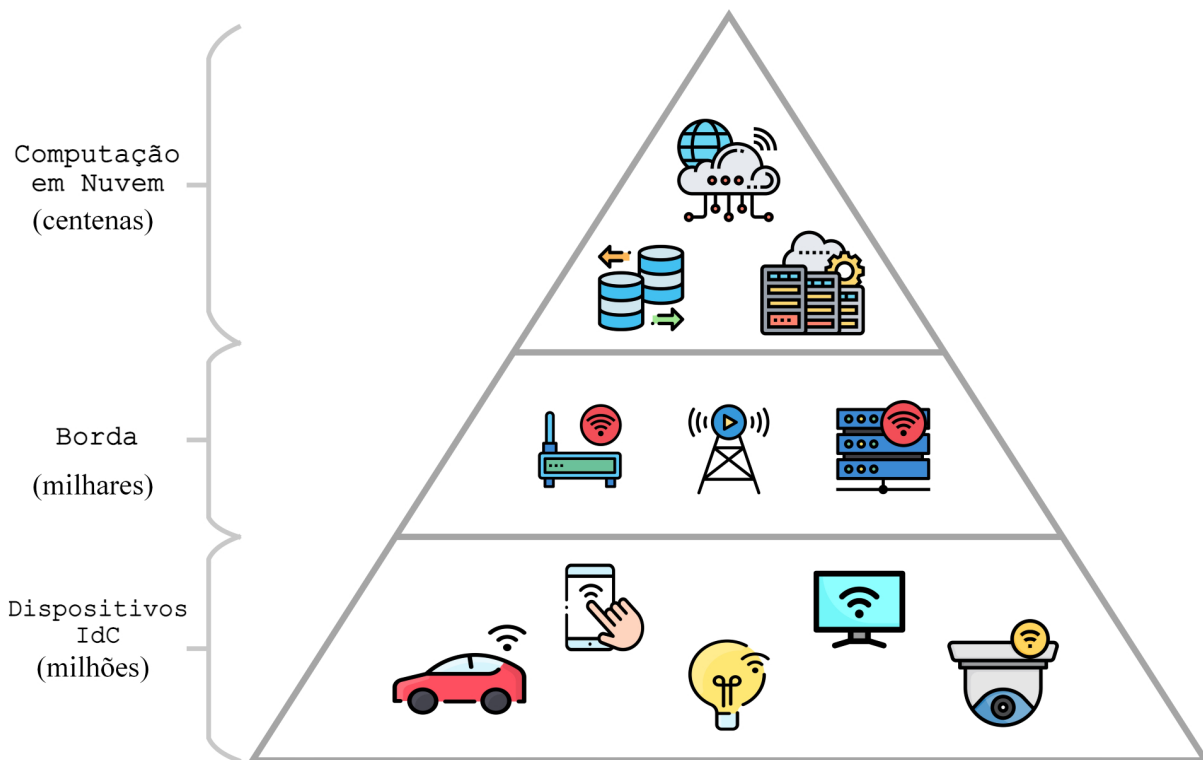
---

<sup>1</sup> Um estudo bibliométrico possibilita identificar tendências de pesquisas por meio da coleta, análise e avaliação de dados quantitativos referentes a publicações científicas

automação residencial, cidades inteligentes, agricultura, saúde e logística (ASGHARI; RAHMANI; JAVADI, 2019).

Em um detalhado levantamento do estado da arte, Kassab e Darabkh (2020) sintetiza o modelo de arquitetura baseado em três camadas, ilustrado na Figura 5, como o mais difundido da literatura de IdC: dispositivos, borda (*fog/edge*) e nuvem (*cloud*). A camada de dispositivos envolve as coisas em si, contemplando desde sensores e atuadores de ambientes a carros autônomos. Borda é a camada intermediária, próxima aos dispositivos, que opera como interface de comunicação (sem fio) entre eles e a camada final, podendo realizar processamentos a fim de reduzir redundâncias, otimizar o tráfego de dados e, conseqüentemente, proporcionar uma experiência em tempo real do sistema ao reduzir a latência de rede. A camada de computação em nuvem provê infraestrutura, plataformas e serviços por demanda, realizando o processamento e armazenamento finais das informações coletadas.

Figura 5 – Arquitetura IdC baseada em três camadas: dispositivos, borda e nuvem



Fonte: Produzido pelo autor

Segundo Hassija et al. (2019), a arquitetura de três camadas supracitada representa o modelo atual de implementação da Internet das Coisas (IdC). Em um momento anterior, os dispositivos interagiam diretamente com a camada de nuvem e, futuramente, com maior capacidade de conexão, espera-se que estes dispositivos possam por vezes comunicar-se entre si diretamente, sem a necessidade de gateways. A camada intermediária ou de borda, além de oferecer melhor experiência para os objetos inteligentes conectados dada

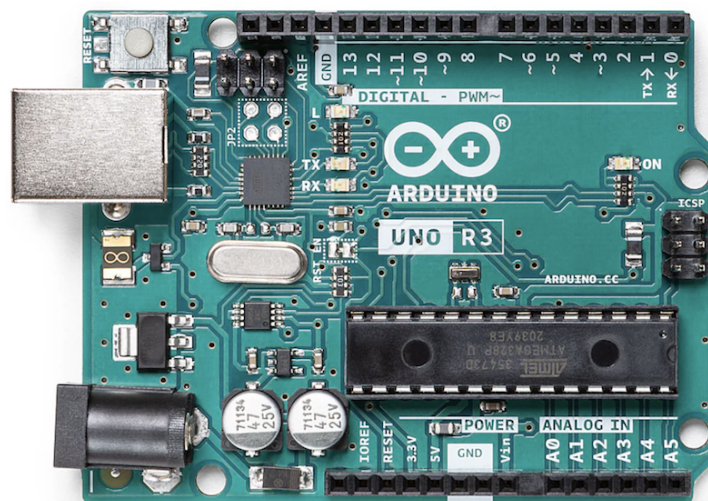
a baixa latência, reduz custos de banda e processamento ao evitar tráfego desnecessário até a *cloud* (QIU et al., 2020).

## 4.2 Arduino e placas compatíveis

Os dispositivos de IdC na cadeia fria, frequentemente chamados de *tags* ou *data loggers*, são compostos de pelo menos um microcontrolador e sensores vinculados para monitorar as condições ambientais. Nos estudos prospectados na RSL realizada nesta pesquisa, a plataforma eletrônica de código aberto Arduino<sup>2</sup> foi a mais empregada em projetos com hardware personalizado (GARCIA; SOUZA, 2023).

O Arduino combina hardware e software para facilitar a construção e programação de dispositivos eletrônicos. Na Figura 6 é ilustrada a Arduino Uno, a placa mais conhecida da plataforma, que possui o microcontrolador “ATmega328” como hardware central. A placa possui diversos componentes, como entradas analógicas e digitais, pinos de alimentação, um botão de *reset* e uma conexão *Universal Serial Bus* (USB) para carregar códigos e fornecer energia. Além disso, o Arduino pode ser alimentado por fontes externas e interagir com sensores, motores, *Light-Emitting Diodes* (LEDs) e outros dispositivos, permitindo o envio e recepção de dados entre o microcontrolador e o circuito conectado (BADAMASI, 2014).

Figura 6 – Terceira revisão da placa Arduino Uno



Fonte: Adaptado de <<https://arduino.cc>>

O software utilizado é o Arduino IDE, que oferece uma interface para escrita e carregamento de código-fonte usando uma versão simplificada de C++. Os códigos, ou *sketches*, possuem duas funções principais: a `setup`, que configura o estado inicial dos componentes,

<sup>2</sup> <<https://arduino.cc>>

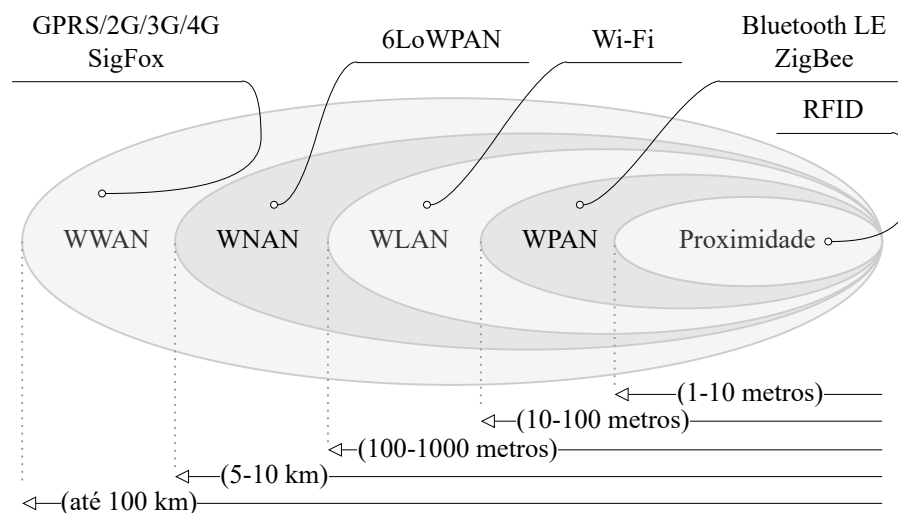
e a `loop`, que executa continuamente o programa principal. A combinação de hardware e software flexíveis possibilita a criação de projetos em diversas aplicações, motivo pelo qual a placa foi amplamente adotada na última década (KONDAVEETI et al., 2021).

O Arduino é uma plataforma aberta e permite que microcontroladores que seguem princípios semelhantes sejam explorados, como o suporte a linguagens de programação simplificadas e integração com sensores e atuadores. Os microcontroladores ESP8266 e ESP32, utilizados neste projeto, são exemplos de alternativas custo-efetivas (LITAYEM; AL-SA'DI, 2023).

### 4.3 Redes sem Fio

Esta seção pincela sobre as principais redes sem fio encontradas na RSL realizada nesta pesquisa (GARCIA; SOUZA, 2023). Comumente chamadas de redes de sensores sem fio ou WSN no contexto dos sensores da IdC, elas desempenham um papel crucial ao permitir a comunicação entre dispositivos e aplicações na nuvem. Na Figura 7 são ilustradas algumas tecnologias de comunicação sem fio e seus respectivos escopos geográficos, resumidas por Kassab e Darabkh (2020).

Figura 7 – Tecnologias sem fio de acordo com a sua cobertura geográfica



Fonte: Adaptado de (KASSAB; DARABKH, 2020)

Exemplo de tecnologia por proximidade, a identificação por radiofrequência ou RFID é geralmente composta por uma etiqueta que pode armazenar uma identificação por meio de transponders. Quando tem sua própria fonte de energia, é considerada uma etiqueta ativa, caso contrário, é uma etiqueta passiva e pode ser ativada por um sinal eletromagnético emitido por um leitor RFID (WEINSTEIN, 2005). As etiquetas RFID passivas são baratas e, juntamente com a capacidade de identificar algo, é uma tecnologia chamativa na área de IdC.

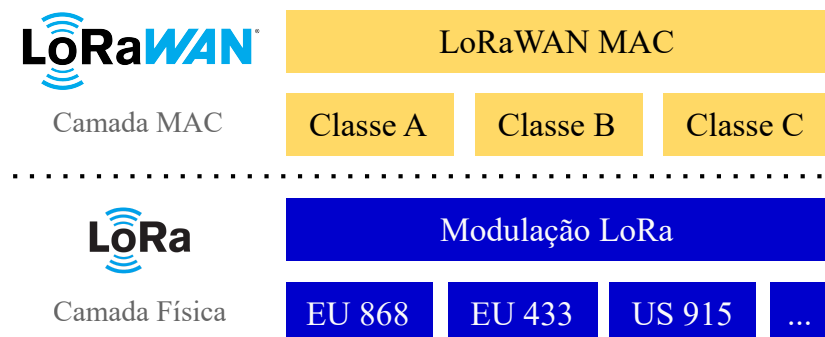
O Wi-Fi, baseado na família IEEE 802.11 (CROW et al., 1997), é a rede local sem fio ou *Wireless Local-Area Network* (WLAN) mais difundida, geralmente usada dentro de uma casa ou edifício. Por sua vez, ZigBee (ERGEN, 2004), 6LoWPAN (MULLIGAN, 2007), Bluetooth e sua evolução de baixa energia *Bluetooth Low Energy* (BLE) (MULLIGAN, 2007) são redes pessoais sem fio ou *Wireless Personal Area Network* (WPAN), que usualmente funcionam no espaço pessoal de 10 a 100 metros (KASSAB; DARABKH, 2020).

SigFox, LoRaWAN e redes de celular são exemplos de redes de longa distância sem fio, ou *Wireless Wide Area Networks* (WWAN), com cobertura regional ou mesmo nacional. O GSM e as gerações seguintes (Terceira Geração da telefonia móvel (3G), 4G e Quinta Geração da rede móvel (5G)) desempenham papel dominante na área de IdC dada a infraestrutura celular existente (KASSAB; DARABKH, 2020). SigFox e LoRaWAN são ditos WWAN de baixa potência devido à sua característica de baixa energia, apesar do longo alcance. O primeiro requer uma assinatura de um provedor sem fio licenciado, enquanto o último, alvo deste estudo, permite possuir a infraestrutura de comunicação (LAWRENCE et al., 2017).

## 4.4 LoRa e LoRaWAN

LoRa e LoRaWAN se complementam para definir um protocolo LPWAN. Conforme destacado na Figura 8, LoRa se refere à camada física, que aplica uma técnica de modulação sem fio baseada em *Chirp Spread Spectrum* (CSS) que permite comunicação de longo alcance com consumo mínimo de energia. Essa modulação permite transmissão de dados robusta por vários quilômetros e opera nas bandas sub-Gigahertz sem licença, como 915 MHz, 868 MHz e 433 MHz, dependendo da região (AUGUSTIN et al., 2016).

Figura 8 – Camadas dos protocolos de rede LoRa e LoRaWAN



Fonte: Adaptado de <<https://www.thethingsnetwork.org>>

Por sua vez, LoRaWAN é o protocolo de camada LoRaWAN que opera em cima do LoRa. Ele define o protocolo de comunicação e a arquitetura do sistema para orquestrar a operação da rede, incluindo como os dispositivos se conectam e transmitem dados,

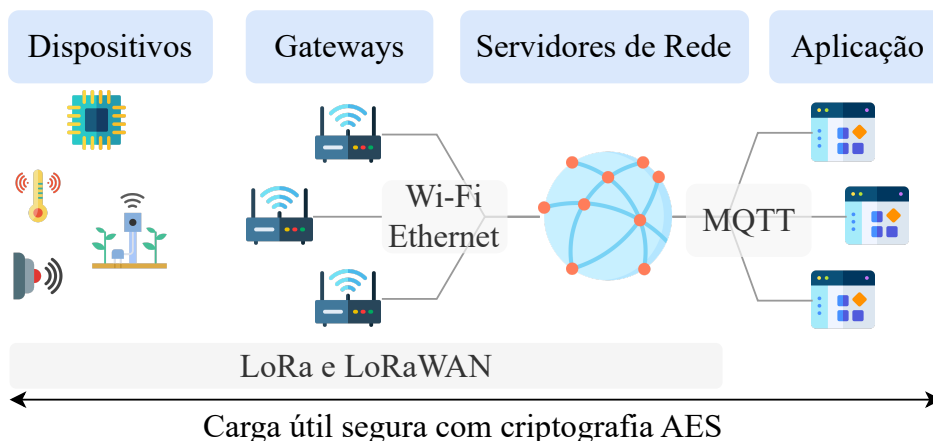
garantindo segurança e escalabilidade (CATTANI; BOANO; RÖMER, 2017). Como o protocolo opera em bandas regionais sem licença, os governos geralmente regulam o ciclo de trabalho dos dispositivos de rádio, que indica a fração de tempo que um recurso pode estar ocupado.

LoRaWAN define três classes de dispositivos para atender às diferentes necessidades de aplicação em relação à latência de comunicação e eficiência energética: dispositivos de classe A têm o menor consumo de energia, são projetados para sensores operados por bateria e permitem duas janelas de recebimento curtas após cada transmissão de envio, ou *uplink*; dispositivos de classe B adicionam intervalos de recebimento programados, usando um *beacon* sincronizado do gateway para abrir janelas de recebimento adicionais em horários definidos, equilibrando o consumo de energia com oportunidades de recebimento de mensagem (ou *downlink*) mais frequentes; Dispositivos de classe C quase sempre escutam *downlinks*, oferecendo latência mínima ao custo de maior consumo de energia, adequados para aplicações onde a resposta imediata é crítica, como sistemas de controle ou atuadores (AUGUSTIN et al., 2016; CATTANI; BOANO; RÖMER, 2017).

#### 4.4.1 Arquitetura LoRaWAN

Uma arquitetura LoRaWAN típica é ilustrada na Figura 9, que é composta de dispositivos finais, gateways, servidores de rede e aplicações (AUGUSTIN et al., 2016). Os dispositivos finais contendo sensores ou atuadores enviam e recebem mensagens sem fio moduladas por LoRa para os gateways, que são conectados ao servidor de rede por meio de um *backhaul*, como Wi-Fi, Ethernet, redes celulares ou satélite. O servidor de rede gerencia todos os componentes de rede LoRaWAN e faz a interface da troca de dados com a aplicação onde a lógica de negócios reside, geralmente por meio de *brokers* MQTT. As comunicações do LoRaWAN são encriptadas com chaves *Advanced Encryption Standard* (AES)-128 *bit*.

Figura 9 – Camadas da arquitetura LoRaWAN e protocolos relacionados



Fonte: Produzido pelo autor

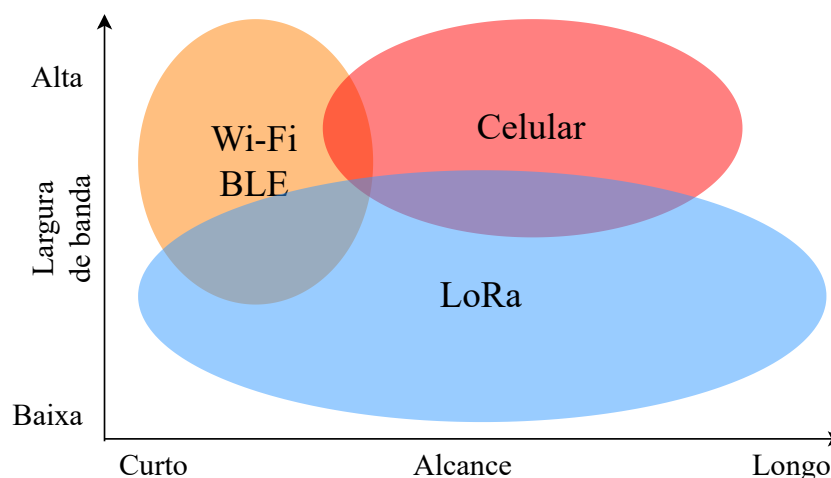
O servidor de rede inclui o software responsável por manipular a ativação de dispositivos na rede, o *Join Server*, que permite que os nós finais se conectem usando *Over-The-Air-Activation* (OTAA) ou *Activation By Personalisation* (ABP) (AUGUSTIN et al., 2016). O primeiro é um método preferencial, onde o dispositivo envia uma solicitação de adesão ao servidor de rede com as chaves *AppEUI* e *DevEUI*, que então responde com chaves de sessão usadas para criptografar comunicações futuras (*AppSKey* e *NwkSKey*), permitindo a adesão dinâmica e segurança elevada. Em contraste, o último envolve a configuração prévia de dispositivos com parâmetros de rede como chaves de sessão e endereços fixos antes da implantação, permitindo comunicação imediata, mas sem a segurança dinâmica e a flexibilidade do OTAA.

Servidores de rede empresarial como *The Things Stack* (TTS)<sup>3</sup> oferecem recursos *plug-and-play* para dispositivos e gateways personalizados operarem juntos. Os provedores de telecomunicações geralmente incluem gateways como parte da solução com recursos criados em cima disso, como geolocalização sem GPS. Em contraste, alternativas de código aberto como ChirpStack<sup>4</sup> permitem operações de rede privada completas atendendo às especificações LoRaWAN.

#### 4.4.2 Largura de banda e alcance

Conforme ilustrado na Figura 10, redes sem fio como Wi-Fi e Bluetooth (incluindo sua versão Low Energy BLE) têm curto alcance, mas alta largura de banda, tornando-as ideais para transmissões de pacotes de vídeo e voz. As tecnologias celulares são destinadas a casos de uso externo de missão crítica e exigem mais energia.

Figura 10 – Largura de banda e alcance do LoRaWAN comparado com outras redes sem fio



Fonte: Adaptado de <<https://www.thethingsnetwork.org>>

<sup>3</sup> <<https://www.thethingsindustries.com>>

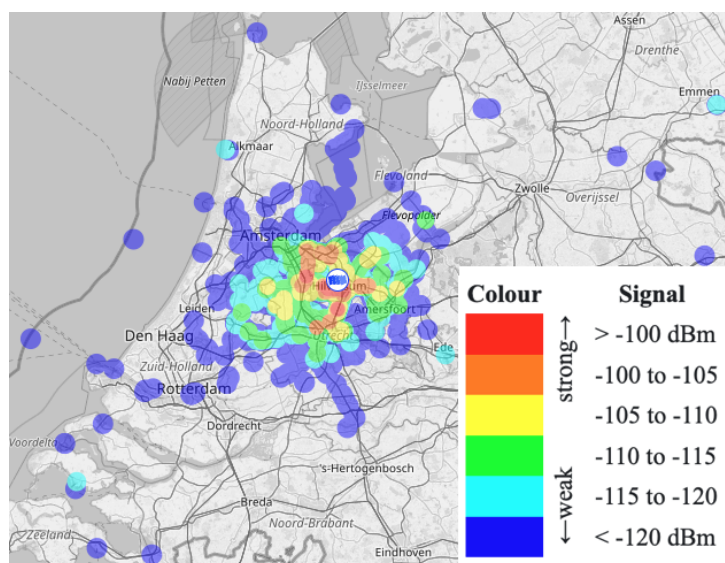
<sup>4</sup> <<https://www.chirpstack.io>>

Protocolos celulares mais orientados para IdC, como *Narrowband Internet of Things* (NB-IoT) e *Long-Term Evolution for Machines* (LTE-M), são energeticamente eficientes em condições favoráveis; no entanto, eles ainda exigem investimento adicional em chip e assinatura periódica, além da infraestrutura móvel em espectro licenciado (LABDAOUI; NOUVEL; DUTERTRE, 2023). Em contraste, a modulação LoRa visa atingir longo alcance com baixo consumo de energia com o custo de ter baixa largura de banda, o que é adequado para sensores e atuadores que não exigem mensagens com grande carga útil. A modulação LoRa tem seis Fatores de Espalhamento, do inglês *Spreading Factor* (SF), de SF7 a SF12, que influenciam a taxa de dados, o tempo no ar, a duração da bateria e a sensibilidade do receptor (AUGUSTIN et al., 2016).

### 4.4.3 Linha de visão

O desempenho do LoRaWAN é significativamente impactado pela linha de visão, do inglês *Line of Sight* (LoS), pois a tecnologia depende de comunicações de radiofrequência de longo alcance e baixa potência. Com uma distância clara entre o transmissor e o gateway receptor, o LoRaWAN pode atingir longo alcance e confiabilidade, geralmente se estendendo por vários quilômetros em áreas rurais, mas pode ser bloqueado por obstáculos como edifícios e árvores, causando atenuação no canal de comunicação. As implantações em áreas urbanas exigem vários gateways ou um mecanismo de roteamento para garantir cobertura de rede robusta e transmissão de dados devido ao LoS restrito (SABAN et al., 2021).

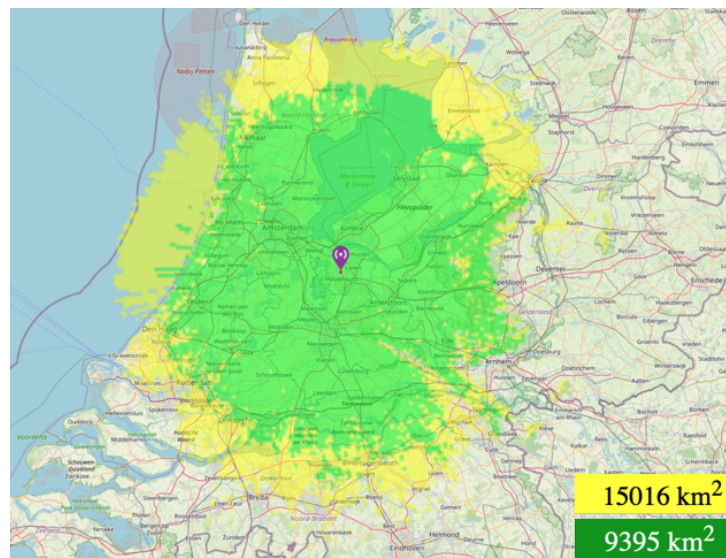
Figura 11 – Mapa de calor do sinal de transmissões recebidas por um gateway a 136 metros de altura em Hilversum, Países Baixos



Fonte: Captura de tela da ferramenta <<https://ttnmapper.org>>

A Figura 11 foi retirada da ferramenta “TTN Mapper”<sup>5</sup>, na qual é ilustrado o mapa de calor do sinal medido em decibéis por miliwatt (dBm) de um gateway posicionado fisicamente em uma torre de 136 metros de altura em Hilversum, Países Baixos. Adicionalmente, uma simulação de uma antena de 902 MHz colocada no mesmo ponto e altitude usando a ferramenta “Radio Mobile Online” (Roger Coudé VE2DBE, 2024) é exibida na Figura 12. A correlação entre a cobertura esperada e a transmissão efetiva do gateway indica a simulação como uma alternativa de mapeamento para posicionamento de antena *ad-hoc* (LAWRENCE et al., 2018) e como esse ponto de acesso se beneficia da LoS devido à topografia plana do país. A atuação deste gateway na validação desta pesquisa é detalhada na seção 6.2.

Figura 12 – Simulação da área de cobertura de uma antena de 902 MHz a 126 metros de altura em Hilversum, Países Baixos



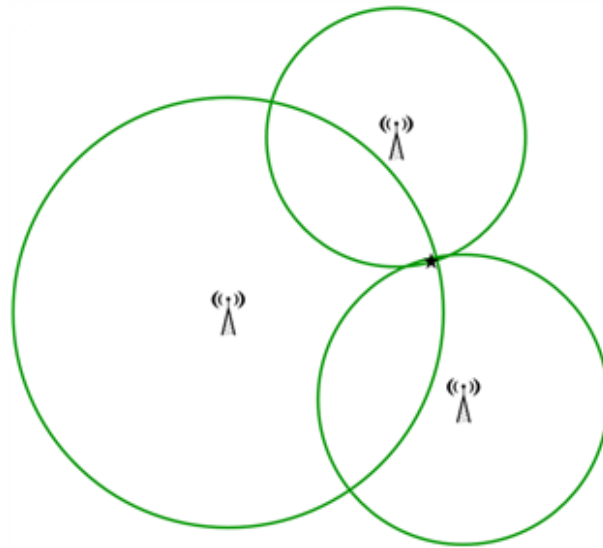
Fonte: Captura de tela da ferramenta de Roger Coudé VE2DBE (2024)

#### 4.4.4 Geolocalização

As três técnicas mais comuns para a resolução da geolocalização de um dispositivo são triangulação, trilateração e multilateração. A triangulação utiliza ângulos de incidência do sinal para determinar a posição através de fórmulas trigonométricas. A trilateração requer a distância entre transmissor e receptor, obtida por métodos como o indicador de intensidade do sinal recebido, ou RSSI. A posição é calculada como a interseção de três círculos, como ilustrado na Figura 13, e necessita de sincronização entre os dispositivos. Já a multilateração baseia-se no tempo de diferença de chegada, ou TDoA, sincronizando os transmissores entre si, mas não o receptor, e determina a posição pela interseção de hipérbolas (KPN, 2024e).

<sup>5</sup> <<https://ttnmapper.org>>

Figura 13 – Geolocalização por trilateração usando RSSI



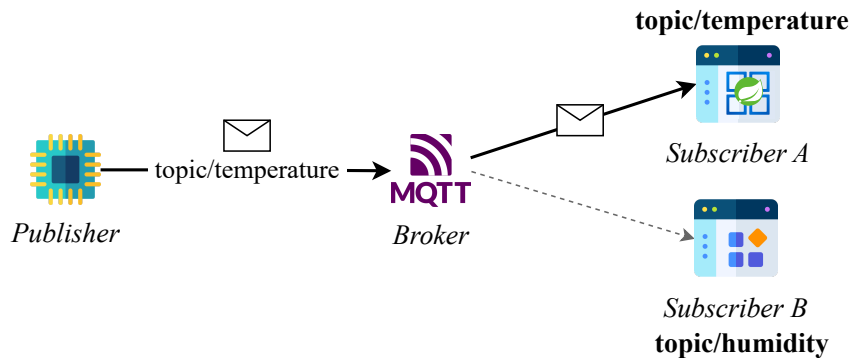
Fonte: (KPN, 2024e)

Os gateways da rede KPN Things suportam geolocalização por trilateração e multilateração via RSSI e TDoA, respectivamente (KPN, 2024e). A seção 5.4 provê maiores informações sobre o desempenho dos algoritmos e a abordagem utilizada para a resolução da geolocalização por esse provedor.

## 4.5 MQTT

MQTT é um protocolo de camada de aplicação sobre *Transmission Control Protocol* (TCP) inventado por Andy Stanford-Clark e Arlen Nipper em 1999, especificado para ser simples de implementar, eficiente em largura de banda e fornecer diferentes níveis de entrega de dados de Qualidade de Serviço, do inglês *Quality of Service* (QoS) (MISHRA; KERTESZ, 2020). O protocolo usa o modelo de publicação e assinatura ou *publish/subscribe*, permitindo comunicação bidirecional entre milhões de clientes por meio de um *broker* intermediário.

Depois que um cliente estabelece uma conexão reconhecida com o *broker* MQTT, ele pode publicar mensagens em tópicos, que são representados por chaves hierárquicas contendo vários níveis separados por barras. Os clientes podem se conectar e assinar esses tópicos no *broker*, que rastreia assinantes e encaminha mensagens para eles. Na Figura 14 é ilustrado um exemplo de comunicação MQTT, onde o cliente “Publisher” publica uma mensagem no tópico `topic/temperature` do *broker*, que a encaminha para o cliente “Subscriber A” inscrito neste tópico. O “Subscriber B” não recebe esta mensagem pois está inscrito apenas no tópico `topic/humidity`.

Figura 14 – *Publisher*, *broker* e *subscriber* envolvidos em uma comunicação via MQTT

Fonte: Produzido pelo autor

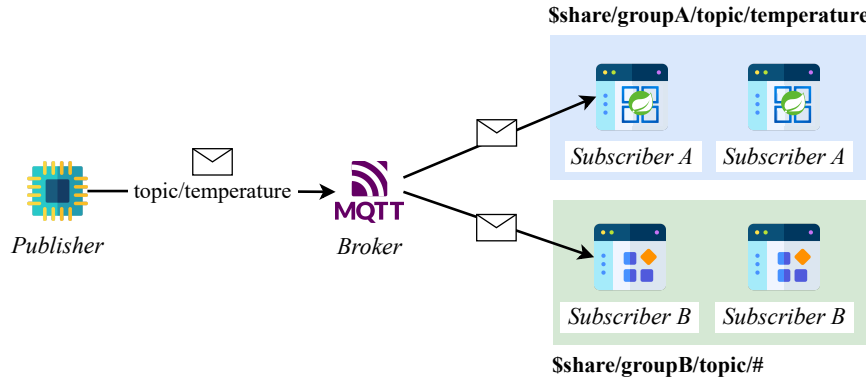
O MQTT define três níveis de QoS. O nível 0 significa “no máximo uma entrega”, o que é útil quando a perda ocasional de dados devido à falta de confirmação é aceitável. O nível 1 deve ser usado se a mensagem exigir confirmação, mas permitir duplicação, pois fornece “pelo menos uma entrega”. O nível 2 tem o mais alto nível de qualidade e fornece entrega de mensagem “exatamente uma vez”, o que é recomendado para aplicações críticas onde a perda e duplicação de dados são inaceitáveis. Níveis mais altos de QoS aumentam o consumo de recursos, como armazenamento e tráfego de rede, devido às etapas adicionais necessárias para confirmação. O protocolo também fornece recursos para aumentar a confiabilidade e a segurança, como mensagens persistentes, autenticação de clientes com usuário e senha no momento da conexão e criptografia de ponta a ponta por meio de *Transport Layer Security* (TLS) (NIKOLOV, 2020).

A versão 5 do protocolo introduz funcionalidades aprimoradas como mecanismos de autenticação, relatório de erros e enfileiramento de mensagens. Essas melhorias garantem que o MQTT possa lidar com as demandas cada vez maiores dos ambientes modernos de IdC, onde o número de dispositivos conectados e a quantidade de dados que eles geram estão crescendo exponencialmente. Uma das melhorias significativas no MQTT 5 é a introdução de Assinaturas Compartilhadas, ou *Shared Subscriptions*, um recurso que permite o balanceamento de carga de mensagens em várias instâncias de cliente (MISHRA; KERTESZ, 2020).

Conforme ilustrado na Figura 15, a assinatura compartilhada garante que somente um cliente de um grupo de assinantes processe a mensagem, removendo a complexidade do lado do cliente de lidar com mensagens concorrentes. Todos os clientes do tipo “Subscriber A” informam um prefixo contendo *\$share* e o identificador do grupo antes do nome do tópico i.e. *\$share/groupA/topic/temperature*. Quando o cliente “Publisher” publica uma mensagem no tópico *topic/temperature*, o *broker* encaminha a mensagem para apenas um cliente de cada assinatura compartilhada. Note que o “Subscriber B” também recebe a mensagem, pois se inscreve para *topic/#*, onde *#* é o chamado *wildcard* multi-

nível, ou caractere coringa.

Figura 15 – Funcionalidade de assinatura compartilhada do MQTT versão 5



Fonte: Produzido pelo autor

## 4.6 HTTP(S)

*Hypertext Transfer Protocol* (HTTP), um protocolo da camada de aplicação do modelo *Open Systems Interconnection* (OSI) (ZIMMERMANN, 2003), foi criado no início dos anos 90 e a sua versão 1.1 padronizada em 1997, sendo esta amplamente adotada em serviços de Internet (FIELDING et al., 1999). Ela trouxe melhorias como o suporte a conexões persistentes, que permitiram a reutilização de conexões TCP para múltiplas requisições. No entanto, suas limitações a tornam ineficiente para páginas modernas devido a problemas como o *head-of-line blocking*, onde uma requisição lenta bloqueia as demais (TREVISAN et al., 2021).

A versão subsequente do protocolo, HTTP/2, foi padronizada em 2015 e introduziu mudanças significativas para melhorar a velocidade do carregamento de páginas, como a compressão de cabeçalhos e multiplexação, que permite múltiplos pedidos simultâneos em uma única conexão (BELSHE; PEON; THOMSON, 2015). Trevisan et al. (2021) cita que, apesar dessas vantagens, a operação sobre o TCP como camada de transporte ainda implica no problema de bloqueio, especialmente em redes com alta latência ou perda de pacotes.

HTTP/3 é a versão atual do protocolo e substitui o TCP pelo QUIC, um protocolo baseado em *User Datagram Protocol* (UDP) desenvolvido pelo Google (BISHOP, 2022). QUIC resolve o problema de bloqueio *head-of-line* ao permitir retransmissões independentes de fluxos dentro de uma conexão e incorpora TLS 1.3, garantindo segurança com tempos de *handshake* reduzidos. O protocolo oferece benefícios em condições adversas, como alta latência ou largura de banda limitada, mas seus ganhos em redes ideais são menos perceptíveis (TREVISAN et al., 2021).

A versão segura do protocolo, chamada de HTTPS, combina os protocolos HTTP com segurança via TLS. Essa camada adicional protege a comunicação entre clientes e servidores, garantindo confidencialidade, integridade e autenticidade. O uso de TLS é obrigatório em HTTP/3, enquanto é opcional nas versões anteriores.

Apesar de ser o protocolo mais conhecido para a comunicação dos serviços de Internet, o HTTP possui sobrecarga em comparação ao protocolo MQTT, principalmente devido a uma quantidade significativa de dados adicionais transmitidos junto com o conteúdo da mensagem (YOKOTANI; SASAKI, 2016). Por esse motivo, aplicações de IdC, especialmente na camada de dispositivos e sensores, tendem a adotar o último em detrimento do primeiro.

## 4.7 REST

REST é um estilo arquitetural amplamente utilizado no desenvolvimento de APIs devido à sua simplicidade e flexibilidade. O protocolo HTTP é utilizado para comunicação entre cliente e servidor, aproveitando os métodos como GET, POST, PUT e DELETE para realizar operações CRUD. Uma API RESTful é projetada para ser *stateless*, ou seja, cada requisição enviada pelo cliente contém todas as informações necessárias para que o servidor a processe e retorne uma resposta, sem depender de um estado armazenado no servidor entre diferentes solicitações (RICHARDSON; RUBY, 2008).

O padrão REST promove a separação de responsabilidades entre o cliente e o servidor. Os recursos são identificados por *Uniform Resource Identifiers* (URIs) e podem ser aceitar e retornar diferentes formatos de dados, como *JavaScript Object Notation* (JSON) ou *Extensible Markup Language* (XML), embora o JSON seja o mais comum devido à sua leveza e compatibilidade com tecnologias de aplicações web modernas (PEZOA et al., 2016). REST permite que sistemas heterogêneos se comuniquem com simplicidade e eficiência, tornando-o uma escolha popular para integrar serviços e criar aplicações escaláveis na *web* com alta interoperabilidade (GARRIGA et al., 2016).

## 4.8 Considerações Finais

As principais tecnologias envolvidas no projeto do SBIdC-RALCF foram exploradas nesse capítulo. Foi apresentada a relevância da IdC no cenário tecnológico e científico, a sua arquitetura de camadas e aplicabilidade em diversos domínios de aplicações. A flexibilidade da plataforma Arduino foi exposta, incluindo placas com princípios similares e de baixo custo. Uma revisão das redes sem fio comumente empregadas em aplicações IdC foi apresentada de acordo com a sua categorização por cobertura geográfica. Os principais conceitos dos protocolos LoRa e LoRaWAN foram detalhados, como as características de largura de banda, comunicações de longa distância, baixo consumo energético, os impac-

tos da linha de visão e o recurso de geolocalização sem GPS. Depois, foram discutidas as versões dos protocolos de aplicação MQTT e HTTPS, além do padrão REST, apresentando suas vantagens e adequações a diferentes cenários de comunicação e integração de sistemas. As discussões aqui apresentadas servirão de base para aprofundamentos técnicos e aplicações práticas nos capítulos subsequentes.

---

## Capítulo 5

# SBIIdC-RALCF

---

O Sistema Baseado em Internet das Coisas para o Rastreo Aplicado à Logística da Cadeia de Frio (SBIIdC-RALCF), é um sistema projetado para abordar os desafios impostos pelo contexto da cadeia de frio ao rastrear produtos perecíveis em áreas remotas. Para tal, diversas tecnologias da IdC, das computações ubíqua e pervasiva, em especial as que circundam o protocolo LoRaWAN, foram empregadas para a aferição, submissão e processamento das condições do ambiente que envolve os bens perecíveis.

Ao suportar vários provedores LoRaWAN, a solução expande a área de cobertura e é benéfica em implantações em larga escala em zonas remotas, garantindo que todos os dispositivos permaneçam dentro do alcance de um servidor de rede. Essa redundância potencial ajuda a manter a transmissão e a recepção de dados consistentes, o que é essencial para aplicações como logística de cadeia fria.

Os atores envolvidos, sabidamente os profissionais de saúde e de logística, podem verificar as condições mensuradas dos produtos, já que os sensores do dispositivo os monitoram constantemente durante o frete. Devido às características da Rede de sensores sem fio (RSSF) LoRaWAN, o sistema também fornece notificações essenciais em caso de violações mesmo que o entregador não possua um dispositivo conectado à Internet.

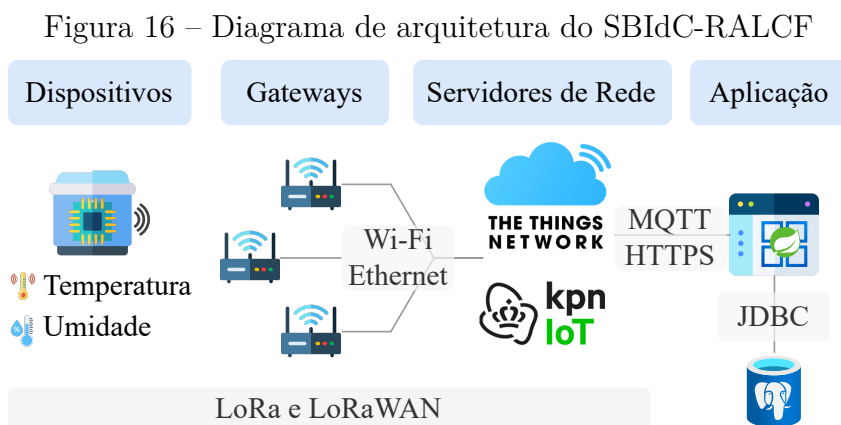
Este capítulo central desta dissertação de mestrado apresenta o projeto do SBIIdC-RALCF da seguinte forma: a seção 5.1 decorre sobre as camadas da arquitetura do sistema, incluindo os principais componentes e suas interações; a seção 5.2 detalha as capacidades funcionais e não-funcionais contidas no sistema para o contexto da cadeia de frio; a seção 5.3 apresenta os dispositivos de uso geral e os sensores empregados como nós finais LoRaWAN; a seção 5.4 descreve as características dos dois servidores de rede LoRaWAN selecionados e os motivos destas escolhas; a seção 5.5 expõe a implementação da aplicação *web* e as suas funcionalidades; e, por fim, a seção 5.6 tece comentários finais

sobre o projeto.

## 5.1 Arquitetura do Sistema

De acordo com a DSM (WIERINGA, 2014), a arquitetura de um sistema é uma estrutura conceitual na qual cada sistema é uma entidade que pode ser decomposta em componentes que interagem para produzir o comportamento geral do sistema. Ela dá suporte à pesquisa baseada em casos, na qual investigamos casos individuais, estudamos sua arquitetura, identificamos mecanismos pelos quais fenômenos gerais de nível de sistema são produzidos e generalizamos caso a caso.

Conforme ilustrada na Figura 16, a arquitetura do SBIdC-RALCF é semelhante à arquitetura típica LoRaWAN, composta por dispositivos, gateways, servidores de rede e a aplicação. A diferença fundamental da abordagem aqui apresentada é o uso de múltiplos servidores de rede para convalidar a teoria do projeto: KPN Things e TTN. Estes servidores permitem o registro e a recuperação de dispositivos nos Países Baixos, onde este projeto de mestrado foi majoritariamente realizado. O KPN Things foca suas operações neste país, enquanto o TTN pode ser utilizado em diversos países ao redor do mundo, incluindo o Brasil com 39 comunidades ativas. O SBIdC-RALCF pode ser executado no cenário nacional com esforço mínimo alterando-se as configurações regionais do protocolo LoRaWAN<sup>1</sup>.



Fonte: Produzido pelo autor

Os dispositivos finais enviam dados de mensagens de *uplink* usando modulação sem fio LoRa para *gateways* conectados por *backhaul* ao servidor de rede. Cada provedor expõe as mensagens do dispositivo ao SBIdC-RALCF por meio de tópicos MQTT e fornece uma API HTTPS que permite comunicação bidirecional. Finalmente, a aplicação *web* consome as mensagens dos dispositivos por meio de subscrição ao *broker* MQTT, armazenando informações importantes em um banco de dados PostgreSQL via API *Java Database*

<sup>1</sup> <<https://www.thethingsnetwork.org/country/brazil>>

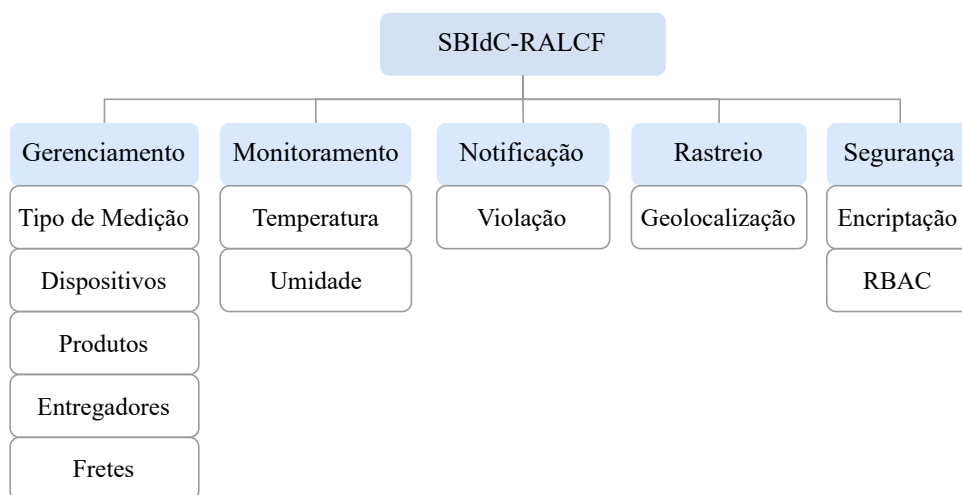
*Connectivity* (JDBC). A aplicação *web*, nativa de nuvem, usa as APIs do servidor de rede para enviar comandos de *downlink* para os dispositivos quando ocorre uma violação.

A escolha das tecnologias empregadas na arquitetura do SBIdC-RALCF preferiu aquelas de código-fonte aberto. O Sigfox, por exemplo, é outro protocolo de rede LPWAN e possui características semelhantes ao LoRaWAN, no entanto, é proprietário (CENTENARO et al., 2016). Outro caso é o uso do MQTT para a integração entre os servidores de rede e a aplicação, pois, além de ser amplamente utilizado em projetos de IdC, é livre de uso, diferentemente de alternativas de plataformas de computação em nuvem como Azure IoT e AWS IoT.

## 5.2 Capacidades

Wieringa (2014) define capacidade como a habilidade de um componente da arquitetura de um sistema de responder a estímulos, a mudanças nas condições ou a eventos temporais. Para propor um sistema que possibilite a rastreabilidade e segurança de produtos da cadeia de frio em áreas remotas, o SBIdC-RALCF suporta os cinco grupos de capacidades detalhados na Figura 17: Gerenciamento, Monitoramento, Notificação, Rastreo e Segurança.

Figura 17 – As principais capacidades do SBIdC-RALCF



Fonte: Produzido pelo autor

Os dois atores ou tipos de usuário contemplados pelo SBIdC-RALCF são Administrador e Entregador. Após acessar a aplicação, conforme detalhado no Apêndice A.1, o Administrador possui acesso irrestrito à capacidade de Gerenciamento, que abrange operações CRUD (acrônimo do inglês para *Create* (Criar), *Read* (Ler), *Update* (Atualizar) e *Delete* (Excluir)) das principais entidades da aplicação, incluindo Tipos de Medição, Dispositivos, Produtos, Entregadores e Frete. Os componentes envolvidos no gerenciamento

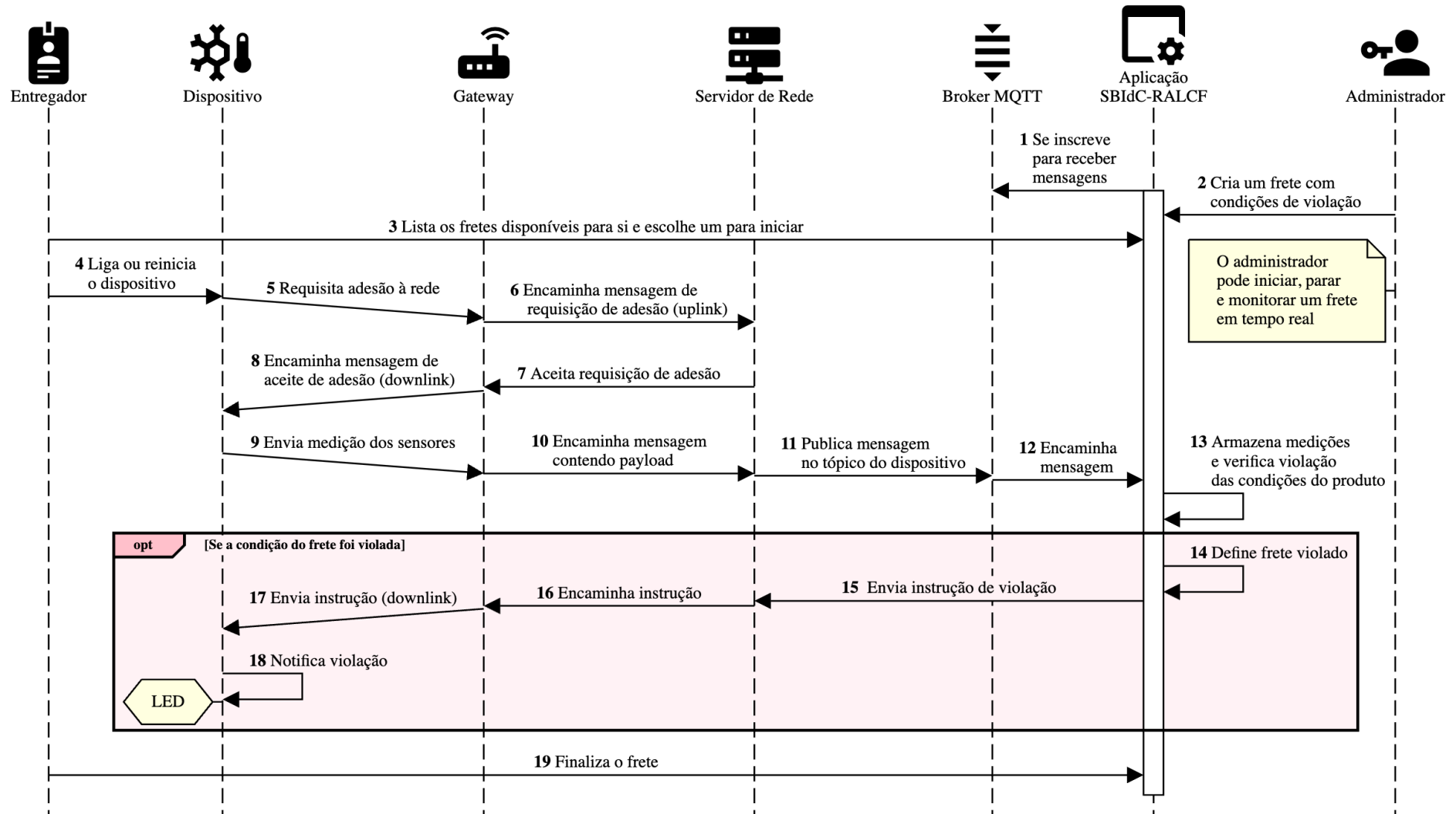
destas entidades estão detalhados nos diagramas de sequência entre os Apêndices A.2 e A.6.

A capacidade de Monitoramento engloba a decodificação, o armazenamento e o acompanhamento das condições medidas pelos sensores dos dispositivos, como temperatura e umidade. O SBIdC-RALCF identifica e emite alertas para qualquer violação das condições pré-definidas para o produto de um frete. A capacidade de rastreamento é baseada na geolocalização LoRaWAN usando o recurso de multilateração, quando disponibilizado pelo servidor de rede. Além da troca de dados criptografados entre os componentes do sistema, a capacidade de segurança define o Controle de acesso baseado em função, do inglês *Role-based access control* (RBAC), para os usuários do sistema.

O diagrama de sequência da Figura 18 mostra como os componentes do SBIdC-RALCF interagem para monitorar uma carga e alertar violações de condições. Com os dispositivos registrados na rede e a aplicação SBIdC-RALCF inscrita em suas medições, o Administrador cria um frete como descrito no Apêndice A.6. O entregador pode listar seus fretes disponíveis, escolher um deles para iniciar e ligar o dispositivo, que imediatamente tenta se juntar ao servidor LoRaWAN. O dispositivo então envia os dados de medição para os gateways via modulação sem fio LoRa, que encaminham as mensagens recebidas para seus respectivos servidores de rede. A aplicação SBIdC-RALCF então escuta e armazena esses eventos, verificando quaisquer limites violados e enviando uma mensagem *downlink* em caso de violação. O dispositivo então emite um alerta para o entregador.

Os demais diagramas de sequência das funcionalidades do SBIdC-RALCF são apresentados no Apêndice A para favorecer a legibilidade desta seção.

Figura 18 – Diagrama de sequência da funcionalidade de frete



Fonte: Produzido pelo autor

## 5.3 Dispositivos

Há uma variedade de dispositivos comerciais LoRaWAN que oferecem funcionalidades de detecção avançadas prontas para uso. Para possibilitar uma personalização completa e obter um entendimento mais profundo do protocolo, as placas de desenvolvimento Az-Delivery NodeMCU v3 (baseada no ESP8266) e Heltec WiFi LoRa 32 (baseada no ESP32) foram escolhidas. Essas placas possuem pinos de entrada/saída de uso geral, do inglês *general-purpose input/output* (GPIO), que permitem flexibilidade para a configuração de diversos sensores e atuadores.

As subseções a seguir detalham o mapeamento das placas aos módulos necessários para que se tornem dispositivos funcionais do SBIdC-RALCF e o código-fonte embarcado, ou *firmware*, que controla a leitura, envio e recebimento de mensagens utilizando o protocolo LoRaWAN.

### 5.3.1 Mapeamento de pinos

As placas foram conectadas com módulos DHT22 para coletar dados de temperatura (de -40 a +80 °C) e umidade (0 a 100% de umidade relativa) do ambiente. O Apêndice D detalha este módulo e a configuração de cada um de seus pinos. Além dos sensores, um transceptor SX1276 também foi conectado à placa NodeMCU v3 para habilitar a modulação LoRa na região da Europa, conforme protótipo ilustrado na Figura 3. A Tabela 1 contém o mapeamento dos pinos da placa e os módulos LoRa e DHT22, disposição esta que também é ilustrada na Figura 19. O Apêndice E detalha o módulo LoRa empregado, incluindo a função atrelada a cada pino.

Tabela 1 – Mapeamento da placa NodeMCU aos módulos SX1276 e DHT22

NodeMCU v3	SX1276	DHT22
D8 (15)	NSS	
D7 (13)	MOSI	
D6 (12)	MISO	
D5 (14)	SCK	
RST	RST	
D1 (5)	DIO0	
D0 (16)	DIO1	
3.3V	VCC	3.3V (+)
GND	GND	GND (-)
D2 (4)		OUT (DATA)

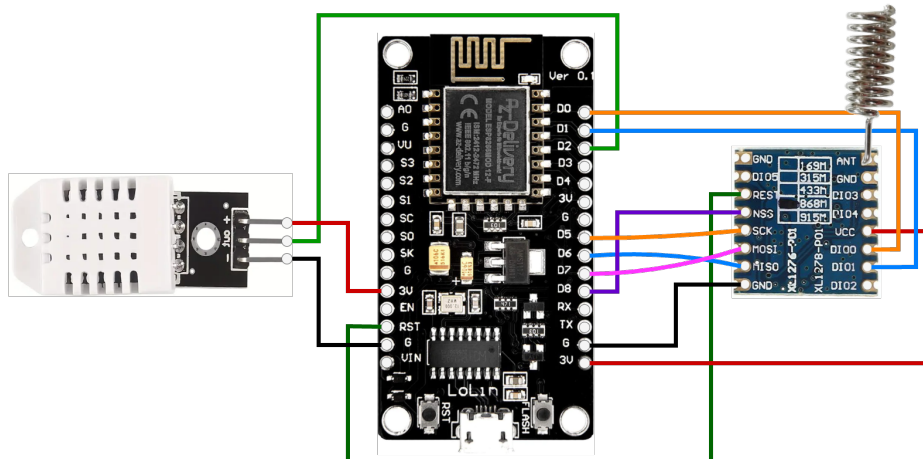
Fonte: Produzido pelo autor

Tabela 2 – Mapeamento da placa WiFi LoRa 32 ao módulo DHT22

WiFi LoRa 32	DHT22
3.3V	3.3V (+)
GND	GND (-)
D2 (2)	OUT (DATA)

Fonte: Produzido pelo autor

Figura 19 – Disposição dos pinos da placa NodeMCU v3 e dos módulos DHT22 e SX1276



Fonte: Produzido pelo autor

A disposição de pinos é simplificada para a placa WiFi LoRa 32 pois ela já contém o transceptor LoRa embarcado. Além dos pinos de alimentação (VCC) e terra (GND) do módulo DHT, o pino OUT (DATA) é conectado ao pino de entrada digital D2 da placa para a leitura das condições do ambiente. O mapeamento deste protótipo, ilustrado na Figura 4, é detalhado na Tabela 2.

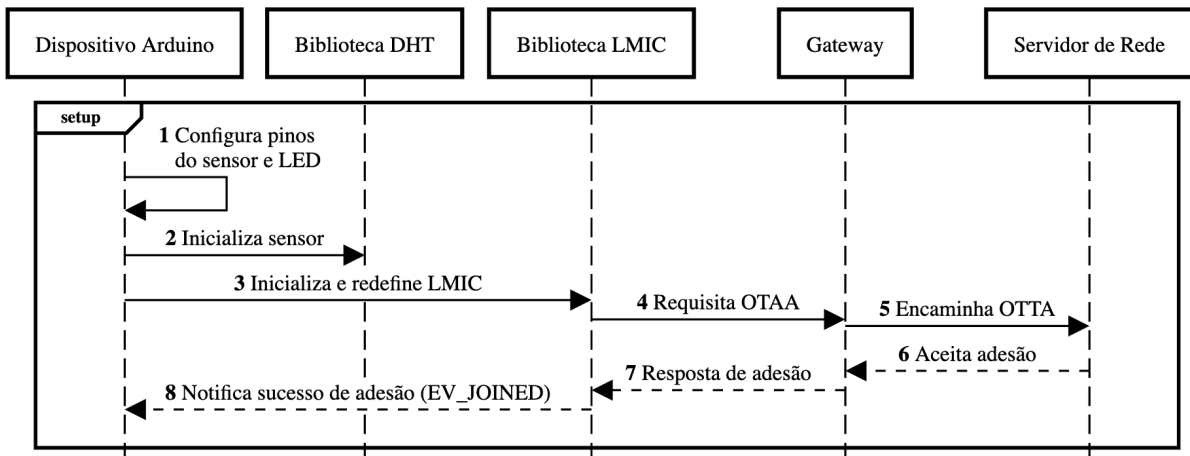
### 5.3.2 Firmware

O Arduino IDE foi utilizado para carregar o *firmware* (código-fonte embarcado) do SBIdC-RALCF em ambas as placas. O código-fonte inclui a biblioteca de sensores DHT<sup>2</sup> para ler a entrada digital do módulo DHT22 e a biblioteca *LoRaWAN-MAC-in-C* (LMIC)<sup>3</sup>, uma implementação LoRaWAN Classe A e Classe B, configurada para a região europeia (863-870 MHz). Os parâmetros específicos de cada placa são definidos no início do código-fonte, como as chaves para a adesão do dispositivo aos servidores de rede LoRaWAN (*DevEUI* e *AppEUI*) e o mapeamento de pinos (descrito na seção anterior).

O código-fonte segue com as funções `setup` e `loop` de um típico programa Arduino. A Figura 20 contém o diagrama de sequência dos componentes envolvidos no fluxo da função `setup`, que 1) configura o pino conectado ao módulo DHT22 no modo entrada (para leitura dos dados) e o pino do LED no modo saída; 2) inicializa o módulo do sensor ao invocar a função `begin` da biblioteca DHT; 3) inicializa o transceptor LoRa e descarta qualquer sessão ou transferência de dados pendentes por meio das funções `os_init` e `LMIC_reset` da biblioteca LMIC, respectivamente; e 4) envia a primeira mensagem LoRa que automaticamente inicia o processo de solicitação de adesão à rede (*join request*) por meio do método de ativação OTAA.

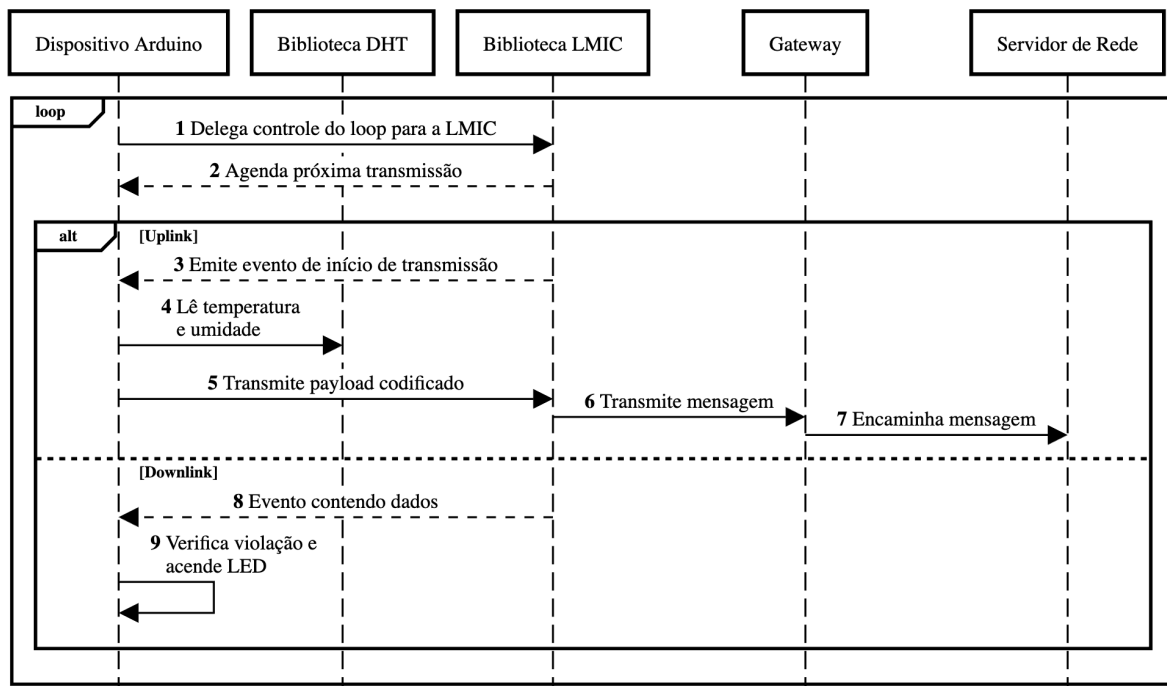
<sup>2</sup> <<https://github.com/adafruit/DHT-sensor-library>>

<sup>3</sup> <<https://github.com/mcci-catena/arduino-lmic>>

Figura 20 – Diagrama de sequência da função *setup* do código-fonte Arduino

Fonte: Produzido pelo autor

O dispositivo recebe - do gateway - a mensagem de confirmação de adesão do servidor de rede (*join accept*) contendo os campos *DevAddr* e *AppNounce*. Este último é decodificado pelo dispositivo para derivar as chaves de sessão (*AppSKey* e *NwkSKey*), que são armazenadas e utilizadas nas comunicações do dispositivo deste momento em diante.

Figura 21 – Diagrama de sequência da função *loop* do código-fonte Arduino

Fonte: Produzido pelo autor

O fluxo da função *loop* é ilustrado no diagrama de sequência da Figura 21. O único procedimento invocado é o *os\_runloop\_once* da biblioteca LMIC, crucial no sistema

orientado a eventos da comunicação LoRaWAN. Ele garante que as operações sejam tratadas prontamente sem bloquear o fluxo principal do programa, como a adesão à rede, o envio de dados (*uplink*) e o recebimento de dados e confirmações (*downlink*). Ao término de cada evento, a próxima transmissão é agendada seguindo o intervalo definido pela constante `TX_INTERVAL`.

Quando o evento de transmissão concluída `EV_TXCOMPLETE` é recebido, a função `readTemperatureAndHumidity` é invocada para ler a temperatura e a umidade através do sensor DHT. Os dados são então codificados e preparados para envio no próximo intervalo de transmissão possível pela função `LMIC_setTxData2` da LMIC. Por outro lado, no fluxo *downlink* com um evento recebido contendo *payload* (carga útil), a função `handleConditionViolation` é chamada para manipular os dados enviados pela aplicação SBIdC-RALCF. A implementação atual liga o LED integrado para indicar que uma violação ocorreu.

## 5.4 Servidores de Rede

A seção de arquitetura prenuncia os servidores de rede adotados no projeto do SBIdC-RALCF que compõem a infraestrutura para acesso à rede LoRaWAN: KPN Things e TTN. Ambos garantem a interoperabilidade dos dispositivos ao implementarem o protocolo e, por suportar o encaminhamento de suas mensagens a *brokers* MQTT e oferecer APIs próprias, permitem o desacoplamento da camada de aplicação.

Em contrapartida, os servidores de rede aqui dispostos possuem distintos modelos de negócio, cobertura de rede e funcionalidades. As subseções a seguir detalham essas especificidades e motivos de terem sido empregados.

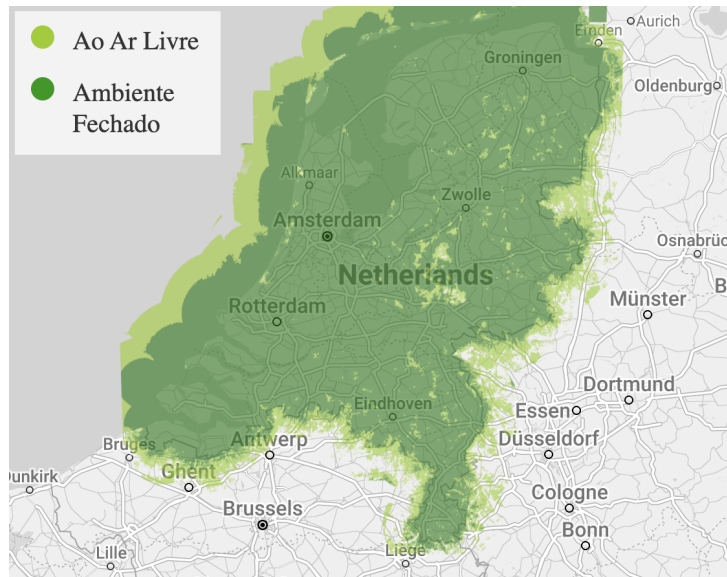
### 5.4.1 KPN Things

KPN Things<sup>4</sup> é uma solução empresarial de IdC da empresa de telecomunicações neerlandesa KPN. Na Figura 22 a abrangente cobertura da rede LoRa da KPN nos Países Baixos é ilustrada, sendo este o principal motivo para a sua escolha no projeto SBIdC-RALCF.

---

<sup>4</sup> <<https://portal.kpnthings.com>>

Figura 22 – Cobertura da KPN Things nos Países Baixos em ambientes fechados e ao ar livre



Fonte: Adaptado de <<https://www.kpn.com>>

O provedor permite o provisionamento de um conjunto de nós LoRaWAN para uma aplicação registrada. Uma das opções é o dispositivo LoRa próprio programável gerando as credenciais utilizadas no processo de ativação OTAA, aderente ao propósito deste projeto. Os dispositivos NodeMCU v3 e WiFi LoRa v3 registrados no portal são mostrados na Figura 23. O processo de registro de um dispositivo no portal KPN Things é detalhado no Apêndice G.1.

Figura 23 – Dispositivos registrados no portal KPN Things

Name	Type	DevEUI / I...	Processing
NodeMCU v3 on KPN	Own LoRa device (...)	0059AC0000...	● Active
WiFi LoRa 32 v3 on KPN	Own LoRa device (...)	0059AC0000...	● Active

Fonte: Captura de tela do portal KPN Things

As mensagens que os dispositivos enviam são codificadas no formato *Sensor Measurement List* (SenML) (JENNINGS et al., 2018) e podem ser encaminhadas para vários tipos de destino, como *brokers* MQTT ou caminhos HTTPS personalizados. O servidor de rede não fornece um *broker* MQTT por padrão, por isso a opção gratuita da EMQ (EMQ Technologies Inc, 2024) foi utilizada para a integração com o SBIdC-RALCF. Os Apêndices G.2 e G.3 mostram a configuração da conexão MQTT e do fluxo de encaminhamento de mensagens para o *broker*, respectivamente. KPN Things fornece uma API REST

que permite que processos remotos consultem dados dos dispositivos e enviem mensagens de *downlink* para eles (KPN, 2024a), possibilitando a comunicação bidirecional entre a aplicação e os nós finais.

#### 5.4.1.1 Multilateração

A plataforma KPN Things permite localizar dispositivos LoRa usando multilateração (KPN, 2024c). Esta funcionalidade remove a necessidade de módulos GPS quando a precisão elevada não é determinante para um projeto, o que reduz o consumo de energia e viabiliza o uso de dispositivos de baixo custo. Ao receber uma mensagem do dispositivo com a opção de geolocalização ativada no portal, o servidor de rede emprega o seu algoritmo de multilateração para estimar a localização e, além de mensagem contendo o payload, também envia a latitude e a longitude estimada do dispositivo no formato SenML para os destinos configurados.

Tabela 3 – Os algoritmos de geolocalização LoRa da KPN Things.

Algoritmo	Quantidade de Gateways	Precisão
TDoA em áreas urbanas	3 ou mais	Até 250 metros
TDoA em áreas rurais	3 ou mais	Até 100 metros
RSSI	1 ou mais	Até 500 metros
<b>Ambos:</b> primeiro TDoA, depois RSSI como reserva	Melhor dos dois	Melhor dos dois

Fonte: Adaptado de (KPN, 2024c)

A Tabela 3 contém um resumo dos três algoritmos da KPN Things utilizados para estimar a geolocalização do dispositivo LoRa: TDoA em áreas urbanas, TDoA em áreas rurais e RSSI (KPN, 2024e). A opção “Ambos” é a padrão, na qual TDoA e RSSI são combinados para atingir o melhor resultado possível. A escolha do algoritmo é feita automaticamente pela plataforma de acordo com o plano do cliente. Na Tabela 4 são apresentados os testes de desempenho realizados pela KPN, onde o algoritmo RSSI e “Ambos” ter maior taxa de sucesso que TDoA, enquanto a precisão do TDoA é melhor do que “Ambos” e, de acordo com a empresa, significativamente melhor que RSSI (KPN, 2024d).

Tabela 4 – Desempenho dos algoritmos de geolocalização LoRa da KPN Things.

Algoritmo	Taxa de sucesso	Precisão mediana	Precisão média
TDoA	78%	181 m	78 m
RSSI	98%	1.475 m	4.173 m
Ambos	99%	373 m	1.002 m

Fonte: Adaptado de (KPN, 2024d)

### 5.4.2 The Things Network

*The Things Network*<sup>5</sup> é uma rede comunitária operada pela *The Things Stack* (TTS). Esta rede foi escolhida para simular uma implantação privada pois permite a adição de gateways próprios e, por conter gateways dos usuários da comunidade, possui cobertura reduzida. A TTN aplica uma Política de Uso Justo, do inglês *Fair Use Policy* (FUP), que limita o tempo de transmissão de *uplink* a 30 segundos por dia por nó e de *downlink* a 10 mensagens por dia por nó.

A configuração da TTN é semelhante à da KPN. Após definir a aplicação para a região desejada (Europa), registra-se o dispositivo próprio no servidor de rede e as credenciais para ativação OTAA são geradas para ele. Os dispositivos do SBIdC-RALCF registrados na TTN são mostrados na Figura 24. A configuração da aplicação e a inclusão de um dispositivo de uso geral no servidor de rede são detalhados nos Apêndice H.

Figura 24 – Dispositivos registrados no portal TTN



NAME AND ID	DEVEUI	JOINEUI	LAST ACTIVITY
WiFi Lora v3 on TTN eui-70b3d57ed0066e49	70 B3 D5 7E D0 06 6E 49	1E 9A 6F DF 31 96 69 15	Apr 21, 2024
NodeMCU v3 on TTN eui-70b3d57ed0066e45	70 B3 D5 7E D0 06 6E 45	CA D5 B5 56 A2 EA E4 72	Jun 29, 2024

Fonte: Captura de tela do portal TTN

A rede TTN fornece um *broker* MQTT versão 3.1.1 (somente QoS 0) para o qual as mensagens enviadas pelos dispositivos são publicadas (The Things Industries, 2024c). As mensagens de *downlink* podem ser enviadas pela aplicação para tópicos dos dispositivos neste mesmo *broker* ou através de uma API REST (The Things Industries, 2024b). O SBIdC-RALCF utiliza esta última opção para encaminhar mensagens de violação para os nós finais, além de consultar suas informações gerais.

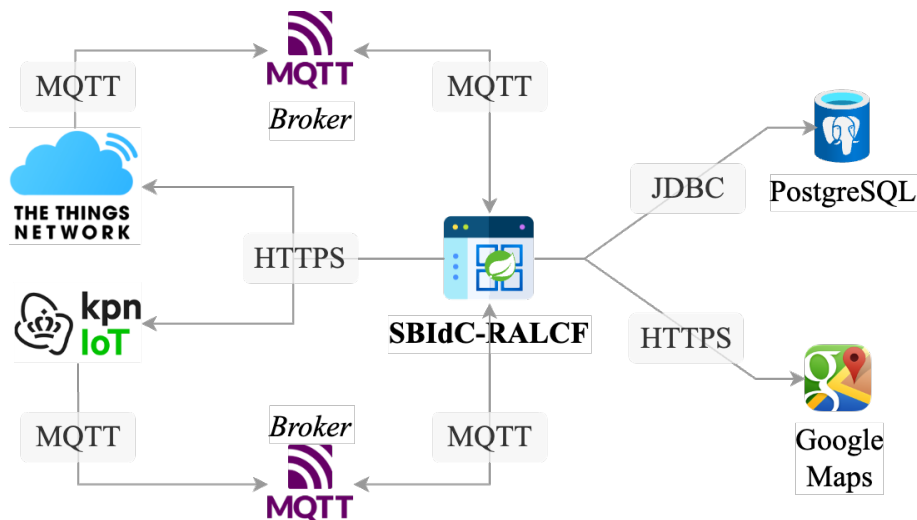
Diferentemente do KPN Things, o TTN permite que gateways LoRaWAN próprios sejam registrados na sua rede. Depois, o aplicativo “TTN Mapper” pode ser usado para mostrar sua cobertura em um gráfico semelhante ao da Figura 11. No entanto, o escopo deste projeto de mestrado não contempla gateways próprios.

<sup>5</sup> <<https://www.thethingsnetwork.org>>

## 5.5 Aplicação Web

A aplicação *web* SBIdC-RALCF, elemento central na Figura 25, foi implementada como um microsserviço Java Spring Boot <sup>6</sup> e é nativo da nuvem, ou seja, pode ser executada dentro de um ambiente de computação em nuvem. Por meio do *framework* Spring Integration<sup>7</sup>, ela se inscreve como consumidor dos tópicos dos *brokers* MQTT e processa as mensagens dos dispositivos que foram encaminhadas pelos provedores KPN Things e TTN. Há também a integração síncrona estes servidores de rede através de uma API REST, que habilita a recuperação de dados dos dispositivos e o envio de mensagens (*downlink*) para eles no caso de violações das condições do produto. O acesso ao banco de dados PostgreSQL<sup>8</sup>, que armazena todos os dados da aplicação, é abstraído via JDBC. A API REST do Google Maps (Google Maps Platform, 2024) é usada para obter a geolocalização dos endereços de origem e destino dos fretes.

Figura 25 – Integrações da aplicação *web* do SBIdC-RALCF



Fonte: Produzido pelo autor

Algumas interfaces devem ser implementadas para incluir um novo servidor de rede na aplicação: `NetworkEndDevice`, `MessageHandler` e `NetworkProviderService`. A primeira é uma generalização dos dados básicos de um dispositivo, como o seu identificador único, nome e servidor de rede. A segunda representa um cliente assinado em tópicos do *broker* MQTT, que deve decodificar a mensagem de acordo com o formato estabelecido pelo provedor de rede. Por exemplo, a implementação `KpnMessageHandler` extrai a carga útil (medidas e geolocalização) das mensagens no formato SenML publicadas pela plataforma da KPN. Por fim, a última define os métodos `listDevices` e `notifyViolation` que devem ser implementados para listar os dispositivos do servidor de

<sup>6</sup> <<https://spring.io/projects/spring-boot>>

<sup>7</sup> <<https://spring.io/projects/spring-integration>>

<sup>8</sup> <<https://www.postgresql.org>>

rede e notificar os dispositivos sobre uma violação ocorrida. Como referência, a implementação `TtnNetworkProvider` integra-se com a API da rede TTN (The Things Industries, 2024b) para recuperar os dispositivos registrados e enviar mensagens de *downlink* a eles.

Outro exemplo de extensibilidade do SBIdC-RALCF é a interface `DevicePayloadDecoder`. As suas implementações são instâncias responsáveis por decodificar a carga útil enviada por um dispositivo. Com isso, diversos conjuntos de sensores e tipos de medidas podem ser embarcados na aplicação. No escopo deste trabalho há apenas a implementação `TemperatureHumidityDecoder`, que espera um par de números que representam a temperatura e a umidade relativa do ar, aferidas pelo DHT22.

O arquivo `application.yml` do código-fonte<sup>9</sup> da aplicação contém os parâmetros necessários para conectar-se aos *brokers* MQTT para os quais as mensagens dos dispositivos são publicadas (Apêndice G.2, The Things Industries (2024c)) e às APIs dos servidores de rede e do Google Maps (KPN, 2024a; The Things Industries, 2024a; Google Maps Platform, 2024).

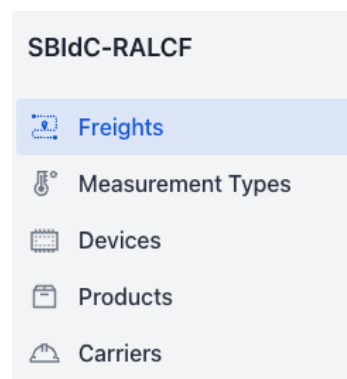
### 5.5.1 Interface do Usuário

A biblioteca Vaadin<sup>10</sup> foi utilizada para gerar a interface *web*, incluindo componentes da versão *Pro*, como gráficos, mapas e CRUD. Vaadin integra-se com o *framework* Spring Security<sup>11</sup>, que garante autenticação e autorização da aplicação *web*, controlando rotas públicas e privadas da aplicação de acordo com o papel do usuário (RBAC), além de gerar interfaces completas como o formulário de acesso ilustrado na Figura 26. Após se autenticar no SBIdC-RALCF, o usuário tem acesso aos itens de menu de acordo com o seu papel: Administrador ou Entregador. As opções apresentadas na Figura 27 abrangem todas as funcionalidades CRUD descritas no Apêndice A.

Figura 26 – Formulário de *login* gerado pela biblioteca Vaadin

Fonte: Captura de tela do SBIdC-RALCF

Figura 27 – Itens de menu de um usuário Administrador



Fonte: Captura de tela do SBIdC-RALCF

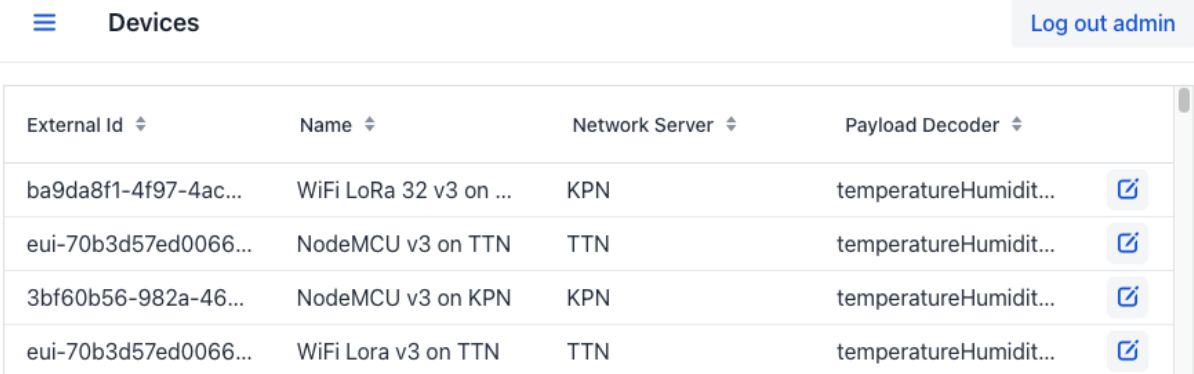
<sup>9</sup> <<https://github.com/alexfabgarcia/cold-chain-tracking>>

<sup>10</sup> <<https://vaadin.com>>

<sup>11</sup> <<https://spring.io/projects/spring-security>>

O Administrador pode listar os dispositivos de ambos os provedores e atribuir a cada um deles um decodificador de carga útil - ou *payload* - disponível. Na Figura 28, os nós finais provenientes de ambos os servidores de rede são listados. O “Decodificador de Payload de Temperatura e Umidade” é a implementação padrão para dados de dispositivos desconhecidos e decodifica a carga útil codificada pela biblioteca LMIC em uma implementação de interface Java Map de par de chaves.

Figura 28 – Listagem de dispositivos registrados na KPN Things e TTN



External Id	Name	Network Server	Payload Decoder
ba9da8f1-4f97-4ac...	WiFi LoRa 32 v3 on ...	KPN	temperatureHumidit...
eui-70b3d57ed0066...	NodeMCU v3 on TTN	TTN	temperatureHumidit...
3bf60b56-982a-46...	NodeMCU v3 on KPN	KPN	temperatureHumidit...
eui-70b3d57ed0066...	WiFi Lora v3 on TTN	TTN	temperatureHumidit...

Fonte: Captura de tela do SBIdC-RALCF

Na Figura 29 está ilustrada a configuração da vacina *Novavax COVID-19* como um tipo de produto no SBIdC-RALCF. O Administrador configurou previamente o tipo de medição “Temperatura” com sua unidade de preferência (°C ou °F) e o associou ao produto, indicando os limites mínimo e máximo para garantir a segurança do bem perecível. Esse recurso generaliza o SBIdC-RALCF para uso em outros domínios além da cadeia de frio, pois o usuário pode configurar vários tipos de medição e atribuí-los aos tipos de produto adequadamente.

Figura 29 – Restrições da vacina “Novavax COVID-19” gerenciadas no SBIdC-RALCF



Name • **Novavax COVID-19**

Category • **VACCINE**

Measurement Type

Temperature X


Measurement Type	Minimum	Maximum
Temperature (°C)	2.0	8.0

Fonte: Captura de tela do SBIdC-RALCF

Após definir todas as entidades básicas, o Administrador do SBIdC-RALCF pode criar um frete definindo o tipo de produto, o dispositivo que o acompanhará, o entregador, a origem, o destino e uma descrição opcional. Na Figura 30, é exibida uma configuração de frete para um frete de vacina “Novavax COVID-19” entre dois centros de vacinação nos Países Baixos.

Figura 30 – Frete entre dois centros de vacinação nos Países Baixos

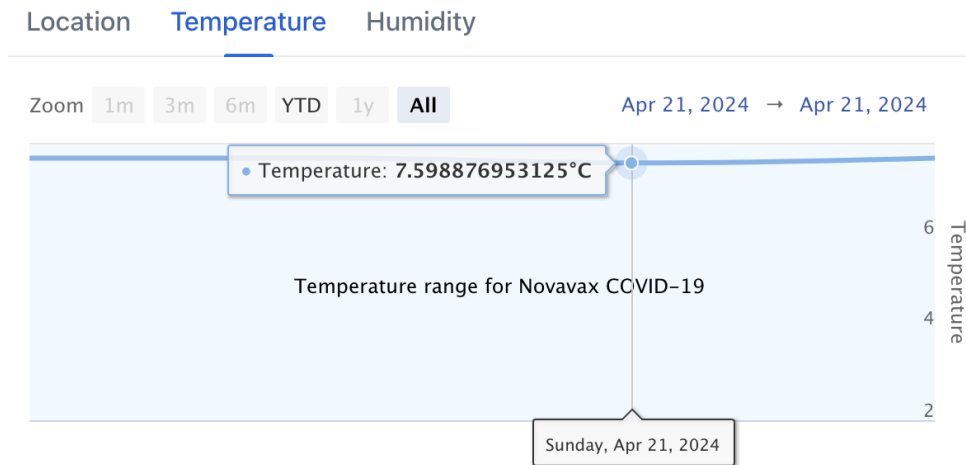
<b>Product •</b> <input type="text" value="Novavax COVID-19"/> × ▾ <small>Type at least 3 characters and press enter</small>	<b>Device •</b> <input type="text" value="NodeMCU v3 on KPN"/> × ▾ <small>Type at least 2 characters and press enter</small>
<b>Carrier •</b> <input type="text" value="Alex Garcia"/> × ▾ <small>Type at least 3 characters and press enter</small>	<b>Description</b> <input type="text" value="Batch 12345"/>
<b>Origin •</b> <input type="text" value="Burgemeester de Bordesstraat 80, 1404"/> × ▾ <small>Type at least 5 characters and press enter</small>	<b>Destination •</b> <input type="text" value="Albert Schweitzerlaan 31, 7334 DZ Apeld"/> × ▾ <small>Type at least 5 characters and press enter</small>



Fonte: Captura de tela do SBIdC-RALCF

O usuário Entregador pode então acessar o SBIdC-RALCF iniciar um frete atribuído a ele ou apenas ligar o dispositivo final em sua posse, que inicia o processo de ativação e, em seguida, envia mensagens para o servidor de rede. Os assinantes MQTT da aplicação manipulam constantemente os eventos do dispositivo e armazenam as medições e a geolocalização (no caso da KPN Things). Estes dados podem ser visualizados em tempo real na aplicação pelos usuários, conforme ilustrado na Figura 31.

Figura 31 – Medições de temperatura para um frete da vacina “Novavax COVID-19”



Fonte: Captura de tela do SBIdC-RALCF

Além de armazenar e exibir a carga útil enviada pelos dispositivos, a aplicação marca um frete como violado quando identifica que ao menos uma das condições de segurança do produto foi violada. Uma mensagem de *downlink* é enviada ao dispositivo por meio da API do servidor de rede neste caso. Desta forma, o Entregador é notificado e fica ciente da violação mesmo que esteja sem acesso à aplicação SBIdC-RALCF.

### 5.5.2 Diagrama de Entidade-Relacionamento

O Diagrama de Entidade-Relacionamento (DER) do projeto SBIdC-RALCF é ilustrado na Figura 32, composto pelas entidades do sistema: `users`, `authorities`, `carrier`, `freight`, `device`, `product`, `measurement_type`, `product_measurement_type`, e `device_measurement`. A biblioteca Flyway<sup>12</sup> é utilizada na aplicação *web* para gerenciar os arquivos de migração do esquema do banco de dados PostgreSQL. A tabela `flyway_schema_history`, que armazena os metadados utilizados neste processo de migração, não está representada no DER.

A entidade `users` contém informações de *login* e estado do usuário e está vinculada à entidade `authorities`, utilizada para gerenciar os níveis de permissão de acesso (RBAC), e à entidade `carrier`, que armazena os dados pessoais do entregador.

Os dispositivos recuperados dos servidores de rede são representados pela entidade `device`, que armazena, por exemplo, o seu identificador único estendido (`eui`) e o decodificador da carga útil associado (`payload_decoder`). Os dispositivos estão vinculados à entidade `device_measurement`, que registra o momento e o valor medido pelo sensor para determinado tipo de medida, correspondido pela entidade `measurement_type`.

Os tipos de produto gerenciados pelo SBIdC-RALCF são retratados pela entidade `product`, contendo o seu nome e categoria, como vacina e soro antiofídico. A entidade

<sup>12</sup> <<https://github.com/flyway/flyway>>

`product_measurement_type` especifica os tipos de medições permitidos para cada produto, provenientes da entidade `measurement_type`, além dos limites mínimo e máximo válidos das condições de preservação deste tipo de produto.

A entidade `freight` centraliza os dados dos fretes, definindo o produto, dispositivo e entregador associados, além de detalhes de origem, destino e estado do frete. As colunas de origem e destino do frete são do tipo `jsonb` do banco de dados PostgreSQL<sup>13</sup>, que permitem o armazenamento dos dados retornados da API de geolocalização (Google Maps Platform, 2024) no formato JSON. A coluna `violated` indica que a condição de segurança do produto do frete foi violada, enquanto as colunas `start_at` e `finished_at` indicam o momento que o frete foi iniciado e finalizado, respectivamente.

---

<sup>13</sup> <<https://www.postgresql.org/docs/current/datatype-json.html>>

Figura 32 – Diagrama de Entidade-Relacionamento do projeto SBIdC-RALCF



Fonte: Gerado por meio da ferramenta DataGrip<sup>14</sup>

## 5.6 Considerações Finais

Este capítulo apresentou os detalhes da arquitetura e implementação do SBIdC-RALCF, incluindo sua arquitetura, capacidades e principais componentes. Este sistema é uma proposta para enfrentar os desafios do rastreamento de produtos perecíveis na logística da cadeia de frio. A combinação de tecnologias, como LoRaWAN e MQTT, apresentou a capacidade do projeto de operar em ambientes remotos, visando a coleta e transmissão contínua de dados essenciais, como temperatura e umidade, enquanto fornece notificações em tempo real sobre violações de condições vitais dos produtos. A integração com múltiplos provedores de rede amplia a área de cobertura e a redundância do sistema, fatores vitais para aplicações em redes intermitentes. Além disso, a arquitetura modular e componentes extensíveis do SBIdC-RALCF possibilitam o seu uso em outros contextos além da cadeia de frio.

---

## Capítulo 6

# Validação e Análise dos Resultados

---

Este capítulo detalha a validação e análise dos resultados do SBIdC-RALCF, que pode ser dividida em três etapas: a ativação dos dispositivos LoRaWAN, um estudo de caso e um experimento de escala. As integrações e as capacidades desenvolvidas na camada de aplicação foram validadas com testes automatizados com as bibliotecas JUnit<sup>1</sup> e de teste do *framework* utilizado (Spring). A primeira seção se concentra na garantia do funcionamento dos dispositivos em torno do protocolo LoRaWAN, em especial a sua ativação junto ao servidor de rede e o envio e recebimento de mensagens. Em sequência, é apresentado um estudo de caso que visa a verificar os efeitos causados pelo artefato no contexto da logística da cadeia de frio. Por fim, é exposto o experimento de escala empregado para validar o comportamento da aplicação em uma situação próxima ao mundo real.

### 6.1 Ativação do dispositivo

Os protótipos do SBIdC-RALCF foram primeiro colocados em posições estacionárias em ambiente fechado e ao ar livre para verificar a sua ativação em ambos os servidores de rede. A Listagem 6.1 mostra a saída de monitor serial do Arduino IDE de um dispositivo ao manipular um evento `EV_JOINED` da LMIC, contendo as chaves de sessão de rede e de sessão do aplicativo do método OTAA. As mensagens de requisição e aceite de adesão da rede KPN Things são mostradas na Figura 33.

---

<sup>1</sup> <<https://junit.org>>

Listagem 6.1 – Saída de monitor serial do Arduino IDE durante o processo de OTAA

```
730507: EV_JOINED
devAddr: 15AE(...)
AppSKey: 67-F4-EC-(...)
NwkSKey: AC-AC-6A-(...)
```

Fonte: Produzido pelo autor

Figura 33 – Mensagens de requisição e aceite de adesão no portal KPN Things

Message type	Location	Timestamp	FCnt Down	FCnt Up	Signal
Confirmed data + mac		Fri 2024-06-28 00:39:26.935	0		
Unconfirmed data + mac		Fri 2024-06-28 00:39:25.935		0	
Join accept		Fri 2024-06-28 00:39:23.984			
Join request		Fri 2024-06-28 00:39:17.984			

Fonte: Captura de tela do portal KPN Things

Uma vez que o processo de ativação foi finalizado, o dispositivo começa a enviar as medições coletadas pelo sensor DHT de acordo com o intervalo definido. Neste momento já é possível validar que as mensagens *uplink* são publicadas nos *brokers* MQTT e que a aplicação *web* do SBIDC-RALCF recebe, decodifica e armazena a carga útil em sua base de dados.

Para validar a comunicação bidirecional, uma situação de violação de condição foi simulada criando-se um frete com o protótipo atrelado. Após identificar tal situação a partir das medições recebidas, a aplicação *web* notifica a violação ao servidor de rede com uma mensagem *downlink*. A Listagem 6.2 mostra a saída de monitor serial de um protótipo ao receber o evento *EV\_TXCOMPLETE* com uma carga útil, quando o LED do dispositivo é aceso para indicar a violação das condições do ambiente.

Listagem 6.2 – Saída de monitor serial do Arduino IDE durante violação de condição

```
819200: EV_TXCOMPLETE (inc. RX waiting)
Condition violated. Turning LED on...
```

Fonte: Produzido pelo autor

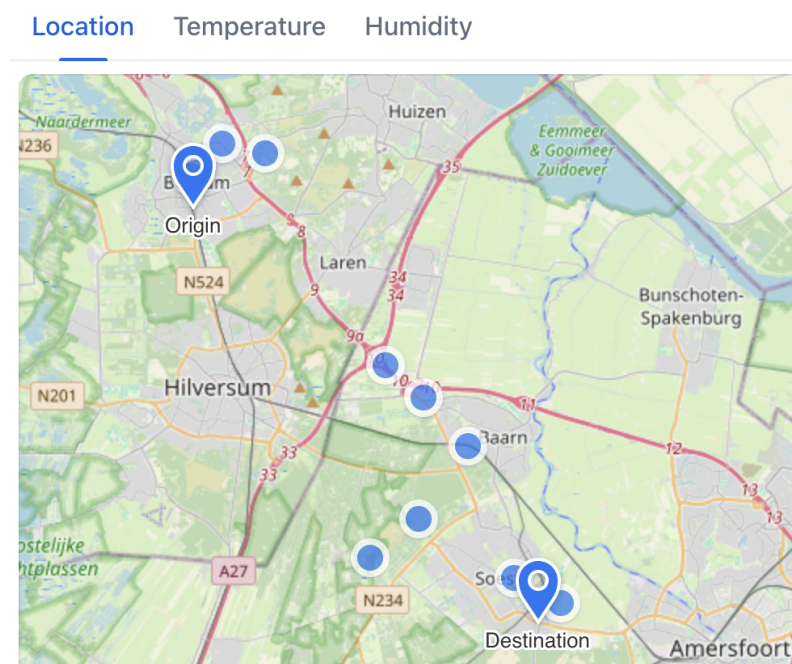
Após algumas iterações da validação supramencionada, o sistema foi considerado suficientemente estável para empregar os métodos de pesquisa da DSM para a avaliação da implementação e investigação do problema.

## 6.2 Estudo de Caso

O SBIdC-RALCF foi submetido à avaliação por meio de um mecanismo de estudo de caso, ou *Single-Case Mechanism* (WIERINGA, 2014), para aferir os efeitos causados pelo artefato no contexto da logística da cadeia de frio. O intuito principal desse mecanismo é garantir a recuperação da localização dos dispositivos sem o uso de GPS e a sua conectividade em áreas remotas, além de atestar a interoperabilidade dos protótipos e o desacoplamento da aplicação *web* dos servidores de rede. Diversos fretes com rotas distintas foram criados no SBIdC-RALCF e os seus roteiros foram seguidos com diferentes modos de transporte para simular a entrega de produtos perecíveis, para ambos os provedores.

Uma rota simulada entre Bussum e Soest (duas cidades nos Países Baixos) com o protótipo WiFi LoRa V3 contempla os resultados gerais obtidos com um dispositivo conectado ao servidor de rede KPN Things, pois contém trajetos com áreas urbana e rural e que requerem movimentações lenta e rápida. A plotagem dos dados de geolocalização recebidos para esta rota é ilustrada na Figura 34. Durante os experimentos, 90% dos pontos recebidos desviam menos de 100 metros da rota efetivamente tomada. Em áreas urbanas, esta acurácia diminuiu para até 300 metros devido à maior interferência e falta de LoS. A recepção de dados de geolocalização em trechos rápidos foi baixa ou inexistente, como neste caso entre as cidades de Bussum e Laren com uma velocidade de até 100 km/h. Os resultados para esta e outras rotas simuladas corroboram a expectativa de acurácia e desempenho dos algoritmos da KPN (KPN, 2024d).

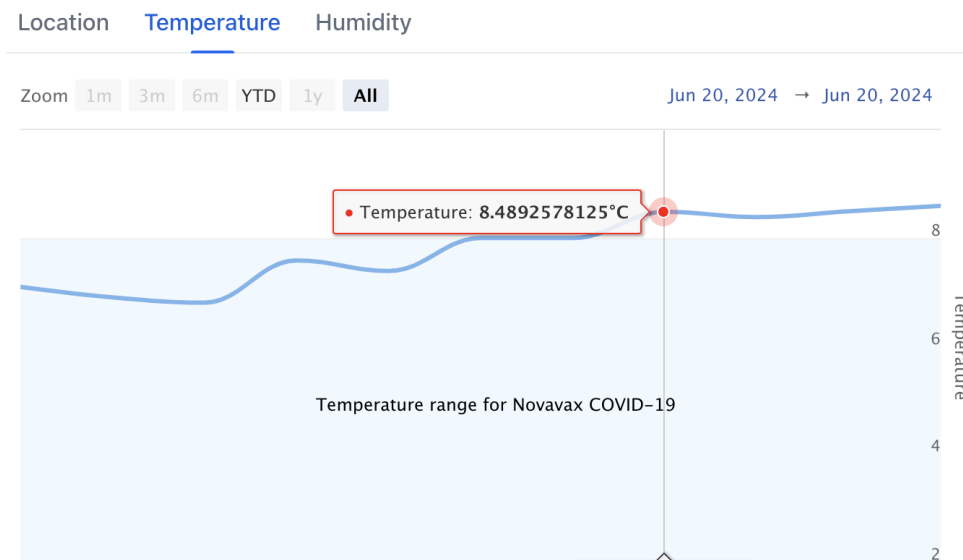
Figura 34 – Geolocalização recebida da KPN para rota entre Bussum e Soest



Fonte: Captura de tela do SBIdC-RALCF

As medições de temperatura para a mesma rota estão ilustradas na Figura 35, incluindo uma violação das condições de segurança do produto que foi intencionalmente causada durante o trajeto. O dispositivo recebeu a mensagem *downlink* condizente e acendeu o seu LED, conforme esperado. As medições foram recebidas ininterruptamente nos intervalos agendados e estavam disponíveis para visualização no SBIdC-RALCF em tempo real.

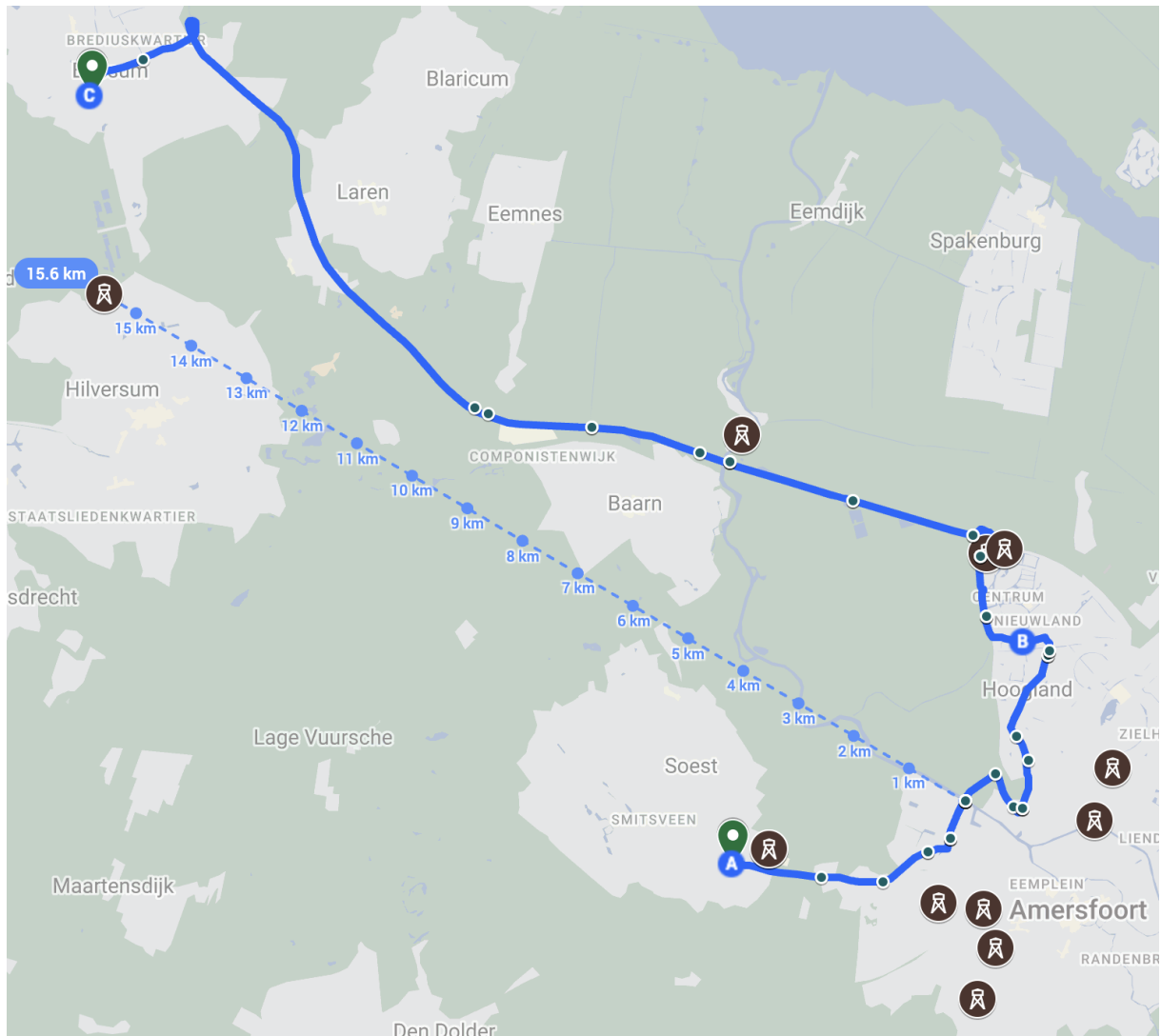
Figura 35 – Aferições de temperatura recebidas de dispositivo conectado à KPN em rota entre Bussum e Soest



Fonte: Captura de tela do SBIdC-RALCF

No portal da KPN Things, é possível identificar a localização dos gateways que receberam a mensagem dos dispositivos e, assim, calcular aproximadamente a distância entre eles. No entanto, como a área de cobertura dos gateways deste provedor é alta, a conectividade em áreas remotas com a rede TTN é o alvo neste estudo de caso. Como este servidor de rede não suporta geolocalização por padrão, o aplicativo “TTN Mapper” foi instalado em um *smartphone* com GPS e conectado ao *broker* MQTT do TTN, permitindo coletar todos os metadados das mensagens e determinar o local de início da transmissão e o gateway que as receberam. Esta configuração é detalhada no Apêndice H.3.

Figura 36 – Gateways e caminho de transmissão mais longo mapeados usando TTN em uma rota de Soest a Bussum



Fonte: Produzido pelo autor

As transmissões do protótipo NodeMCU v3 conectado à TTN são mostradas na Figura 36 para uma rota entre as cidades de Soest e Bussum. A linha grossa azul indica a rota efetiva tomada, os pontos sobre ela representam o local de início da transmissão das mensagens *uplink*, enquanto os gateways são retratados como ícones de torre. Ressalta-se que o gateway da torre do “Hilversum Media Park” recebeu vários pacotes, incluindo um a uma distância de 15,6 km a partir da periferia da cidade de Amersfoort. Isto evidencia o impacto do fator de LoS e a correlação com a potencial área de cobertura representada na Figura 12.

### 6.3 Experimento de escala

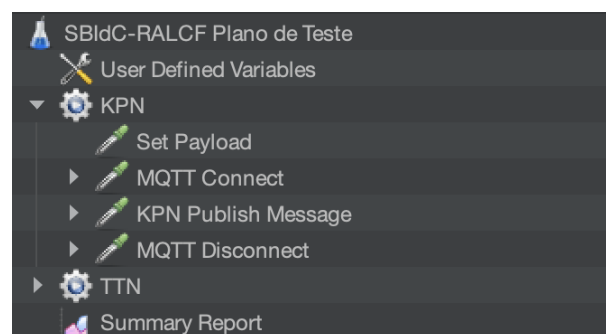
Uma vez validado o sistema no estudo de caso, o SBIdC-RALCF foi submetido à avaliação por meio de um experimento de escala, ou *Scale-Up Experiment* (WIERINGA, 2014), aplicando-se testes de carga para validar a robustez do artefato.

Sendo uma rede comunitária, o TTN não pode ser usado para testes de estresse. Ao mesmo tempo, o plano KPN *Freemium* suporta apenas até três dispositivos, embora o plano empresarial *Standard* suporte mais de mil dispositivos (KPN, 2024b). Por essas razões, nossa validação de desempenho é focada na aplicação LoRaWAN consumindo mensagens simuladas enviadas ao *broker* MQTT.

Um procedimento foi criado na classe `TestScenarioController` da aplicação *web* para auxiliar na configuração dos testes de carga. Ao ser invocado com os parâmetros `n` e `networkServer`, são criados `n` dispositivos do servidor de rede parametrizado e fretes associados que simulam a entrega de uma vacina com restrições de temperatura entre 2 e 8 °C. Como saída, um arquivo *comma-separated values* (CSV) é gerado contendo os identificadores dos dispositivos criados.

Os arquivos com os dispositivos de teste de cada servidor de rede são utilizados como entrada de um plano de teste Apache JMeter<sup>2</sup>, ilustrado na Figura 37. Neste plano, foram definidos 1000 dispositivos para cada servidor de rede, com 30 minutos de duração e um período de *ramp-up* de 60 segundos, respeitando o limite de taxa imposto pelo *broker* público EMQ. Cada dispositivo envia aproximadamente uma mensagem por minuto para o seu tópico MQTT específico. A carga útil enviada é uma medição fixa contendo os dados de temperatura e umidade, com uma chance de 2% de serem violadas as condições de temperatura, aleatoriamente acima de 8 °C.

Figura 37 – Plano de teste JMeter do SBIdC-RALCF



Fonte: Captura de tela da ferramenta JMeter

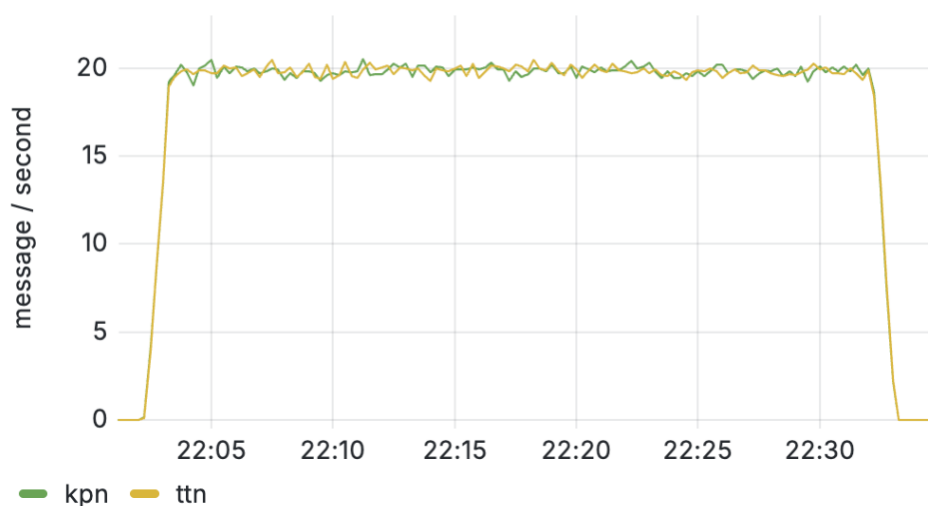
Para facilitar a validação do desempenho da aplicação, além de armazenar os dados recebidos dos nós finais, o SBIdC-RALCF expõe métricas usando o padrão OpenTeleme-

<sup>2</sup> <<https://jmeter.apache.org>>

try<sup>3</sup>. As ferramentas Prometheus<sup>4</sup> e Grafana<sup>5</sup> foram utilizadas para coletar e visualizar estas métricas, respectivamente. O Prometheus é um sistema de monitoramento de código aberto com linguagem de consulta flexível e banco de dados de séries temporais eficiente. O Grafana é uma aplicação *web* de código-fonte aberto que suporta a análise e visualização interativa destes dados temporais, possibilitando a produção de gráficos e tabelas, além da emissão de alertas.

Um painel Grafana é exibido na Figura 38 com a taxa de transferência de mensagens por segundo manipuladas para cada servidor de rede na aplicação SBIdC-RALCF após a execução do plano de teste. Os dados plotados são compatíveis com a taxa de transferência declarada nos resultados do teste JMeter, ou seja, 19,8 mensagens por segundo. Uma única instância do aplicativo atingiu essa taxa de transferência, processando a mensagem de medição em menos de dez milissegundos em média.

Figura 38 – Taxa de transferência de medições de dispositivos por servidor de rede



Fonte: Captura de tela da ferramenta Grafana

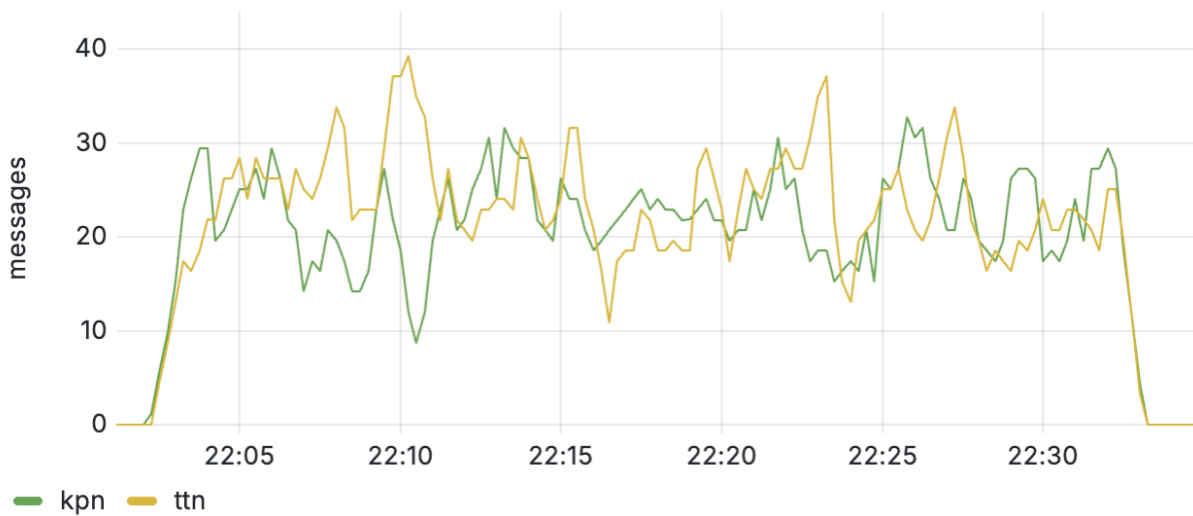
A métrica de mensagens violadas expostas pelo SBIdC-RALCF por servidor de rede é exibida na Figura 39. Considerando que a aplicação processou em torno de 2400 mensagens por minuto, é possível correlacionar a taxa de erro proposta no plano de testes com as 25 violações em média identificadas por minuto para cada servidor de rede.

<sup>3</sup> <<https://opentelemetry.io>>

<sup>4</sup> <<https://prometheus.io>>

<sup>5</sup> <<https://grafana.com>>

Figura 39 – Métrica de condição violada por servidor de rede



Fonte: Captura de tela da ferramenta Grafana

## 6.4 Considerações Finais

Este capítulo apresentou a validação e análise de desempenho do SBIdC-RALCF, demonstrando a viabilidade do sistema em contextos desafiadores, como a logística de cadeia de frio. O mecanismo de estudo de caso evidenciou que o protocolo LoRaWAN é eficiente para comunicação em longas distâncias, ao mesmo tempo que as funcionalidades do sistema, como a detecção de violações de condições e comunicação bidirecional, mostraram-se robustas nos cenários simulados. A geolocalização nativa da rede KPN Things mostrou-se uma alternativa viável para a localização dos dispositivos sem módulo GPS quando a precisão não é crítica para o contexto da aplicação.

Além disso, o experimento de escala validou a resiliência do SBIdC-RALCF ao lidar com cargas significativas, confirmando a compatibilidade da aplicação com padrões de desempenho de uso real. O uso de métricas expostas via OpenTelemetry e a integração com Prometheus e Grafana permitiram uma análise detalhada da taxa de transferência e tempo de processamento. Esses resultados reforçam o potencial do SBIdC-RALCF como um sistema que pode amenizar os problemas enfrentados na logística de produtos perecíveis.

---

# Capítulo 7

## Discussão

---

Wieringa (2014) declara que em um tratamento, a interação de um artefato em um contexto para tratar adversidades do mundo real, pode contribuir para a cura ou aliviar os sintomas de um problema, mas, em contrapartida, introduzir novos problemas. O SBIdC-RALCF fortalece esse enunciado, pois, embora os desafios encontrados no Ciclo de *Design* tenham sido superados para a validação da proposta, também foram introduzidas restrições e novas oportunidades.

As seções deste capítulo final apresentam uma análise das QCs e do PPT, as limitações da abordagem proposta e os trabalhos futuros sugeridos que podem tornar o SBIdC-RALCF mais efetivo no tratamento dos problemas enfrentados no contexto da logística da cadeia de frio.

### 7.1 Análise das Questões de Pesquisa

A Seção 2.2 apresentou as principais Questões de Conhecimento que nortearam a elaboração do Problema de Pesquisa Técnica abordado nesta dissertação. Aqui são apresentadas as sínteses de suas respostas obtidas através de revisões abrangentes e sistemáticas da literatura, além do impacto que elas causaram nesta pesquisa de mestrado.

*QC1: Quais são os domínios de aplicação mais relevantes da IdC?*

Os resultados da RSL discutida na seção 3.1 permitiram identificar que o domínio de Automação Residencial/Smart Home possui alta relevância e crescente interesse na comunidade acadêmica. Não obstante, Saúde, Cidades Inteligentes e Sustentáveis/*Smart City*, Setor Energético/*Smart Grid*, Indústria 4.0, Identificação Inteligente, Logística e SCM,

Agricultura e Meio Ambiente são domínios com forte presença de pesquisa e consequente relevância na área de IdC nos últimos cinco anos.

Com a pandemia da COVID-19 no plano de fundo deste projeto de mestrado, a problemática da distribuição e conservação de vacinas nos direcionou aos domínios de logística e saúde, culminando na investigação da IdC na cadeia de frio, tema das duas próximas QC.

**QC2:** *Quais são as principais tecnologias de IdC utilizadas na logística da cadeia de frio?*

A resposta desta QC provém da fase de análise dos resultados da RSL discutida na seção 3.2. Entre os dispositivos de IdC, destaca-se o uso de sensores de temperatura em todos os estudos, enquanto umidade é uma preocupação secundária, aparecendo em torno de 61% dos estudos e, por último, luminosidade, em pouco mais de 30%. Os dados coletados são usualmente processados *in loco* pelas placas, destacando-se Arduino e Raspberry Pi, plataformas muito populares para o desenvolvimento de aplicações de Internet das Coisas (POLIANYTSIA; STARKOVA; HERASYMENKO, 2016). Na camada de borda, a comunicação sem fio é feita para dispositivos que agem como gateway e, na maioria dos casos, diretamente para a nuvem. As redes móveis GSM/3G/4G são comumente adotadas por todo o período pesquisado. SigFox e LoRaWAN são tecnologias de última geração empregadas para comunicação de longa distância. Enquanto isso, RFID, Bluetooth/BLE e ZigBee são usados para comunicação de curto alcance. Os serviços em nuvem, como “IBM Watson IoT” e “Azure IoT Hub”, oferecem processamento e análise de dados adicionais, suportando diferentes protocolos de aplicação para integração tais quais HTTPS e MQTT.

**QC3:** *Como estas tecnologias de IdC são aplicadas em áreas remotas?*

A problemática das áreas remotas também foi alvo da RSL sintetizada na seção 3.2. Quatro artigos citam explicitamente os principais desafios enfrentados, que podem ser categorizados como (a) falta de conectividade de rede e (b) interrupções de energia. Em um estudo da Albânia, a solução depende da bateria do celular. Já outro, de Bangladesh, concentra-se no baixo consumo de energia do protótipo proposto, bem como sua capacidade de armazenamento local para superar problemas de energia. Um estudo realizado na África do Sul destaca-se pelo uso de painéis solares como fonte de energia para sua solução de monitoramento da caixa de vacinas. Por fim, um estudo proveniente do Congo também aplica energia solar, mas vai além e experimenta com LoRaWAN para criar uma sequência de dispositivos sem fio para formar redes *mesh* desconectadas da Internet.

Apesar do estudo citar explicitamente problemas de conectividade com a rede LoRaWAN em seu cenário, essa tecnologia LPWAN se sobressaiu perante as demais con-

siderando a problemática da distribuição de bens em áreas remotas e as características ressaltadas pelos autores. Isto engatilhou a próxima RSL para esclarecer a QC a seguir.

**QC4:** *Como aplicações LoRaWAN são utilizadas na logística e na cadeia de frio?*

Os estudos prospectados de pesquisas exploratório e sistemática, sendo os principais detalhados na seção 3.3, mostram que as aplicações baseadas em LoRaWAN têm se mostrado valiosas na logística e cadeia de frio devido à sua capacidade de comunicação bidirecional e de longo alcance com baixo consumo de energia. As aplicações em evidência validam a comunicação com nós estacionários, a penetração das CSS em ambientes urbanos com interferência, a característica bidirecional que habilita funcionalidades como a emissão de alertas, a possibilidade de criação de redes privadas com gateways próprios, além da funcionalidade de geolocalização sem GPS. Esses estudos demonstram a adequação do protocolo para atender a demandas específicas da cadeia de frio, como a manutenção de condições ideais e o rastreamento em tempo real, mesmo em cenários com infraestrutura limitada.

Durante o percurso deste projeto de mestrado, o esclarecimento destas QCs possibilitou elaborar o Problema de Pesquisa Técnica e validar a Teoria de Projeto de que um sistema como o SBIdC-RALCF, que suporte múltiplos servidores de rede LoRaWAN, pode:

- a) melhorar a cobertura e redundância da comunicação de rede em áreas remotas devido à interoperabilidade dos dispositivos e desacoplamento da aplicação *web*;
- b) possibilitar o rastreo dos produtos com baixo custo através do recurso de geolocalização por multilateração;
- c) permitir detectar e notificar violações de condições cruciais, sem a necessidade de conectividade do nó final com a Internet.

Espera-se que esta abordagem traga uma nova perspectiva para tomadores de decisões ao estabelecer contratos custo-efetivos, que pode mitigar desperdícios e erros vitais como no contexto da cadeia de frio, mas também em outros casos de uso.

## 7.2 Limitações

Esta seção descreve os desafios encontrados durante o *design* e a implementação do SBIdC-RALCF. Iniciando pela camada de dispositivos, embora a biblioteca LMIC implemente e abstraia a maioria das particularidades do protocolo LoRaWAN para os nós finais, encontrar a configuração correta de mapeamento de pinos para ambas as placas exigiu considerável investigação. Além disso, após conectar o módulo LoRa SX1276 ao NodeMCU v3, não restaram muitos pinos GPIO funcionais para o próprio sensor DHT22,

dificultando o seu mapeamento. Esta limitação deve ser considerada antecipadamente ao se escolher a placa e os sensores necessários de acordo com o projeto que se deseja desenvolver.

Em relação aos servidores de rede, a rede TTN não fornece o recurso de geolocalização por multilateração pronto para uso, pois depende dos gateways implantados pelos usuários da comunidade. Essa insuficiência foi suprida durante o estudo de caso usando o aplicativo TTN Mapper e um dispositivo móvel com GPS conectado à Internet. No entanto, para a implantação de uma rede privada LoRaWAN ao invés da TTN, transceptores LoRa como o LR1110 (The Things Network, 2024b) e gateways com concentradores adequados como o SX1303 (The Things Network, 2024a) devem ser considerados para habilitar o recurso de geolocalização.

Outra limitação da TTN é que o *broker* MQTT versão 3.1.1 suporta apenas o nível de QoS 0, o que significa que não há garantia de que as mensagens sejam entregues. Além disso, esta versão não suporta o recurso de assinatura compartilhada do MQTT 5, que garante que o *broker* envie uma mensagem para apenas um cliente em um grupo de assinantes, melhorando o balanceamento de carga e a tolerância a falhas ao consumir as mensagens de um tópico pelo microsserviço do SBIdC-RALCF. Atualmente podem ocorrer problemas de concorrência com duas ou mais instâncias da aplicação *web* em execução processando as mesmas mensagens provenientes do servidor de rede. Para corrigir essa limitação, a integração via *webhook* da TTN poderia encaminhar mensagens para um *broker* próprio que suporte a versão MQTT 5.

No experimento de escala da etapa de validação, os dispositivos finais são ignorados no teste de carga, pois este envia mensagens simuladas diretamente para os *brokers* MQTT. Esta limitação impede que o experimento de escala cubra todos os componentes da arquitetura do SBIdC-RALCF. Com os recursos financeiros adequados, idealmente um *testbed* deve ser empregado considerando a camada de dispositivos do projeto (MARAIIS; MALEKIAN; ABU-MAHFOUZ, 2017).

### 7.3 Trabalhos Futuros

O estudo de caso desse projeto de mestrado foi realizado nos Países Baixos, onde a topografia majoritariamente plana e a infraestrutura existente proporcionam condições favoráveis para a utilização do protocolo LoRaWAN. Embora tenha sido possível validar a comunicação à longa distância com interoperabilidade de servidores de rede, recomenda-se como continuidade deste estudo o uso do SBIdC-RALCF em um problema concreto em áreas remotas aplicando-se métodos de pesquisa como Pesquisa-ação técnica, ou *Technical action research* (TAR), um dos últimos estágios para escalar uma tecnologia do laboratório para o mundo real (WIERINGA, 2014).

A distribuição de soro antiofídico na região amazônica detalhada por Fan e Monteiro

(2018) é um exemplo concreto do problema da cadeia de frio no mundo real. Primeiramente, o mapeamento da área de cobertura de antenas poderia ser realizado com ferramentas de simulação como a “Radio Mobile Online” (Roger Coudé VE2DBE, 2024). Esta abordagem pode ser adotada para a implantação de gateways próprios em uma rede LoRaWAN privada como a ChirpStack. Estes gateways estrategicamente posicionados podem cobrir áreas com ausência ou intermitência de cobertura de rede, de modo a permitir validar a viabilidade do sistema em ambientes com condições próximas da realidade das áreas remotas.

Os recursos implementados no Ciclo de *Design* do SBIdC-RALCF são suficientes para a validação do tratamento proposto. No entanto, antes do sistema avançar no Ciclo de Engenharia (WIERINGA, 2014), há diversas oportunidades para seu aprimoramento que podem trazer ganhos significativos em eficiência, confiabilidade e usabilidade. Uma sugestão é melhorar a validação do experimento de escala utilizando uma plataforma de *testbed* sobre os dispositivos LoRaWAN (MARAIS; MALEKIAN; ABU-MAHFOUZ, 2017). Outro exemplo é a automação do registro de dispositivos na plataforma utilizando tecnologias como *Near Field Communication* (NFC) e código *Quick-Response* (QR). Essa funcionalidade eliminaria processos manuais, reduzindo erros humanos e abstraindo o cadastro de novos dispositivos nos servidores de rede, além de oferecer uma experiência mais intuitiva para os atores da aplicação.

Outra melhoria seria o envio imediato dos limites de violação cadastrados na aplicação *web* para os dispositivos finais assim que estes se conectassem ao servidor de rede LoRaWAN. O *firmware* deveria ser evoluído e para conter a capacidade básica de identificar violações localmente, sem a necessidade de aguardar eventos *downlink* para notificar tal ocorrência ao entregador. Este recurso manteria a centralização dos parâmetros operacionais ao passo que aumentaria a eficiência da comunicação bidirecional e confiabilidade da funcionalidade de detecção de violações. Ainda no âmbito da validação das condições, permitir que o administrador customize a função de verificação utilizando os limites como parâmetros poderia aumentar a flexibilidade da aplicação.

Os protótipos do SBIdC-RALCF possuem capacidades que não foram explorados no escopo deste projeto de mestrado, como o armazenamento local. Manter os limites operacionais armazenados traria benefícios significativos em situações de falha de comunicação. Essa abordagem aumentaria a confiabilidade do sistema, permitindo que os dispositivos continuem operando mesmo sem conectividade com o servidor. Além disso, ela reduziria a dependência de comunicação em tempo real, o que é crucial para aplicações em locais remotos ou com infraestrutura de rede limitada.

Este trabalho utilizou os recursos de segurança nativos dos protocolos, como o uso de chaves nas integrações por API via HTTPS e a comunicação encriptada AES do LoRaWAN. No entanto, um estudo aprofundado poderia ser realizado para melhorar a segurança do sistema considerando vulnerabilidades conhecidas deste protocolo de rede

LPWAN para a prevenção de ataques (NOURA et al., 2020).

Por fim, plataformas de Inteligência Artificial (IA) e *Big Data* poderiam ser utilizadas na implementação de funcionalidades preditivas para identificar situações em que as condições operacionais se aproximam dos limites evitando violações de forma proativa. Essa capacidade traria vantagens como a prevenção de falhas, a otimização de recursos e a maior segurança em operações críticas da cadeia de frio. A previsão antecipada de problemas permitiria que medidas corretivas fossem tomadas antes que impactos negativos ocorram.

## 7.4 Considerações Finais

Este capítulo apresentou os desafios enfrentados durante o Ciclo de *Design* do SBIdC-RALCF, as limitações atreladas às decisões de projeto tomadas e trabalhos futuros. As sugestões de aprimoramentos tornariam o tratamento mais preparado para atender a cenários do mundo real e ser considerado para um Ciclo de Engenharia. A automação de processos, a capacidade de operação *offline* e as funcionalidades preditivas não apenas melhorariam a experiência do usuário, mas também contribuiriam para a confiabilidade do sistema, ampliando seu alcance e impacto potencial.

---

## Conclusão

---

Este projeto de mestrado apresentou o projeto, a implementação e a validação do SBIdC-RALCF, uma aplicação LoRaWAN que suporta múltiplos servidores de rede, projetada para mitigar os desafios da logística da cadeia de frio. O estudo mostra como vários servidores de rede, aspecto mais valioso desta proposta, podem ser usados na mesma solução com configurações mínimas de parâmetros no dispositivo final devido ao mecanismo de ativação OTAA e à interoperabilidade do protocolo LoRaWAN. Além disso, um microsserviço desacoplado e escalável abstrai a fonte das informações do dispositivo por meio de múltiplos clientes assinantes MQTT, abordagem ratificada por um estudo de caso com cenários em movimento em áreas urbana e rural, além de experimento de escala por meio de teste de carga.

Esta configuração com vários servidores de rede oferece novas perspectivas em vários domínios onde a comunicação de longo alcance e baixa potência se destaca, aumentando a redundância e confiabilidade do sistema. Ela permite que novas aplicações de IdC sejam adicionadas ao se investir em infraestrutura de redes privadas LoRaWAN custo-efetivas, enquanto a infraestrutura de telecomunicação existente é reaproveitada, maximizando o ROI de novas soluções.

Além da estratégia com múltiplos provedores, essa dissertação mostrou como as ferramentas de simulação podem oferecer um equilíbrio adequado entre a cobertura das antenas e as transmissões de modulação LoRa recebidas pelos gateways dos servidores de rede. Este ensaio pode ser empregado na implantação de gateways em uma rede privada como uma alternativa às tentativas *ad-hoc* (LAWRENCE et al., 2018).

Adicionalmente, este estudo reforça, por meio de um estudo de caso em áreas urbana e rural com o servidor de rede KPN Things, como o recurso de geolocalização por multilateração sem GPS do protocolo LoRaWAN é uma alternativa acessível caso uma localização altamente precisa não seja necessária. Isto significa que investir mais na implantação do gateway pode compensar em comparação aos custos de dispositivos finais de alta potência com módulos GPS.

Seguindo as melhores práticas de desenvolvimento orientado a objetos, a aplicação é adequada e extensível para diferentes provedores de rede, tipos de medição e decodificadores de carga útil para trabalhar com os recursos de detecção do dispositivo final. Assim, podemos generalizar essa abordagem para construir soluções para outros domínios além da cadeia de frio, especialmente em áreas remotas onde as redes intermitentes ainda são uma realidade.

O projeto e desenvolvimento do SBIdC-RALCF foi tema do manuscrito *A LoRaWAN Multi-Network Server Application for Smart Cold Chain Tracking in Remote Areas*, aceito como artigo regular pela conferência *27th International Conference on Enterprise Information Systems (ICEIS 2025)*. O Anexo B contém a íntegra da sua versão *camera-ready*.

---

# Referências

---

- ALKHOORI, O. et al. Design and implementation of CryptoCargo: A blockchain-powered smart shipping container for vaccine distribution. **IEEE Access**, IEEE, v. 9, p. 53786–53803, 2021.
- ASGHARI, P.; RAHMANI, A. M.; JAVADI, H. H. S. Internet of Things applications: A systematic review. **Computer Networks**, Elsevier, v. 148, p. 241–261, 2019.
- ASHTON, K. et al. That ‘internet of things’ thing. **RFID journal**, Hauppauge, New York, v. 22, n. 7, p. 97–114, 2009.
- AUGUSTIN, A. et al. A study of LoRa: Long range & low power networks for the Internet of Things. **Sensors**, MDPI, v. 16, n. 9, p. 1466, 2016.
- BADAMASI, Y. A. The working principle of an Arduino. In: IEEE. **2014 11th international conference on electronics, computer and computation (ICECCO)**. [S.l.], 2014. p. 1–4.
- BADIA-MELIS, R. et al. New trends in cold chain monitoring applications-A review. **Food Control**, Elsevier, v. 86, p. 170–182, 2018.
- BELSHE, M.; PEON, R.; THOMSON, M. **Hypertext transfer protocol version 2 (HTTP/2)**. [S.l.], 2015.
- BISHOP, M. **RFC 9114: HTTP/3**. [S.l.]: RFC Editor, 2022.
- BOSE, A.; AITHAL, A. K.; RAJESHWARI, B. Vaccine cold storage monitoring and tracking using LoRaWAN. In: IEEE. **2022 2nd International Conference on Intelligent Technologies (CONIT)**. [S.l.], 2022. p. 1–6.
- CATTANI, M.; BOANO, C. A.; RÖMER, K. An experimental evaluation of the reliability of LoRa long-range low-power wireless communication. **Journal of Sensor and Actuator Networks**, MDPI, v. 6, n. 2, p. 7, 2017.
- CAVALCANTE, A. M. et al. 5G for remote areas: Challenges, opportunities and business modeling for Brazil. **IEEE Access**, IEEE, v. 9, p. 10829–10843, 2021.
- CENTENARO, M. et al. Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios. **IEEE Wireless Communications**, IEEE, v. 23, n. 5, p. 60–67, 2016.

- CHOI, W. et al. Smart home and internet of things: A bibliometric study. **Journal of Cleaner Production**, Elsevier, v. 301, p. 126908, 2021.
- CROW, B. P. et al. IEEE 802.11 wireless local area networks. **IEEE Communications magazine**, IEEE, v. 35, n. 9, p. 116–126, 1997.
- DING, J. et al. IoT connectivity technologies and applications: A survey. **IEEE Access**, IEEE, v. 8, p. 67646–67673, 2020.
- EMQ Technologies Inc. **Free Public MQTT Broker**. 2024. Disponível em: <<https://www.emqx.com/en/mqtt/public-mqtt5-broker>>.
- ENRIKO, I.; ALEMUDA, F.; ADRIANTO, D. An Approach to Monitor Vaccine Quality During Distribution Using Internet of Things. **Journal of Information Science & Engineering**, v. 38, n. 5, 2022.
- ERGEN, S. C. ZigBee/IEEE 802.15. 4 Summary. **UC Berkeley, September**, v. 10, n. 17, p. 11, 2004.
- FAN, H. W.; MONTEIRO, W. M. History and perspectives on how to ensure antivenom accessibility in the most remote areas in Brazil. **Toxicon**, Elsevier, v. 151, p. 15–23, 2018.
- FARGAS, B. C.; PETERSEN, M. N. GPS-free geolocation using LoRa in low-power WANs. In: IEEE. **2017 global internet of things summit (Giots)**. [S.l.], 2017. p. 1–6.
- FIELDING, R. et al. **Hypertext transfer protocol–HTTP/1.1**. [S.l.], 1999.
- FRANGOUDIS, P. A.; TSIGKANOS, C.; DUSTDAR, S. Connectivity technology selection and deployment strategies for IoT service provision over LPWAN. **IEEE Internet Computing**, IEEE, v. 25, n. 1, p. 61–70, 2021.
- GARCIA, A. F.; SOUZA, W. L. de. Internet of Things Applications for Cold Chain Vaccine Tracking: A Systematic Literature Review. In: SPRINGER. **International Conference on Information Technology-New Generations**. [S.l.], 2023. p. 323–330.
- GARRIGA, M. et al. RESTful service composition at a glance: A survey. **Journal of Network and Computer Applications**, Elsevier, v. 60, p. 32–53, 2016.
- Google Maps Platform. **Geolocation API Overview**. 2024. Disponível em: <<https://developers.google.com/maps/documentation/geolocation/overview>>.
- HASANAT, R. et al. Development of a monitoring system and power management for an IoT based vaccine carrier. In: IOP PUBLISHING. **Journal of Physics: Conference Series**. [S.l.], 2021. v. 1755, n. 1, p. 012023.
- HASANAT, R. T. et al. An IoT based real-time data-centric monitoring system for vaccine cold chain. In: IEEE. **2020 IEEE East-West Design & Test Symposium (EWDTS)**. [S.l.], 2020. p. 1–5.
- HASSIJA, V. et al. A survey on IoT security: application areas, security threats, and solution architectures. **IEEE Access**, IEEE, v. 7, p. 82721–82743, 2019.

JENNINGS, C. et al. **Sensor measurement lists (SenML)**. [S.l.], 2018.

KASSAB, W.; DARABKH, K. A. A-Z survey of Internet of Things: Architectures, protocols, applications, recent advances, future directions and recommendations. **Journal of Network and Computer Applications**, Elsevier, v. 163, p. 102663, 2020.

KONDAVEETI, H. K. et al. A systematic literature review on prototyping with Arduino: Applications, challenges, advantages, and limitations. **Computer Science Review**, Elsevier, v. 40, p. 100364, 2021.

KPN. **KPN Things Developer Manual - API Access**. 2024. Disponível em: <<https://docs.kpnthings.com/dm/concepts/api>>.

\_\_\_\_\_. **KPN Things Explorer**. 2024. Disponível em: <<https://www.kpn.com/zakelijk/internet-of-things/kpn-things/explorer.htm>>.

\_\_\_\_\_. **KPN Things LoRa Geolocation**. 2024. Disponível em: <<https://docs.kpnthings.com/lora/geolocation/intro>>.

\_\_\_\_\_. **KPN Things LoRa Performance**. 2024. Disponível em: <<https://docs.kpnthings.com/lora/geolocation/performance>>.

\_\_\_\_\_. **KPN Things LoRa Technical Background**. 2024. Disponível em: <<https://docs.kpnthings.com/lora/geolocation/technical-background>>.

LABDAOUI, N.; NOUVEL, F.; DUTERTRE, S. Energy-efficient IoT Communications: A Comparative Study of Long-Term Evolution for Machines (LTE-M) and Narrowband Internet of Things (NB-IoT) Technologies. In: IEEE COMPUTER SOCIETY. **2023 IEEE Symposium on Computers and Communications (ISCC)**. [S.l.], 2023. p. 823–830.

LAWRENCE, P. W. et al. Developing the IoT to support the health sector: A case study from kikwit, DR congo. In: SPRINGER. **International Conference on Emerging Technologies for Developing Countries**. [S.l.], 2017. p. 45–56.

\_\_\_\_\_. Developing the IoT to support the health sector: A case study from Kikwit, DR Congo. In: SPRINGER. **Emerging Technologies for Developing Countries: First International EAI Conference, AFRICATEK 2017, Marrakech, Morocco, March 27-28, 2017 Proceedings 1st**. [S.l.], 2018. p. 45–56.

LITAYEM, N.; AL-SA'DI, A. Exploring the Programming Model, Security Vulnerabilities, and Usability of ESP8266 and ESP32 Platforms for IoT Development. In: IEEE. **2023 IEEE 3rd International Conference on Computer Systems (ICCS)**. [S.l.], 2023. p. 150–157.

MARAIS, J. M.; MALEKIAN, R.; ABU-MAHFOUZ, A. M. LoRa and LoRaWAN testbeds: A review. **2017 Ieee Africon**, IEEE, p. 1496–1501, 2017.

MISHRA, B.; KERTESZ, A. The use of MQTT in M2M and IoT systems: A survey. **Ieee Access**, IEEE, v. 8, p. 201071–201086, 2020.

MULLIGAN, G. The 6LoWPAN architecture. In: **Proceedings of the 4th workshop on Embedded networked sensors**. [S.l.: s.n.], 2007. p. 78–82.

- NIKOLOV, N. Research of MQTT, CoAP, HTTP and XMPP IoT communication protocols for embedded systems. In: IEEE. **2020 XXIX International Scientific Conference Electronics (ET)**. [S.l.], 2020. p. 1–4.
- NOURA, H. et al. LoRaWAN security survey: Issues, threats and possible mitigation techniques. **Internet of Things**, Elsevier, v. 12, p. 100303, 2020.
- ORGANIZATION, W. H. et al. **Controlled temperature chain: strategic roadmap for priority vaccines 2017-2020**. [S.l.], 2017.
- \_\_\_\_\_. **WHO Director-General’s opening remarks at the media briefing on COVID-19-11 March 2020**. [S.l.]: Geneva, Switzerland, 2020.
- PEZOA, F. et al. Foundations of JSON schema. In: **Proceedings of the 25th international conference on World Wide Web**. [S.l.: s.n.], 2016. p. 263–273.
- POLIANYTSIA, A.; STARKOVA, O.; HERASYMENKO, K. Survey of hardware IoT platforms. In: IEEE. **2016 Third International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T)**. [S.l.], 2016. p. 152–153.
- QIU, T. et al. Edge computing in industrial internet of things: Architecture, advances and challenges. **IEEE Communications Surveys & Tutorials**, IEEE, v. 22, n. 4, p. 2462–2488, 2020.
- RICHARDSON, L.; RUBY, S. **RESTful web services**. [S.l.]: "O’Reilly Media, Inc.", 2008.
- Roger Coudé VE2DBE. **Radio Mobile Online**. 2024. Disponível em: <[https://www.ve2dbe.com/rmonline\\_s.asp](https://www.ve2dbe.com/rmonline_s.asp)>.
- SABAN, M. et al. Experimental analysis of IoT networks based on LoRa/LoRaWAN under indoor and outdoor environments: Performance and limitations. **IFAC-PapersOnLine**, Elsevier, v. 54, n. 4, p. 159–164, 2021.
- SERGI, I. et al. A smart and secure logistics system based on IoT and cloud technologies. **Sensors**, MDPI, v. 21, n. 6, p. 2231, 2021.
- The Things Industries. **The Things Stack API Authentication**. 2024. Disponível em: <<https://www.thethingsindustries.com/docs/api/concepts/auth/>>.
- \_\_\_\_\_. **The Things Stack HTTP (REST) API**. 2024. Disponível em: <<https://www.thethingsindustries.com/docs/api/reference/http/routes/>>.
- \_\_\_\_\_. **The Things Stack MQTT Server**. 2024. Disponível em: <<https://www.thethingsindustries.com/docs/integrations/mqtt/>>.
- The Things Network. **LoRaWAN Concentrators The Things Network**. 2024. Disponível em: <<https://www.thethingsnetwork.org/docs/lorawan/concentrators/>>.
- \_\_\_\_\_. **LoRaWAN Transceivers The Things Network**. 2024. Disponível em: <<https://www.thethingsnetwork.org/docs/lorawan/transceivers/>>.

- TREVISAN, M. et al. Measuring HTTP/3: Adoption and performance. In: IEEE. **2021 19th Mediterranean Communication and Computer Networking Conference (MedComNet)**. [S.l.], 2021. p. 1–8.
- URBANO, O. et al. Cost-effective implementation of a temperature traceability system based on smart RFID tags and IoT services. **Sensors**, MDPI, v. 20, n. 4, p. 1163, 2020.
- VANGELISTA, L.; ZANELLA, A.; ZORZI, M. Long-range IoT technologies: The dawn of LoRa™. In: SPRINGER. **Future Access Enablers for Ubiquitous and Intelligent Infrastructures: First International Conference, FABULOUS 2015, Ohrid, Republic of Macedonia, September 23-25, 2015. Revised Selected Papers 1**. [S.l.], 2015. p. 51–58.
- WANG, J. et al. The evolution of the Internet of Things (IoT) over the past 20 years. **Computers & Industrial Engineering**, Elsevier, v. 155, p. 107174, 2021.
- WEINSTEIN, R. RFID: a technical overview and its application to the enterprise. **IT professional**, IEEE, v. 7, n. 3, p. 27–33, 2005.
- WIERINGA, R. J. **Design science methodology for information systems and software engineering**. [S.l.]: Springer, 2014.
- WOUTERS, O. J. et al. Challenges in ensuring global access to COVID-19 vaccines: production, affordability, allocation, and deployment. **The Lancet**, Elsevier, v. 397, n. 10278, p. 1023–1034, 2021.
- YAN, Y. et al. The COVID-19 vaccines: recent development, challenges and prospects. **Vaccines**, MDPI, v. 9, n. 4, p. 349, 2021.
- YOKOTANI, T.; SASAKI, Y. Comparison with HTTP and MQTT on required network resources for IoT. In: IEEE. **2016 international conference on control, electronics, renewable energy and communications (ICCEREC)**. [S.l.], 2016. p. 1–6.
- YU, Y. B. et al. Grand challenges in pharmaceutical research series: ridding the cold chain for biologics. **Pharmaceutical Research**, Springer, v. 38, p. 3–7, 2021.
- ZHENG, C. et al. Real-world effectiveness of COVID-19 vaccines: a literature review and meta-analysis. **International Journal of Infectious Diseases**, Elsevier, v. 114, p. 252–260, 2022.
- ZIMMERMANN, H. OSI reference model-the ISO model of architecture for open systems interconnection. **IEEE Transactions on communications**, IEEE, v. 28, n. 4, p. 425–432, 2003.
- ZINAS, N. et al. Proposed open source architecture for Long Range monitoring. the case study of cattle tracking at Pogoniani. In: **Proceedings of the 21st Pan-Hellenic Conference on Informatics**. [S.l.: s.n.], 2017. p. 1–6.



# Apêndices



---

# APÊNDICE A

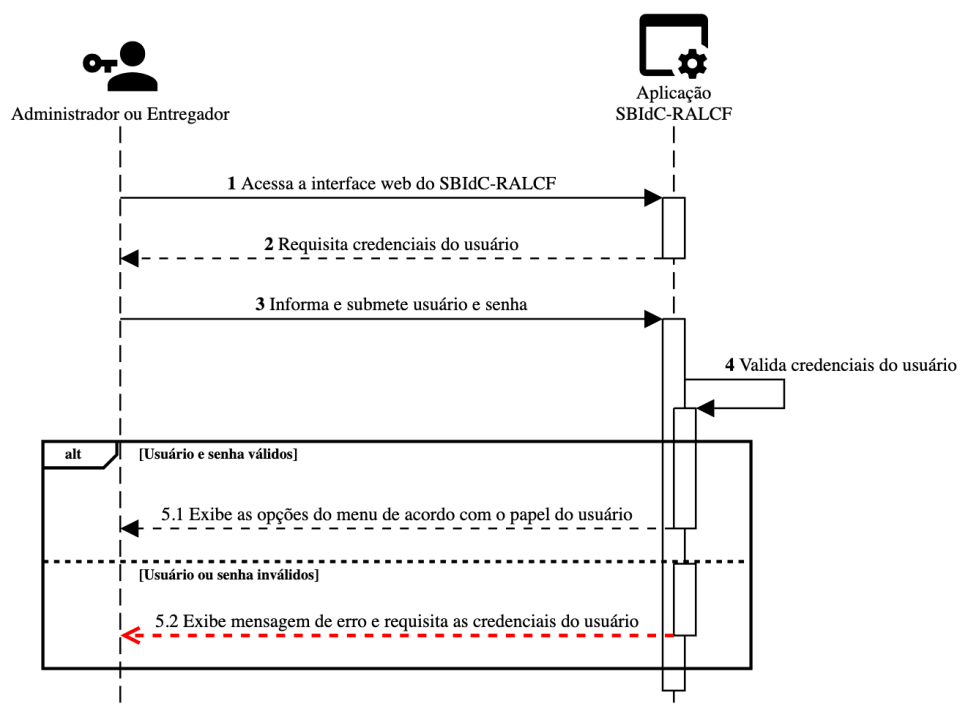
## Diagramas de Sequência

---

As seções deste apêndice apresentam os diagramas de sequência das principais funcionalidades do SBIdC-RALCF.

### A.1 Login do Usuário

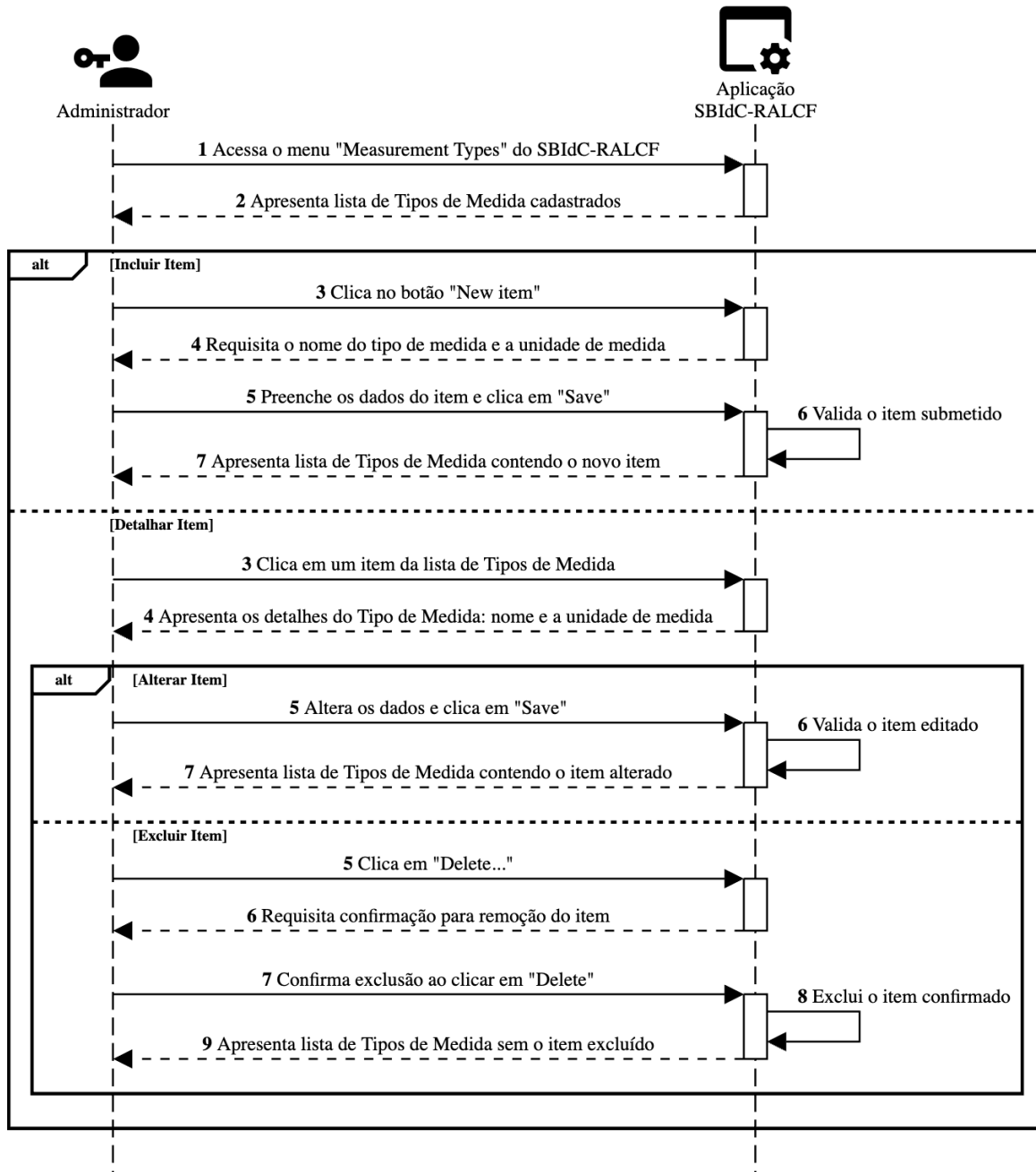
Figura 40 – Diagrama de sequência da funcionalidade “Login”



Fonte: Elaborado pelo autor

## A.2 CRUD de Tipos de Medida

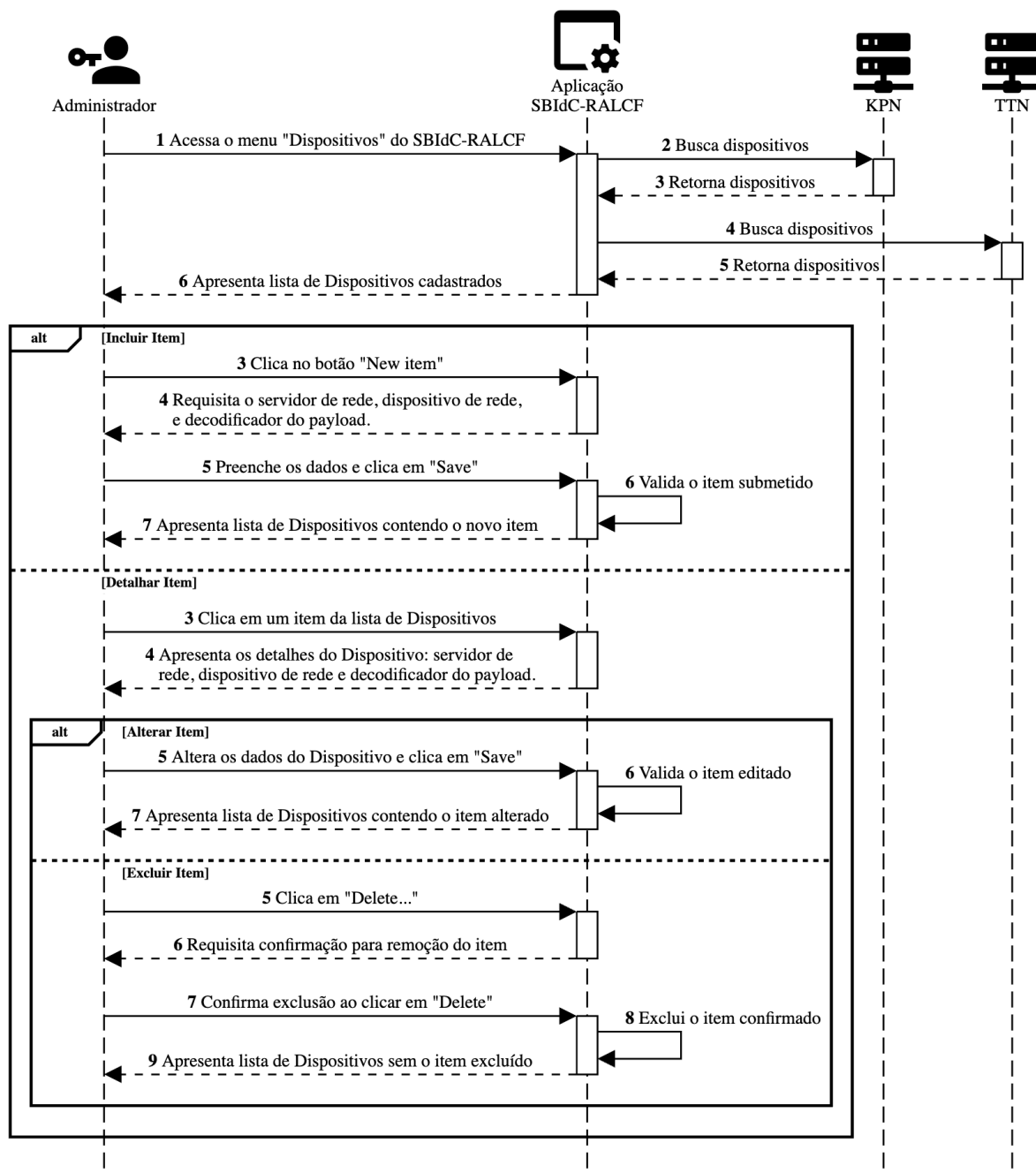
Figura 41 – Diagrama de sequência do CRUD de Tipos de Medida



Fonte: Elaborado pelo autor

## A.3 CRUD de Dispositivos

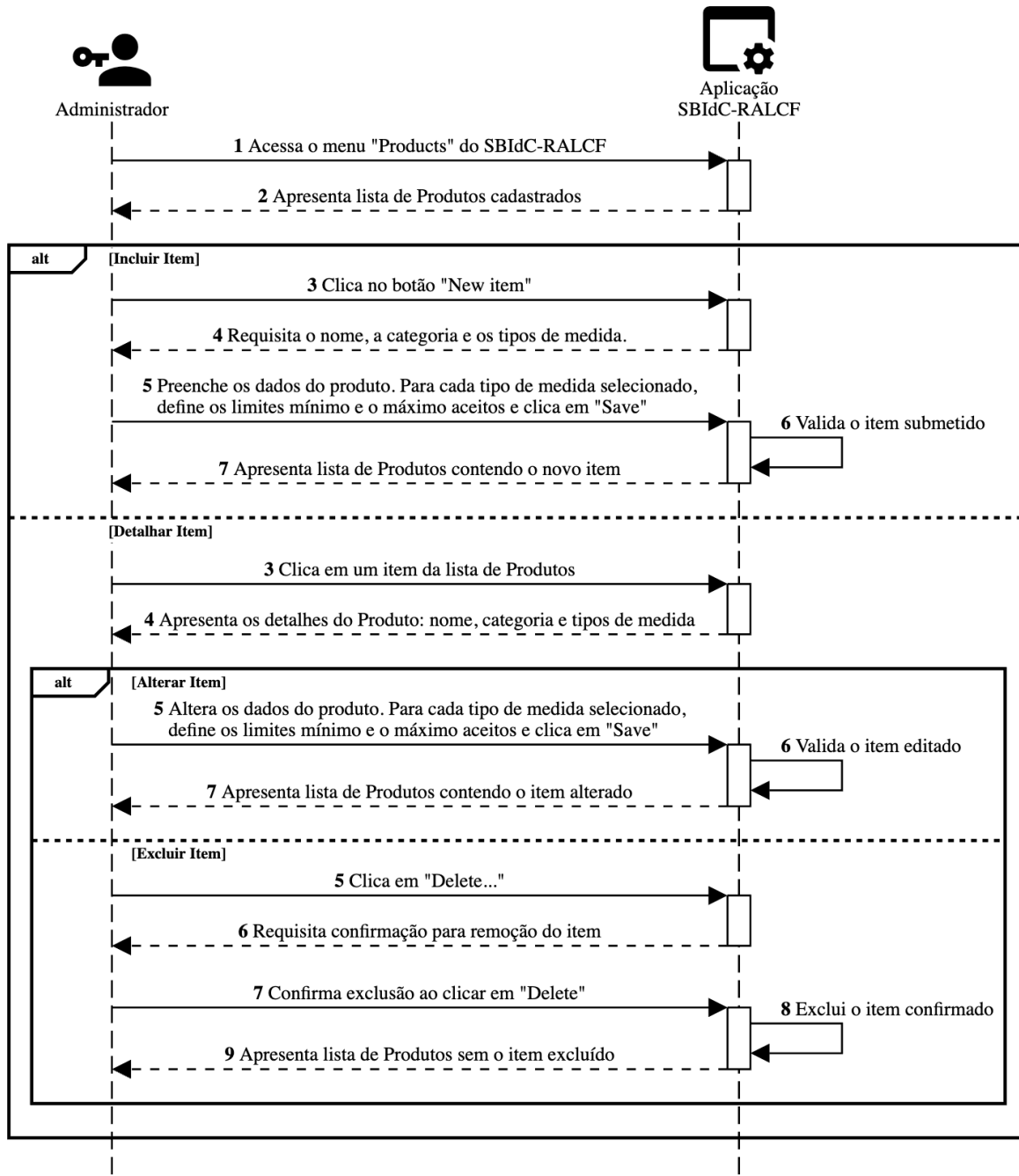
Figura 42 – Diagrama de sequência do CRUD de Dispositivos



Fonte: Elaborado pelo autor

## A.4 CRUD de Produtos

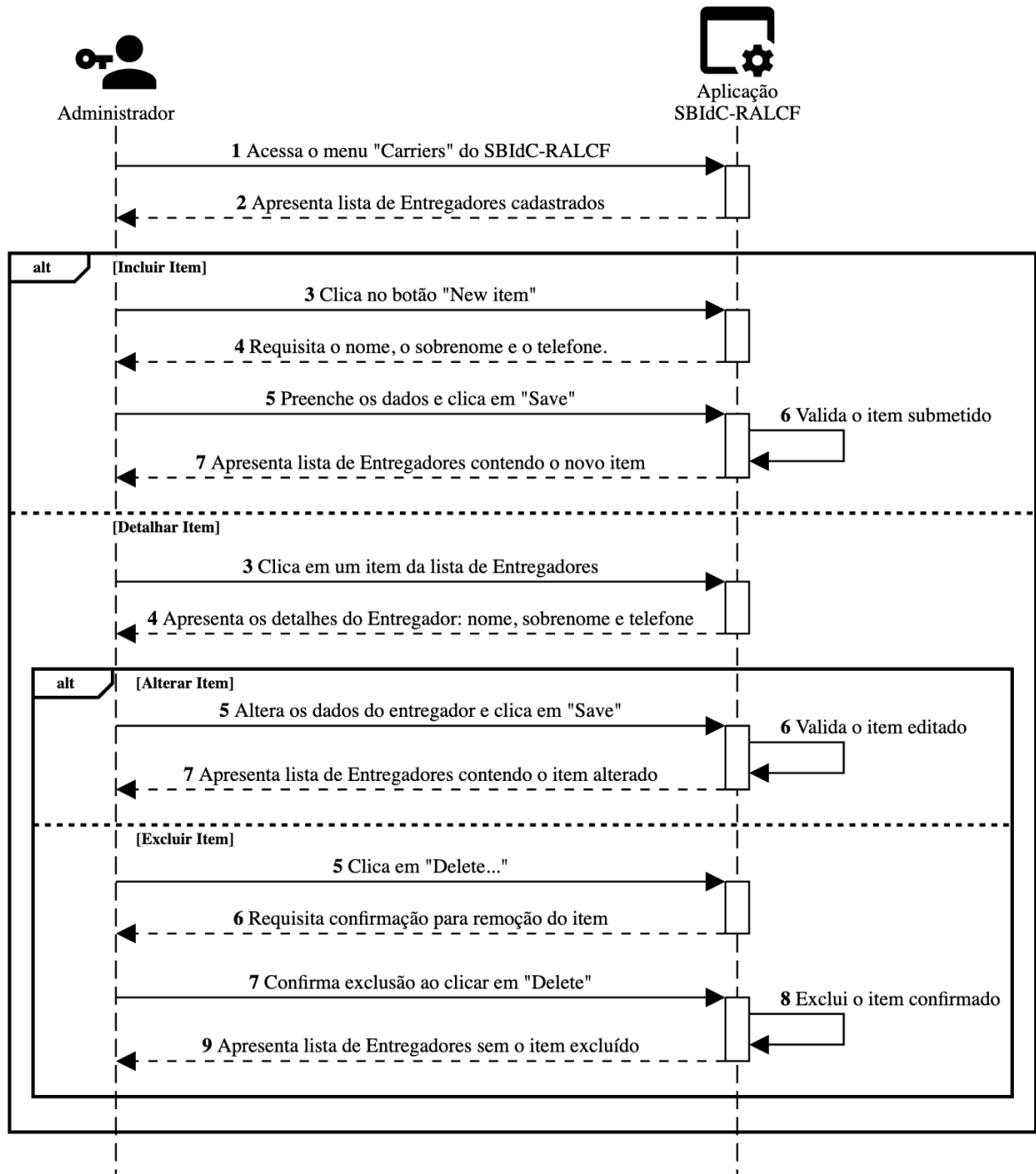
Figura 43 – Diagrama de sequência do CRUD de Produtos



Fonte: Elaborado pelo autor

## A.5 CRUD de Entregadores

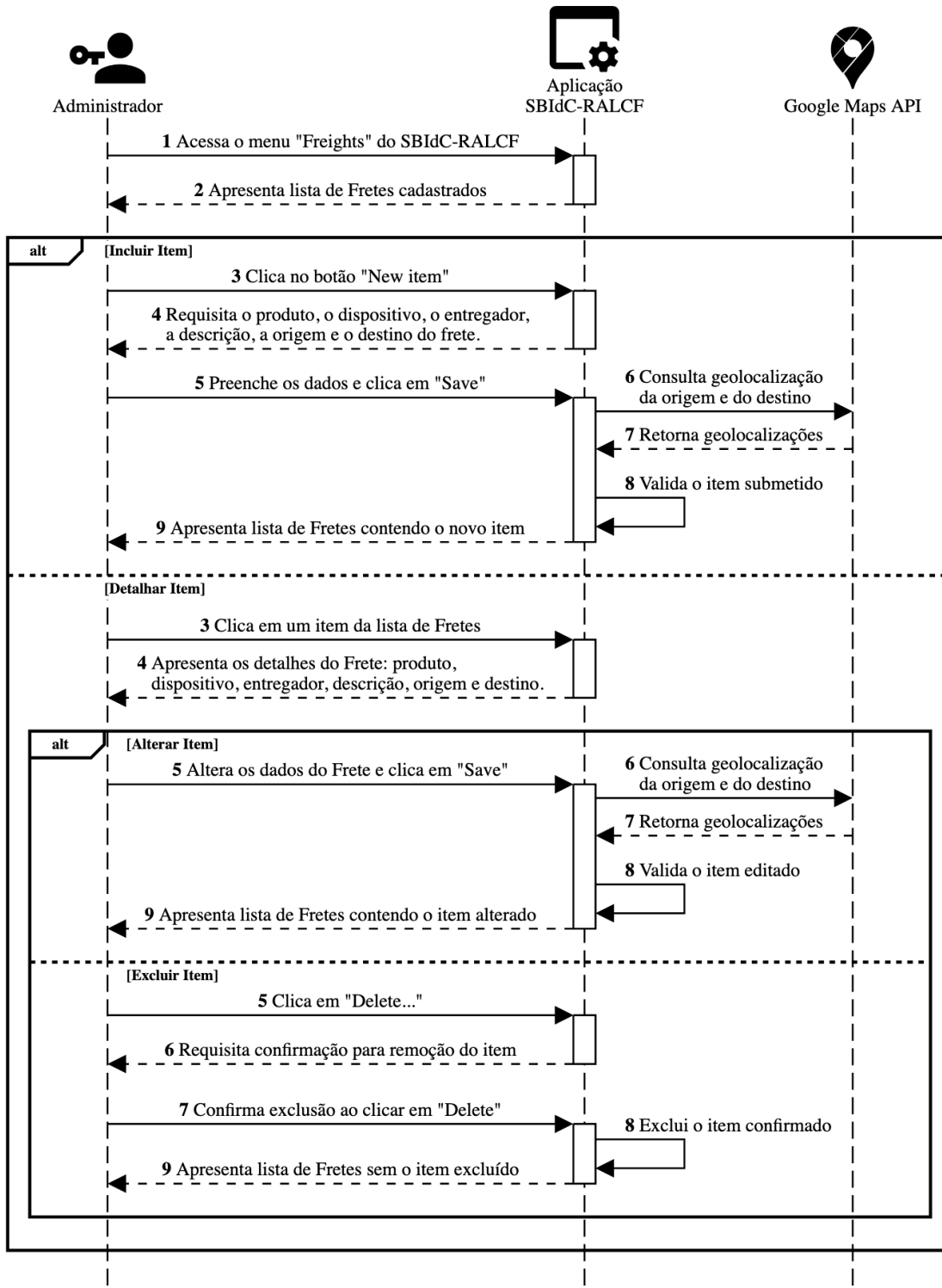
Figura 44 – Diagrama de sequência do CRUD de Entregadores



Fonte: Elaborado pelo autor

## A.6 CRUD de Fretes

Figura 45 – Diagrama de sequência do CRUD de Fretes



Fonte: Elaborado pelo autor

---

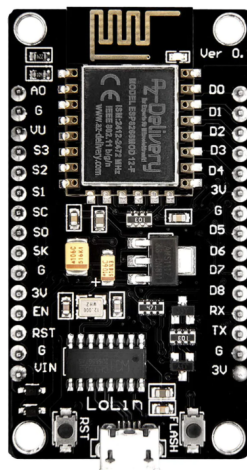
# APÊNDICE B

## NodeMCU V3

---

A placa de desenvolvimento AzDelivery<sup>1</sup> NodeMCU V3, ilustrada na Figura 46, é equipada com o microcontrolador ESP8266 (ESP-12F) com conectividade Wi-Fi integrada e possui 30 pinos distribuídos em suas laterais, com funções de alimentação, entradas e saídas digitais, entrada analógica, dentre outras.

Figura 46 – Placa AzDelivery NodeMCU V3



Fonte: Elaborado pelo autor

A seguir está uma descrição e funções dos principais pinos da placa NodeMCU V3 utilizados no SBIdC-RALCF.

### □ Alimentação

---

<sup>1</sup> <<https://www.az-delivery.com>>

- **VIN**: Entrada de alimentação. Aceita entre 4,5V e 10V via fonte externa ou bateria.
- **3V**: Saída de 3,3V regulada pela placa. Pode ser usada para alimentar sensores ou outros componentes.
- **GND**: Terra. Referência para o circuito.

#### □ Comunicação

- **TX (GPIO1)**: Transmissão de dados pela Transmissor/Receptor Universal Síncrono e Assíncrono, do inglês *Universal Synchronous Asynchronous Receiver Transmitter* (UART). Usado para comunicação serial com outros dispositivos (e.g. outro microcontrolador).
- **RX (GPIO3)**: Recepção de dados pela UART. Usado para comunicação serial.

#### □ GPIO: A placa tem vários GPIOs, numerados de GPIO00 a GPIO16. Eles podem ser usados como entradas ou saídas digitais e configurados para diversas funções.

- **GPIO00 a GPIO15**: Entradas/Saídas Digitais. Alguns pinos suportam Modulação por largura de pulso, do inglês *Pulse-Width Modulation* (PWM), *Inter-Integrated Circuit* (I2C), *Serial Peripheral Interface* (SPI) e funções específicas.
- **GPIO16**: Também usado para funções especiais como *deep sleep*.

#### □ Conversor analógico-digital, em inglês *Analog-to-digital converter* (ADC)

- **A0**: Entrada analógica. Usado para ler sensores analógicos com um intervalo de tensão entre 0V e 3,3V

#### □ Comunicação Serial

- **I2C**: Comunicação com dispositivos I2C como sensores e displays.
  - \* **SCL**: GPIO5 (D1)
  - \* **SDA**: GPIO4 (D2)
- **SPI**: Comunicação com dispositivos SPI como memórias Flash e sensores.
  - \* **MOSI**: GPIO13 (D7)
  - \* **MISO**: GPIO12 (D6).
  - \* **SCK**: GPIO14 (D5).
  - \* **CS**: GPIO15 (D8).
- **PWM**: A maioria dos GPIOs permite a geração de sinais PWM para controle de motores, LEDs e outros dispositivos.

#### □ Controle

- **EN**: Habilita o funcionamento da placa. Em nível baixo, desativa o ESP8266.
- **RST**: Reset da placa. Pode ser ligado a um botão externo para reiniciar o sistema.

Este Apêndice apresentou um resumo da placa NodeMCU V3 para o seu uso no SBIdC-RALCF. Recomenda-se a consulta da ficha de dados (ou *datasheet*) do fabricante para detalhes adicionais.



---

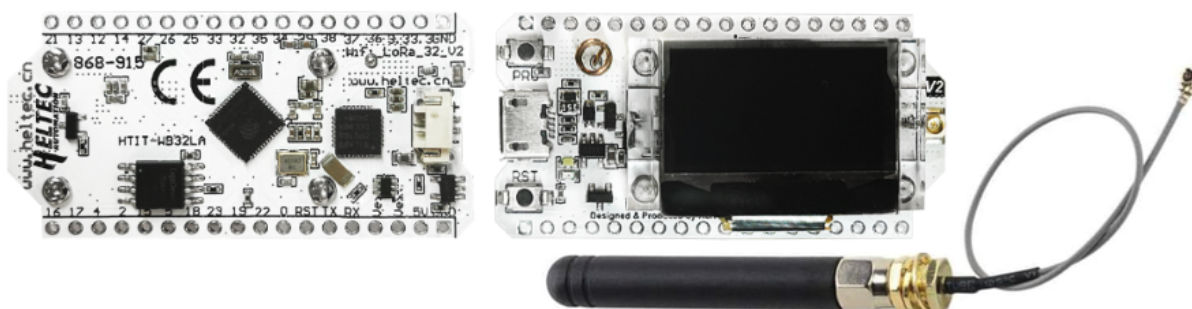
# APÊNDICE C

## WiFi LoRa 32 V2

---

WiFi LoRa 32 V2 é uma placa de desenvolvimento IdC projetada e produzida pela Heltec Automation(TM)<sup>1</sup>. É baseada no microcontrolador ESP32 e integrada com um módulo LoRa SX127x, suportando Wi-Fi, BLE, LoRa e visor diodo orgânico emissor de luz, do inglês *organic light-emitting diode* (OLED) de 0,96 polegada. Pode ser alimentada via USB, bateria Polímero de Lítio, do inglês *Lithium Polymer* (LiPo) com conector *Japan Solderless Terminal* (JST) ou pino 5V. O dispositivo é ilustrado na Figura 47 justamente com uma antena compatível com conector *SubMiniature version A* (SMA).

Figura 47 – Placa Heltec WiFi LoRa 32 V2 com antena compatível



Fonte: Adaptado de <<https://heltec.org>>

A seguir está uma descrição e funções dos principais pinos da placa WiFi LoRa 32 V2 utilizados no SBIdC-RALCF.

### □ Alimentação

---

<sup>1</sup> <<https://heltec.org>>

- **5V**: Entrada de alimentação. Pode ser usada para alimentar a placa via uma fonte externa.
  - **3V**: Fornece uma tensão regulada de 3,3V para a alimentação de periféricos externos.
  - **GND**: Terra. Referência elétrica para o circuito.
- Comunicação: A placa suporta várias interfaces de comunicação, como UART, SPI, I2C, e LoRa
- **UART**
    - \* **TX (GPIO1)**: Transmissão serial.
    - \* **RX (GPIO3)**: Recepção serial.
  - **I2C**
    - \* **SCL (GPIO22)**: Relógio para dispositivos I2C.
    - \* **SDA (GPIO21)**: Dados para dispositivos I2C.
  - **SPI**
    - \* **SCK (GPIO14)**: Clock do barramento SPI.
    - \* **MISO (GPIO12)**: Entrada de dados do SPI.
    - \* **MOSI (GPIO13)**: Saída de dados do SPI.
    - \* **CS (GPIO5)**: Seleção de *chip*.
  - **LoRa (SX1276/78)**: Controlado pelos GPIOs 16 e 17, além dos pinos SPI (MOSI, MISO, SCK e CS).
- GPIO: Os GPIOs são configuráveis como entradas ou saídas digitais, com suporte para PWM, I2C, SPI, UART, entre outros.
- **GPIO0 (IO0)**: *Boot* ou entrada/saída digital.
  - **GPIO2 (IO2)**: Entrada/saída digital.
  - **GPIO4 (IO4)**: Entrada/saída digital, PWM.
  - **GPIO5 (IO5)**: SPI CS.
  - **GPIO12 (IO12)**: SPI MISO.
  - **GPIO13 (IO13)**: SPI MOSI.
  - **GPIO14 (IO14)**: SPI SCK.
  - **GPIO15 (IO15)**: Entrada/saída digital.
  - **GPIO16 (IO16)**: Controle do SX1276.
  - **GPIO17 (IO17)**: Controle do SX1276.

□ ADC

- **ADC0**: Entrada analógica para leitura de tensões de 0V a 3,3V.

□ Controle

- **EN**: Ativa/desativa o regulador de tensão da placa. Em nível baixo, desativa a placa.
- **RST**: Reset da placa. Pode ser ligado a um botão externo para reiniciar o sistema.

Este Apêndice apresentou um resumo da placa WiFi LoRa 32 V2 para o seu uso no SBIdC-RALCF. Recomenda-se a consulta da ficha de dados (ou *datasheet*) do fabricante para detalhes adicionais.



---

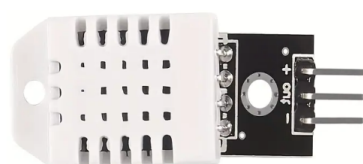
## APÊNDICE D

### Módulo DHT22

---

O módulo DHT22 contém um sensor de temperatura e umidade digital DHT22 que fornece medições de alta precisão. Ele é utilizado em projetos de automação e monitoramento ambiental devido ao seu baixo custo e sua facilidade de uso, pois elimina a necessidade de adicionar um resistor de *pull-up* (de 4,7k a 10k ohms) entre os pinos de dados e alimentação para garantir a estabilidade no sinal externamente.

Figura 48 – Módulo DHT22



Fonte: Elaborado pelo autor

A seguir está uma descrição e funções dos principais pinos do módulo DHT22 ilustrados na Figura 48:

1. **VCC**: Alimentação do sensor. Requer uma alimentação entre 3,3V e 5,5V e é tipicamente conectado à fonte de alimentação do microcontrolador ou circuito.
2. **DATA**: Comunicação e transferência de dados entre o DHT22 e o microcontrolador. Esse pino é conectado ao pino de entrada digital do microcontrolador. O módulo usa um protocolo proprietário de um fio, onde o sensor envia um pulso de dados para o microcontrolador que contém as informações de temperatura e umidade.
3. **GND**: Terra. Conexão de referência para o circuito. Conectado ao terra do microcontrolador ou da fonte de alimentação.

Este Apêndice apresentou um resumo do módulo DHT22 para o seu uso no SBIdC-RALCF. Recomenda-se a consulta da ficha de dados (ou *datasheet*) do fabricante para detalhes adicionais.

---

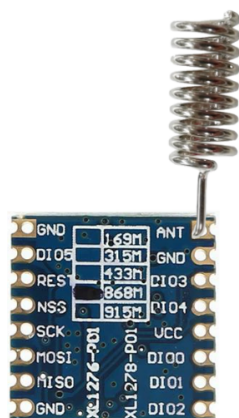
# APÊNDICE E

## Módulo LoRa SX1276

---

O módulo SX1276 é um transceptor de rádio de longo alcance que opera nas faixas de frequência entre 137 MHz e 1020 MHz. Ele é comumente usado em comunicações LoRa e sistemas de rádio de baixa potência.

Figura 49 – Módulo SX1276 com antena soldada



Fonte: Elaborado pelo autor

A seguir está uma descrição e funções dos principais pinos ilustrados na Figura 49:

1. **GND**: Terra. Conexão de referência para o circuito.
2. **VCC**: Alimentação. Esse pino fornece a energia ao chip e geralmente funciona em uma faixa de 1,8V a 37V, dependendo da configuração.

3. **MISO**: Master In Slave Out. Esse é o pino de saída de dados do SX1276 para o microcontrolador ou microprocessador mestre. Ele fornece dados ao dispositivo mestre.
4. **MOSI**: Master Out Slave In. É o pino de entrada de dados para o SX1276. O microcontrolador ou microprocessador mestre envia os dados através deste pino.
5. **SCK**: Serial Clock. Esse é o pino de clock do barramento SPI. O microcontrolador mestre gera o sinal de clock que sincroniza a comunicação entre o SX1276 e o dispositivo mestre.
6. **NSS**: Chip Select (pino NSS). Esse é o pino de seleção de chip, que permite ao mestre selecionar o SX1276 para comunicação via SPI. Quando em nível baixo, o SX1276 é habilitado para a transmissão de dados.
7. **DIO0**: Digital I/O 0. Esse pino geralmente é usado para sinalizar um evento de recebimento ou transmissão, como uma interrupção no microcontrolador. Ele pode ser configurado para diferentes funções, dependendo do modo de operação.
8. **DIO1**: Digital I/O 1. Também é configurável e usado para diferentes sinais de interrupção ou eventos específicos de status.
9. **DIO2**: Digital I/O 2. Outro pino de I/O digital configurável para funções específicas, como sinalização de detecção de portadora (RSSI).
10. **DIO3**: Digital I/O 3. Pode ser configurado para sinalizar eventos, como fim de transmissão ou recepção.
11. **DIO4**: Digital I/O 4. Esse pino tem uso semelhante aos anteriores, sendo configurável.
12. **DIO5**: Digital I/O 5. Último pino de I/O configurável.
13. **ANT**: Conector de Antena. Esse pino é conectado à antena externa para transmissão e recepção de sinais.
14. **RESET**: Pino de Reset. Esse pino é usado para resetar o chip, reiniciando suas configurações e colocando-o no estado inicial.

Este Apêndice apresentou um resumo do módulo SX1276 para o seu uso no SBIdC-RALCF. Recomenda-se a consulta da ficha de dados (ou *datasheet*) do fabricante para detalhes adicionais.

---

# APÊNDICE F

## Arduino IDE

---

As seções deste Apêndice apresentam as configurações que devem ser aplicadas no Arduino IDE<sup>1</sup> para suportar as placas NodeMCU V3 e WiFi LoRa 32 V2, as bibliotecas que devem ser instaladas para utilizar os módulos DHT22 e SX1276, e a parametrização da região para a biblioteca LMIC. A instalação do Arduino IDE não está no escopo deste Apêndice.

### F.1 Habilitar NodeMCU V3

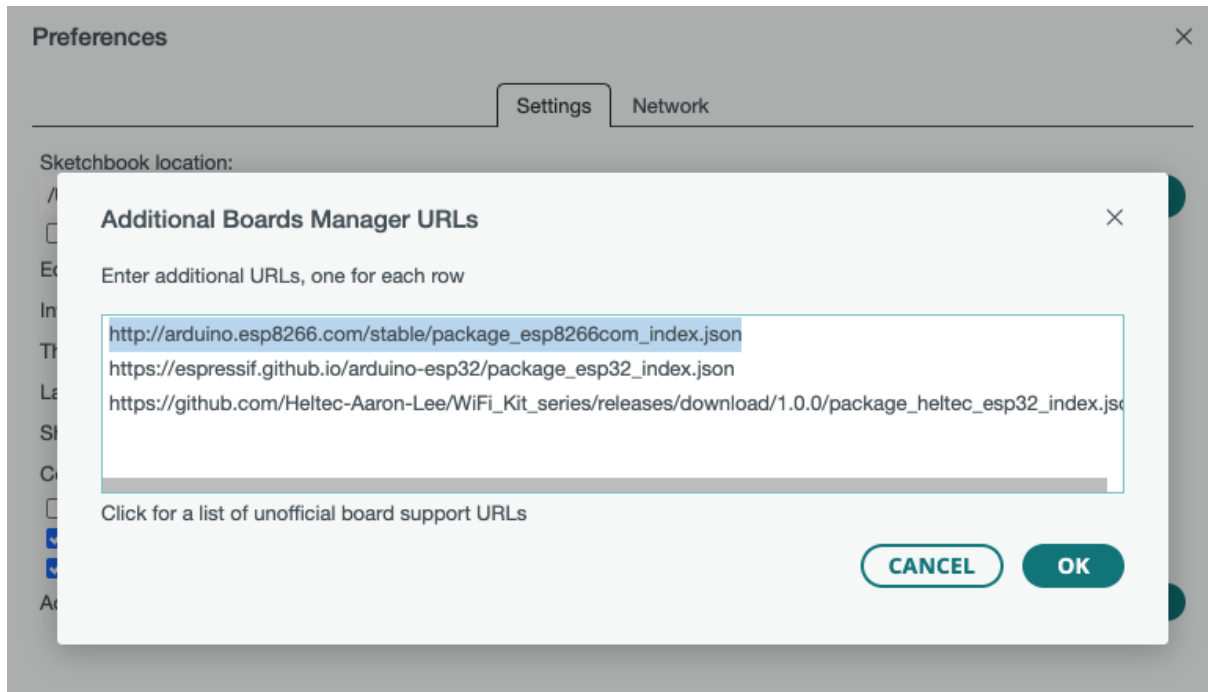
Para que a NodeMCU V3 seja suportada no Arduino IDE é necessário que a URL das placas ESP8266 seja incluída nas configurações adicionais da ferramenta de desenvolvimento. Os passos a seguir são suficientes para que esta configuração seja realizada.

1. A configuração do gerenciador da placa é acessível pelo item de menu “Settings” do Arduino IDE.
2. A URL <[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)> deve ser adicionada no campo “Additional Boards Manager URLs”, como ilustrado na Figura 50, e o botão “OK” clicado para salvar a alteração.
3. Neste momento, após conectar a NodeMCU V3 em uma porta aceita pelo Arduino IDE e selecionar “NodeMCU 1.0 (ESP-12E)” nas opções de placa disponíveis, como ilustrado na Figura 51, é suficiente para embarcar o *firmware* do SBIdC-RALCF na placa.

---

<sup>1</sup> <<https://www.arduino.cc/en/software>>

Figura 50 – Configuração das URLs das placas ESP8266 e ESP32 no Arduino IDE



Fonte: Captura de tela do Arduino IDE

Figura 51 – NodeMCU 1.0 (ESP-12E) listada no Arduino IDE



Fonte: Captura de tela do Arduino IDE

## F.2 Habilitar WiFi LoRa 32 V2

Para que a WiFi LoRa 32 V2 seja suportada no Arduino IDE é necessário que a URL das placas ESP32 seja incluída nas configurações adicionais da ferramenta de desenvolvimento. Os passos a seguir são suficientes para que esta configuração seja realizada.

1. A configuração do gerenciador da placa é acessível pelo item de menu “Settings” do Arduino IDE.
2. A URL `<https://espressif.github.io/arduino-esp32/package_esp32_index.json>` deve ser adicionada no campo “Additional Boards Manager URLs”, como ilustrado na Figura 50, e o botão “OK” clicado para salvar a alteração.
3. Neste momento, após conectar a WiFi LoRa 32 V2 em uma porta aceita pelo Arduino IDE e selecionar “WiFi LoRa 32(V2)” nas opções de placa disponíveis, como ilustrado na Figura 52, é suficiente para embarcar o *firmware* do SBIdC-RALCF na placa.

Figura 52 – WiFi LoRa 32(V2) listada no Arduino IDE



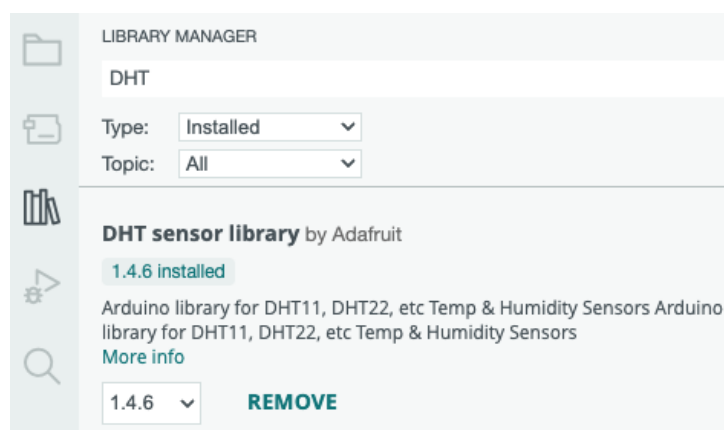
Fonte: Captura de tela do Arduino IDE

## F.3 Instalar biblioteca DHT

Para que os dispositivos sejam capazes de ler as aferições dos sensores do módulo DHT22, a biblioteca DHT da Adafruit<sup>2</sup> deve ser instalada no Arduino IDE. Os passos a seguir são suficientes para que esta configuração seja realizada.

1. A gestão de bibliotecas do Arduino IDE é acessível através do caminho de menu “Tools” e “Manage Libraries”.
2. Ao pesquisar por “DHT”, é possível instalar a biblioteca “DHT sensor library” da Adafruit, conforme indicado na Figura 53.

Figura 53 – Biblioteca DHT da Adafruit instalada no Arduino IDE



Fonte: Captura de tela do Arduino IDE

3. A partir disso, a biblioteca está disponível para ser importada no código-fonte.

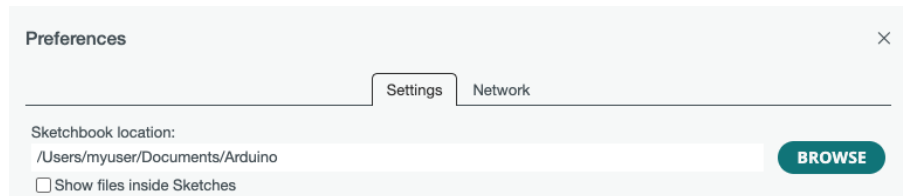
## F.4 Instalar biblioteca LMIC

Para que os dispositivos sejam capazes de enviar e receber mensagens LoRa pelos transceptores SX1276, a biblioteca LMIC<sup>3</sup> deve ser instalada no Arduino IDE. Os passos a seguir são suficientes para que esta configuração seja realizada.

<sup>2</sup> <<https://github.com/adafruit/DHT-sensor-library>>

<sup>3</sup> <<https://github.com/mcci-catena/arduino-lmic>>

Figura 55 – Caminho que contém os arquivos do Arduino IDE



Fonte: Captura de tela do Arduino IDE

1. A gestão de bibliotecas do Arduino IDE é acessível através do caminho de menu “Tools” e “Manage Libraries”.
2. Ao pesquisar por “LMIC”, é possível instalar a biblioteca “MCCI LoRaWAN LMIC Library”, conforme indicado na Figura 54.

Figura 54 – Biblioteca “MCCI LoRaWAN LMIC Library” instalada no Arduino IDE



Fonte: Captura de tela do Arduino IDE

3. A partir disso, a biblioteca está disponível para ser importada no código-fonte.

## F.5 Configuração dos parâmetros da LMIC

Uma vez que a biblioteca LMIC está instalada conforme o Apêndice F.4, os parâmetros regionais necessários para a execução do SBIDC-RALCF podem ser configurados. Os passos a seguir descrevem como a região Europeia pode ser definida para um projeto que utiliza a biblioteca LMIC.

1. O diretório base que contém os arquivos do Arduino é acessível pelo item de menu “Settings” do IDE.
2. O diretório base é definido no campo “Sketchbook location”, conforme ilustrado na Figura 55.
3. Dentro do diretório base, o caminho `libraries/MCCI_LoRaWAN_LMIC_library/project_config` contém o arquivo com as configurações do LMIC para o projeto, chamado `lmic_project_config.h`.

4. Todas as regiões suportadas pelo LMIC são comentadas no arquivo de configuração. No caso do SBIdC-RALCF, a linha `#define CFG_eu868 1` foi adicionada pois a região dos servidores de rede é a Europa (largura de banda entre 863 MHz e 870 MHz).

Uma vez configurada a região, o código-fonte Arduino do SBIdC-RALCF define as constantes `APPEUI`, `DEVEUI` e `APPKEY`. Elas devem ser preenchidas respectivamente com os valores das chaves `AppEUI`, `DevEUI`, e `AppKey`, provenientes dos servidores de rede no processo registro do dispositivo.

Os valores da TTN devem ser definidos para ambas as placas dentro do bloco iniciado por `#ifdef TTN_NETWORK`, após o registro do dispositivo descrito no Apêndice H.2. Por outro lado, o bloco `#ifdef KPN_NETWORK` controla as constantes para os valores provindos da KPN Things, detalhados no Apêndice G.1.

A LMIC espera os valores das constantes em formatos específicos. No caso de `APPEUI` e `DEVEUI`, o formato esperado é *little-endian*, com o byte menos significativo primeiro. Já a constante `APPKEY` deve ser definida com o formato *big endian*.



---

## APÊNDICE G

# Configuração do Servidor de Rede KPN Things

---

As seções deste apêndice detalham os passos necessários para se utilizar a rede KPN como Servidor de Rede LoRaWAN no SBIdC-RALCF: o registro de dispositivos, a configuração de um *broker* MQTT como destino e, por fim, a vinculação destes dispositivos a este destino para o encaminhamento das mensagens recebidas.

A configuração necessária em relação ao Servidor de Rede KPN Things é possível após a criação de um usuário no portal <<https://portal.kpnthings.com>>. O aceite dos termos e condições e a verificação do e-mail utilizado no cadastro são suficientes para a definição da senha de acesso. Um projeto no plano gratuito da KPN (*Freemium*) é automaticamente criado e os seus detalhes, como o seu nome, podem ser alterados através do menu “Account & Projects”, “Project Details”, e “Edit”.

### G.1 Registro de dispositivo

As etapas a seguir descrevem como um dispositivo próprio pode ser registrado na KPN Things. O pacote gratuito da KPN suporta até três dispositivos da categoria *Bring Your Own Device* (BYOD).

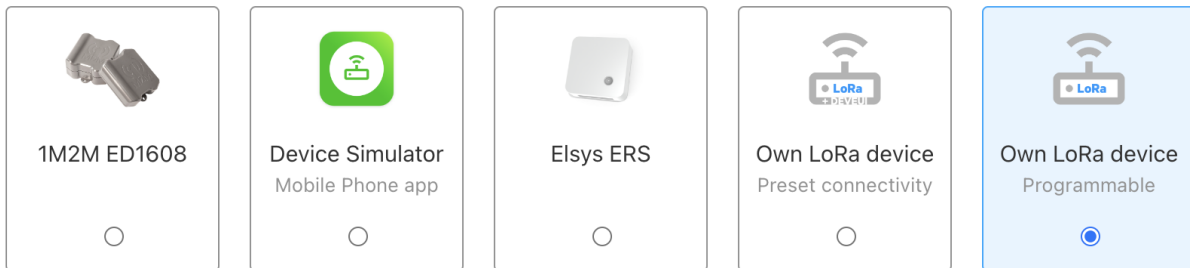
1. Após entrar o portal KPN Things, é necessário acessar o menu “The Things Manager”, item “Devices”, e depois apertar o botão “Add new device” para adicionar um novo dispositivo.
2. Uma série de opções de dispositivos é exibida, como ilustrado na Figura 56, dentre elas o dispositivo próprio programável, *Own LoRa device - Programmable* em inglês.

Após selecioná-la, é necessário definir um nome e apertar o botão “Add device” para adicionar um dispositivo.

Figura 56 – Adicionar dispositivo no portal KPN Things

### 1 of 2. Add your Device to SBIdC-RALCF

Which Device type do you want to provision?



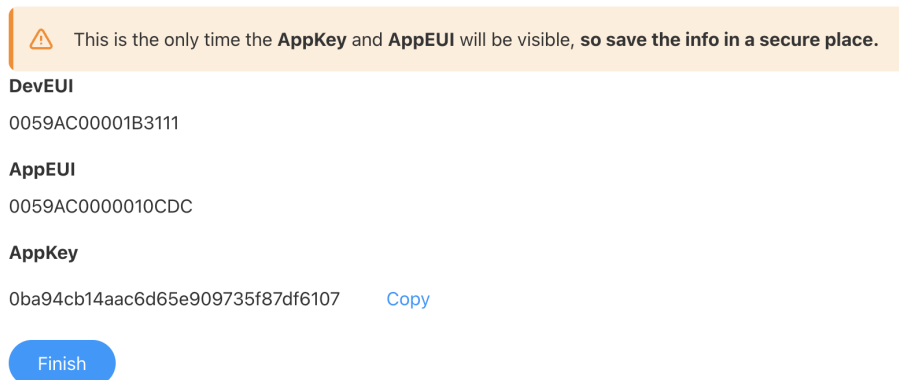
Fonte: Captura de tela do portal KPN Things

- Os detalhes para a conexão com a rede LoRa (*DevEUI*, *AppEUI* e *AppKey*) são gerados, como o exemplo apresentado na Figura 57. Estes dados são utilizados no código-fonte Arduino para o processo de solicitação de adesão à rede KPN via OTAA.

Figura 57 – *DevEUI*, *AppEUI* e *AppKey* gerados pela KPN Things para o dispositivo

### 2 of 2. Add Network Info

Network details for the LoRa network are generated for you.



Fonte: Captura de tela do portal KPN Things

Após seguir estas etapas e utilizar os parâmetros gerados no código-fonte embarcado no dispositivo, ele está apto a enviar solicitações de adesão à rede LoRaWAN KPN, além de enviar e receber mensagens para os gateways dela. A lista de dispositivos disponíveis é acessível através do menu “The Things Manager”, item “Devices”.

## G.2 Configuração do *broker* MQTT

O provedor KPN Things não oferece um broker MQTT, mas provê a opção de encaminhamento de mensagens para um *broker* MQTT externo à sua infraestrutura. Os seguintes passos descritos são suficientes para se configurar um *broker* como destino, sendo que o SBIdC-RALCF utiliza a opção pública gratuita da EMQ (EMQ Technologies Inc, 2024).

1. Uma lista de tipos de destino é exibida ao acessar o menu “The Things Manager”, item “Destinations”, e apertar o botão “Add new destination”.
2. Aperte o botão “Choose MQTT” para adicionar um *broker* MQTT como destino.
3. O *broker* é criado como um novo destino após se definir um nome, os detalhes de conexão MQTT, e o formato do nome do tópico, como ilustrado na Figura 58, e apertar o botão “Add destination”.

Figura 58 – Especificação do *broker* MQTT como destino para as mensagens dos dispositivos KPN

### Connection details

Enter the details of your MQTT broker. Would you like KPN to create an MQTT broker for you? Contact [iot@kpn.com](mailto:iot@kpn.com).

#### Host

#### Port

#### Username

#### Password

#### Topic

Enter the target location for the messages. You can use several {variables} that will be replaced dynamically when sending the message.

Fonte: Captura de tela do portal KPN Things

## G.3 Fluxo de mensagens do *broker* MQTT

A KPN possui uma funcionalidade, chamada Fluxo (*Flow*), que possibilita encaminhar as mensagens recebidas dos dispositivos para o destino configurado. Os passos descritos a seguir descrevem como esta configuração deve ser realizada.

1. Um fluxo pode ser criado ao acessar o menu “The Things Manager”, item “Flows”, e apertar o botão “Add new flow”.
2. Após definir um nome para o fluxo, deve-se vincular o(s) dispositivo(s) e o(s) destino(s) previamente cadastrados.
3. Adicionalmente, é possível configurar codificador e decodificador para as mensagens recebidas. O SBIdC-RALCF utiliza “Raw LoRa payload” para ambos, o que significa que as mensagens são encaminhadas em formato bruto.

---

# APÊNDICE H

## Configuração do Servidor de Rede TTN

---

A configuração necessária em relação ao Servidor de Rede TTN é possível após a criação de um usuário no portal <<https://id.thethingsnetwork.org>>. O aceite dos termos e condições e a verificação do e-mail utilizado no cadastro são suficientes para a definição da senha de acesso.

As seções deste apêndice detalham os passos necessários para se utilizar a rede TTN como Servidor de Rede LoRaWAN no SBIdC-RALCF: a criação de uma aplicação na região adequada e o registro de dispositivos. A configuração do aplicativo “TTN Mapper” também é descrita. Adicionalmente, The Things Industries (2024c) detalha como a chave da aplicação para acesso ao *broker* MQTT pode ser configurada.

### H.1 Criação de aplicação

Uma aplicação no TTN é criada dentro de um *cluster* de uma região. A região selecionada deve estar de acordo com a frequência dos dispositivos que se deseja registrar. As seguintes etapas são suficientes para se configurar uma aplicação:

1. Após acessar o portal <<https://console.cloud.thethings.network>>, é necessário escolher a região desejada: “Europe 1”, “North America 1” ou “Australia 1”.
2. O formulário de inclusão de uma aplicação é acessível através do item de menu “Applications” e posterior clique no botão “Add new application”.

3. Após a definição do identificador da aplicação e de seus campos opcionais (nome e descrição), ao clicar em “Create application” uma nova aplicação é criada.

Figura 59 – Formulário para inclusão de um dispositivo no portal TTN

## Register end device

Does your end device have a LoRaWAN® Device Identification QR Code? Scan it to speed up onboarding.

[Device registration help](#)

### End device type

#### Input method

- Select the end device in the LoRaWAN Device Repository
- Enter end device specifics manually


#### Frequency plan \*

Europe 863-870 MHz (SF9 for RX2 - recommended) | 

#### LoRaWAN version \*

LoRaWAN Specification 1.0.3 | 

#### Regional Parameters version \*

RP001 Regional Parameters 1.0.3 revision A | 

[Show advanced activation, LoRaWAN class and cluster settings](#)

### Provisioning information

#### JoinEUI \*

To continue, please enter the JoinEUI of the end device so we can determine onboarding options

Fonte: Captura de tela do portal TTN

## H.2 Registro de dispositivo

O SBIdC-RALCF utiliza dispositivos de uso geral como nós finais LoRaWAN. As etapas a seguir descrevem como tal dispositivo pode ser registrado no TTN.

1. Um dispositivo é registrado dentro de uma aplicação, que é listada através do item de menu “Applications”. Após clicar em uma aplicação, as suas possíveis configurações são exibidas.
2. Dentro de uma aplicação, o formulário de registro de um dispositivo é acessível pelo item de menu “End devices” e posterior clique no botão “Register end device”.

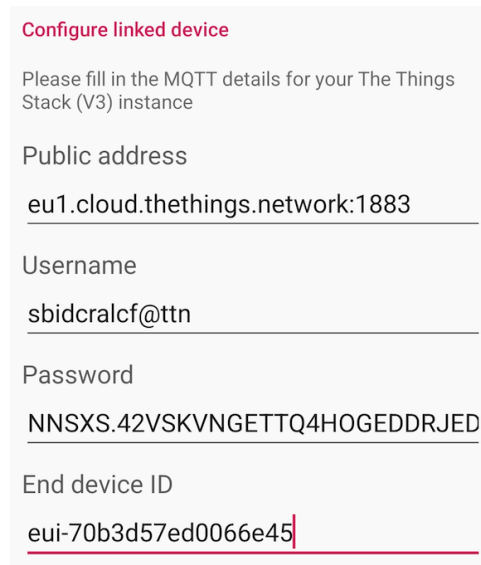
3. Há a opção de registro através de código QR. No entanto, no caso do dispositivo de uso geral do SBIdC-RALCF, utiliza-se a entrada manual (*Enter end device specifics manually*), escolhe-se a frequência, a versão LoRaWAN e preenche-se o parâmetro *JoinEUI* com zeros, conforme ilustrado na Figura 59.
4. Em sequência, define-se o nome do dispositivo, gera-se o par de chaves *DevEUI* e *AppEUI*, utilizadas para adesão à rede. Ao clicar-se no botão “Register end device”, o dispositivo é registrado.
5. Os detalhes do dispositivo registrados são exibidos. Além das chaves previamente geradas, a chave *AppKey* também pode ser copiada.

Após se seguir estas etapas e utilizar os parâmetros gerados (*DevEUI*, *AppEUI* e *AppKey*) no *firmware* no dispositivo, ele está apto a enviar solicitações de adesão OTAA à rede TTN, além de enviar e receber mensagens para os gateways dela.

## H.3 Configuração do TTN Mapper

A TTN não suporta a geolocalização por multilateração por padrão como a KPN Things. No entanto, o aplicativo “TTN Mapper” instalado em um *smartphone* é uma alternativa para calcular a distância entre o dispositivo LoRaWAN e o gateway receptor. Os passos a seguir são suficientes para configurar o “TTN Mapper” em um celular para se conectar com o *broker* MQTT da TTN e, utilizando o GPS deste dispositivo, registrar a geolocalização do protótipo no momento que o gateway recebeu a mensagem.

1. O aplicativo “TTN Mapper” deve ser instalado em um celular Android ou iPhone.
2. Deve ser concedida a permissão de utilização do recurso de GPS.
3. Ao abrir o aplicativo e clicar no item “Link a new device” e selecionar a opção “The Things Stack (V3)”, as configurações do *broker* MQTT e identificador do dispositivo são solicitadas, conforme ilustrado na Figura 60. Nota-se aqui que “TTN Mapper” também suporta o servidor de rede ChirpStack.

Figura 60 – Configuração do *broker* MQTT no “TTN Mapper”

**Configure linked device**

Please fill in the MQTT details for your The Things Stack (V3) instance

Public address  
eu1.cloud.thethings.network:1883

Username  
sbidcralcf@ttn

Password  
NNSXS.42VSKVNGETTQ4HOGEDDRJED

End device ID  
eui-70b3d57ed0066e45

Fonte: Captura de tela do aplicativo “TTN Mapper”

4. As configurações provenientes do *broker* disponibilizado pela TTN (The Things Industries, 2024c) devem ser inseridas no aplicativo e o botão “Link device” clicado.

Após seguir os passos supracitados, o “TTN Mapper” está apto a registrar a geolocalização do dispositivo. A seguir, são descritos os passos para iniciar este processo.

1. Ao clicar no botão (ou ícone) “Play” no início do aplicativo, o “TTN Mapper” torna-se um assinante dos tópicos do *broker* MQTT da TTN.
2. Quando o servidor de rede publica uma mensagem enviada por um nó final no *broker*, o “TTN Mapper” recebe esta mensagem de *uplink* e registra a geolocalização do dispositivo celular. Considerando que o celular e o nó final LoRaWAN utilizados estejam próximos, há uma correspondência da geolocalização para este nó.
3. O “TTN Mapper” encerra a assinatura ao se clicar no botão (ou ícone) “Pause”.
4. Por meio do item “Recorded sessions” do aplicativo é possível acessar todos os metadados registrados para uma sessão. Um arquivo CSV pode ser gerado com estas informações.

---

# APÊNDICE I

## Execução da aplicação

---

As seções deste apêndice apresentam os pré-requisitos para que a aplicação do SBIdC-RALCF seja executada localmente, além da execução do teste de carga sobre a mesma. Para executar a aplicação localmente, os softwares e as variáveis de ambiente para as integrações com os servicores de rede e Google Maps API devem ser configurados.

### I.1 Pré-requisitos de software

Dois softwares são pré-requisitos mínimos e devem ser instalados para executar localmente a aplicação do SBIdC-RALCF:

1. Java Development Kit (JDK) 21<sup>1</sup>
2. Docker<sup>2</sup>

Opcionalmente, o Git SCM<sup>3</sup> também pode ser instalado para a obtenção e o versionamento do código-fonte<sup>4</sup> do projeto.

### I.2 Variáveis de ambiente

As seguintes variáveis de ambiente devem ser definidas antes da execução da aplicação do SBIdC-RALCF:

---

<sup>1</sup> <<https://www.oracle.com/br/java/technologies/downloads/#java21>>

<sup>2</sup> <<https://docs.docker.com/engine/install/>>

<sup>3</sup> <<https://git-scm.com/>>

<sup>4</sup> <<https://github.com/alexfabgarcia/cold-chain-tracking>>

1. KPN\_GRIP\_APPLICATION\_ID
2. KPN\_GRIP\_CLIENT\_ID
3. KPN\_GRIP\_CLIENT\_SECRET
4. KPN\_GRIP\_TENANT\_ID
5. KPN\_THINGS\_CLIENT\_ID
6. TTN\_STACK\_APPLICATION\_ID
7. TTN\_STACK\_MQTT\_PASSWORD
8. TTN\_STACK\_TENANT\_ID
9. TTN\_STACK\_TOKEN
10. VAADIN\_PRO\_KEY
11. GOOGLE\_GEOLOCATION\_API\_KEY

As primeiras cinco variáveis de ambiente são referentes ao servidor de rede KPN Things e devem ser configuradas de acordo com a documentação da KPN<sup>5</sup>. As próximas quatro variáveis, iniciadas em “TTN\_” e que se referem ao servidor de rede TTN, podem ser configuradas de acordo com as documentações TTN API Key<sup>6</sup> e TTN MQTT API Key<sup>7</sup>. A penúltima variável se refere à licença de uso do framework Vaadin<sup>8</sup>, livre para estudantes, enquanto a última é a chave de API do Google Maps<sup>9</sup>.

### I.3 Comando de execução

Com os pré-requisitos instalados, as variáveis de ambiente definidas e em posse do código-fonte, o seguinte comando pode ser utilizado na raiz do projeto para a sua compilação e execução:

```
./mvnw spring-boot:run
```

Basta acessar o a URL <<http://localhost:8080/>> para ter acesso à aplicação em execução no ambiente local.

<sup>5</sup> <<https://developer.kpn.com/documentation/kpn-grip-api-documentation>>

<sup>6</sup> <<https://www.thethingsindustries.com/docs/api/concepts/auth/#creating-user-api-keys>>

<sup>7</sup> <<https://www.thethingsindustries.com/docs/api/concepts/auth/>>

<sup>8</sup> <<https://vaadin.com/docs/latest/flow/configuration/licenses>>

<sup>9</sup> <<https://developers.google.com/maps/documentation/geolocation/get-api-key>>

## I.4 Execução do teste de carga

Com a aplicação em execução, o teste de carga JMeter<sup>10</sup> - parte do JDK 21 - pode ser executado com o seguinte comando:

```
jmeter -t jmeter/cct-load-test.jmx
```

Os resultados podem ser acompanhados em tempo real na ferramenta de visualização de métricas através da URL <<http://localhost:3000>>.

---

<sup>10</sup> <<https://jmeter.apache.org/>>



## Anexos



---

## ANEXO A

# Artigo regular apresentado na ITNG 2023

---

A RSL que apoia este trabalho de Mestrado foi detalhada no artigo *Internet of Things Applications for Cold Chain Vaccine Tracking: A Systematic Literature Review*, apresentado no dia 25 de abril de 2023 como artigo regular na *20th International Conference on Information Technology: New Generations (ITNG 2023)*, conferência classificada como A4 de acordo com o estrato Qualis<sup>1</sup> da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) no momento da submissão. O manuscrito representa o Capítulo 37 do Livro *ITNG 2023 20th International Conference on Information Technology-New Generations*, DOI 10.1007/978-3-031-28332-1\_37.

---

<sup>1</sup> <<https://qualis.capes.gov.br>>

Alex Fabiano Garcia<sup>1</sup>, Wanderley Lopes de Souza<sup>1,2</sup>

---

## Abstract

The COVID-19 pandemic has led to an immense effort by laboratories, governments, and non-governmental organizations to develop vaccines in record time. According to the World Health Organization (WHO), one of the major challenges for global immunization is to keep the temperature conditions of these supplies in remote areas, especially in underdeveloped countries. Monitoring the cold chain, the supply chain specializing in refrigerated goods, such as food and health items, becomes critical to their preservation. Technologies inherent to the Internet of Things, like RFID and temperature sensors, are commonly used to identify and detect physical changes in the environment in which they are embedded. This survey presents a Systematic Literature Review (SLR) conducted to identify the most relevant studies on this topic from digital libraries. Through this SLR, 23 primary studies were selected from which important data were extracted and analyzed. The presented results in this study should explain which and how the key technologies are used to control the delivery of vaccines in the cold chain, focusing on remote regions.

---

## Keywords

Internet of Things, Ubiquitous and Pervasive Computing, IoT applications, Transport, Cold Chain, Vaccine, COVID-19, Pandemic preparedness, Systematic Literature Review, Survey

---

## 2086.1 Introduction

COVID-19 is a respiratory disease caused by the SARS-CoV-2 virus, which was characterized as a pandemic by the World Health Organization (WHO) on March 11, 2020, after the number of cases increased by 13 times and the number of affected countries tripled in the two weeks before that date [1]. Since this disease's emergence, governments and Non-Governmental Organizations (NGOs) have joined forces to

help pharmaceutical companies develop effective vaccines to fight this disease in record time.

The control of temperature conditions for the storage of these vaccines is vital for their usefulness and controlling the disease itself, being a challenge mainly for underdeveloped and emerging countries [2], [3]. According to [3], by February 2021, most prominent vaccine candidates required storage between 2-8°C and later proven high-effective vaccines [4], such as Pfizer-BioNTech and Moderna, even needed lower temperatures: -70°C and -20°C, respectively.

The management of the Cold Chain, a term that refers to the production, storage, and distribution of products sensitive to variations in temperature and humidity, has been explored by the WHO since 2007 to facilitate the distribution of vaccines outside the conventional range temperature, which is from 2°C to 8°C. This endeavor culminated in the development of an innovative approach called the Controlled Temperature Chain (CTC) [5]. A vaccine with the CTC label can withstand up to 40°C for at least three days before its application, which makes it possible to immunize populations in remote areas, besides making its handling cheaper, as it does not require refrigeration and facilitates transport because of the reduction in weight of the container.

Making Cold Chain management unnecessary for these biological complexes would be the most effective solution to address the difficulties imposed, but this is still one of the significant challenges of pharmaceutical research [6]. Vital supplies such as antivenom for venomous snakebites, commonly applied in remote areas of the Amazon region, also face challenges of effectiveness and safety because of the use of different modes of transport and susceptibility to failure of refrigeration equipment [7]. A similar challenge is confronted by the food sector, where it is estimated that 30% of food produced for human consumption is wasted because of failures in Cold Chain management, caused mainly by insufficient refrigeration [8].

As stated by [8], several technologies have been applied to monitor changes in the conditions of the Cold Chain, highlighting Radio Frequency Identification (RFID) and Wireless Sensor Networks (WSN). While the first refers to object identification, for example, by reading tags, the second mentions networks capable of collecting and centralizing physical data from an environment through sensors. Both technologies are part of the Internet of Things (IoT), a term

1

---

<sup>1</sup> Graduate Program in Computer Science (PPG-CC), Department of Computing (DC)  
Federal University of São Carlos (UFSCar), São Carlos, SP, Brazil  
email: alex.garcia@estudante.ufscar.br, desouza@ufscar.br

<sup>2</sup> Services and CyberSecurity (SCS), Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS)  
University of Twente (UT), Enschede, The Netherlands  
email: w.lopesdesouza-1@utwente.nl

coined by Kevin Ashton in 1999, according to which objects common to our daily lives, called "things", with integrated technologies capable of identifying, observing, intervening, and understanding the world, communicating via the Internet [9]. IoT has gained significant relevance, especially in the last decade, when there was an average annual growth of publications above 85% between 2015 and 2019 [10].

To investigate, evidence, and report the significant studies related to IoT applications for vaccine logistics in the cold chain, a Systematic Literature Review (SLR) was carried out and is detailed throughout this survey. We describe the dominant technologies found and their usage in different contexts, aiming to help researchers who may face the same problems due to the challenges imposed by diverse environments and locations during transportation.

This paper is further structured: section 2 details the method applied to search and select relevant documents; section 3 presents in a structured way the results through the extracted data; section 4 shows discussions about the results to answer the research questions correctly; finally, section 5 summarizes the relevant findings and highlights future work in regarding of this topic by the authors.

## 2086.2 Method

An SLR was elaborated to search for primary studies about the addressed topic in an organized and auditable way. According to [11], an SLR is a secondary study capable of identifying, evaluating, and interpreting relevant research. It is divided into three phases: planning, conducting, and reporting. In the first phase, the objective and protocols are defined. Afterward, the studies are collected and refined during the conduction. Finally, the results are reported in the last step.

Many tools are designed to perform an SLR within the context of Software Engineering [12]. In our case, the Parsifal (<https://parsif.al>) tool was chosen because it covers most of the stages of the systematic review process; it is open-source and online available, which enables collaborative work even if the researchers are geographically distributed.

We conducted our SLR by applying [11] guidelines through the Parsifal tool. In the planning phase, we have defined the Research Protocol (RP), the Quality Assessment Checklist (QAC), and the Data Extraction Form (DEF). During the review conduction, studies were searched, imported, selected, and qualified, with their relevant data extracted and analyzed, following the previously defined RP. Afterward, all the data collected was organized so that we could show the most relevant results of the reporting phase.

### 2086.2.1 Planning the Review

The RP defines the objective of the SLR, the research questions that guide it, the search string used for querying papers, the chosen search sources, and the inclusion and exclusion criteria applied for selecting studies.

To inspect the main issues and their current solutions involving IoT applications for cold chain vaccine tracking, our SLR sought to answer the following **Research Questions (RQ)**:

- RQ1 - What are the main IoT technologies used in cold chain vaccine logistics?
- RQ2 - Which of these technologies are applied in remote areas?

The **Search String** was defined aiming to answer both RQs, combining keywords and their synonyms with "AND" and "OR" logical operators:

("internet of things" OR "IoT" OR "pervasive computing" OR "ubiquitous computing") AND ("cold chain" OR "logistics" OR "SCM" OR "supply chain") AND "vaccine"

Pertinent **digital libraries and databases** of Computer Science and Health areas have been chosen as search sources of the RP: ACM Digital Library, IEEE Digital Library, PubMed Central: PMC, Scopus, and Springer Link.

To keep only topic-related manuscripts, the **RP inclusion criteria** were determined in such a manner:

- 1) Title, abstract, or keywords refer to cold chain vaccine logistics;
- 2) Title, abstract, or keywords are related to IoT, Ubiquitous and Pervasive Computing;
- 3) Study published between 2010 and 2021.

Dealing with the **RP exclusion criteria**, the following were defined to avoid outdated and secondary articles, besides keeping only those focused on IoT applications:

- 1) Book, chapter, or secondary study;
- 2) Complete article unavailable;
- 3) Absence of IoT application for cold chain vaccine logistics.

In addition to the RP, the **Quality Assessment Checklist** was specified in the planning phase, intending to rate the selected manuscripts containing five items. For each question of the QAC, there were three answers with their respective weight: Yes (1.0), Partially (0.5), and No (0.0). Thus, an article's maximum Quality Assessment Score (QAS) would be 5.0.

- Are the research aims specified?
- Is the study compared to related work?
- Is there hardware and/or software specification?
- Is there an IoT application detailed?
- Are the data collection and results adequately detailed?

### 2086.2.2 Conducting the Review

The SLR was conducted in July 2021 as illustrated in Fig. 1. In total, 575 studies were retrieved by submitting the aforementioned search string to the digital libraries and databases. All of them were introduced to the Parsifal tool, totaling 514 papers after 61 were classified as duplicated.

The first step of selection was reading the title and abstract of the non-duplicated manuscripts while applying the RP inclusion and exclusion criteria, which ended up in a list of 46 articles for the complete analysis. Finally, the full

reading with the same criteria was employed in the final round, with 23 selected papers for data extraction.

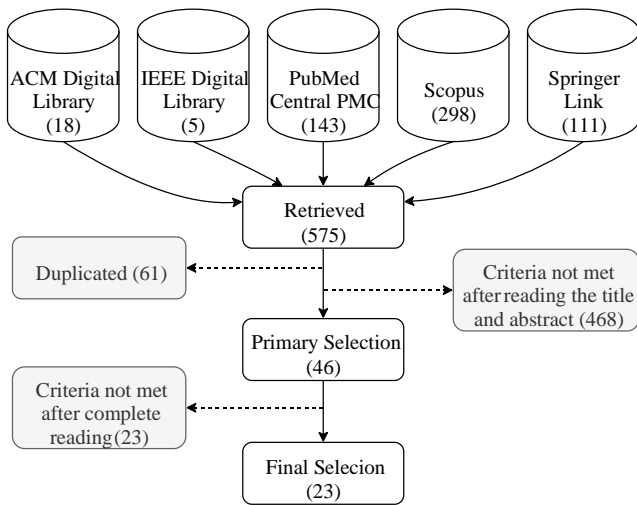


Fig. 1 Summary of the SLR conduction

**2086.2.3 Reporting the Review**

The most relevant information from the selected studies, i.e., sensors and protocols used, the local storage capability, the countries where they were carried out, and the QAS are presented in Table 1. All the manuscripts employ temperature sensors, while humidity is a secondary concern applied in 60.9%, and only 30.4% detect lighting. The communication protocol was not specified in the applications of the S14, S17, and S20 papers. Local storage was employed in 43.5% of these articles, enabling their applications to operate without an Internet connection.

Table 1 Summary of extracted Sensors, Local Storage, Protocols/Technologies, Countries, and Score from the Selected Studies

Study	Sensors				Local Storage	Communication		Quality Score
	Temperature	Humidity	Light	Protocols/Technologies		Country		
S1	✓					6LoWPAN, RFID	Spain	4.0
S2	✓	✓			✓	GSM	Albania	4.0
S3	✓	✓	✓		✓	GSM	Laos	3.5
S4	✓				✓	RFID	Brazil	3.0
S5	✓					Bluetooth	Taiwan	5.0
S6	✓				✓	3G, 4G, BLE	Colombia	5.0
S7	✓	✓	✓		✓	Bluetooth	Hong Kong	5.0
S8	✓	✓	✓			GSM, WiFi, ZigBee	India	3.5
S9	✓					LoRaWAN	Congo	4.0
S10	✓	✓				WiFi	India	3.0
S11	✓	✓				Bluetooth 3G	Taiwan	4.5
S12	✓					SigFox	Colombia	4.0
S13	✓	✓			✓	3G, 4G	United Arab Emirates	4.5
S14	✓					-	South Africa	4.5
S15	✓	✓				GSM	Bangladesh	4.0
S16	✓	✓			✓	BLE, 3G, 4G, WiFi	Hong Kong	5.0
S17	✓					-	Denmark	5.0
S18	✓	✓			✓	BLE, WiFi	Italy	3.5
S19	✓	✓				RFID	China	5.0
S20	✓				✓	-	China	4.5
S21	✓	✓	✓			BLE	Portugal	5.0
S22	✓	✓	✓			4G, WiFi	United Arab Emirates	5.0
S23	✓	✓	✓		✓	GSM	Bangladesh	5.0

The extracted data through the DEF is described in detail in the Results section, while the complete list and the detailed information of these papers, including the Digital Object Identifier (DOI), title, and authors' names, can be accessed via <https://bit.ly/3VTY6Cb>.

**2086.3 Results**

The main characteristics raised from the selected studies were: the year of publication for indicating the trend of the subject; the employed technologies for identifying, monitoring, and tracking vaccines during transport, mainly the used sensors and protocols; and the places where these studies were carried out. The SLR results for each one of these characteristics are reported in the sequel.

**2086.3.1 Time-Spreading of Selected Studies**

This SLR was conducted from May to July 2021, focusing on the last ten years. The distribution per year of the selected studies handling IoT applications for vaccine logistics is shown in Fig. 2.

From 2011 to 2017, the average of papers addressing this SLR subject was less than one per year. From 2018 to 2021, this average grew to 4,25, with peaks in 2018 and 2021. These two periods coincide, respectively, with the "development" and "rapid development" stages of the area defined by [10]. The former phase contains action plans and financing from Asian and European countries to promote the direction of the IoT, while the latter reaps the benefits of such investment.

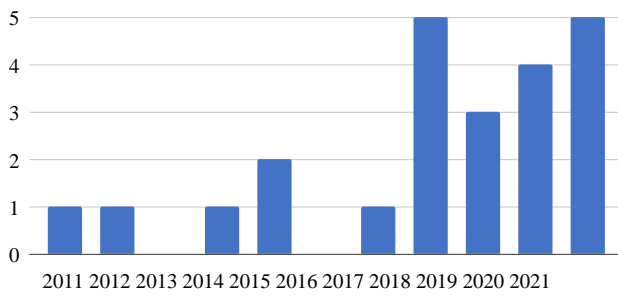


Fig. 2 Selected studies per year of publication

Apart from the previously mentioned trend in the IoT field, the high number of articles in the last years can also be explained by the demand for monitoring vaccines during their storage and transport, given that such solutions are considered a Return on Investment (ROI) by the WHO CTC Workgroup in the strategic roadmap for priority vaccines [5].

### 2086.3.2 Technologies of Selected Studies

The data relating to the employed technologies extracted from the selected studies allows us to answer RQ1. For a better explanation, the gathered technologies were divided based on the IoT three-layer architecture, i.e., devices, communication, and cloud, which, according to [13], is the most widespread Internet of Things architecture in the literature.

**2086.3.2.1 IoT Devices:** The IoT devices in the cold chain, often called tags or data loggers, are composed of at least one microcontroller and linked sensors to monitor environmental conditions. Table 2 shows the studies, among the selected ones, that specify the employed hardware, i.e., microcontroller/board and/or sensor model.

Table 2 Studies and employed hardware.

Study	Microcontroller/Board	Sensor model
S3	Fridge-Tag 30	-
S6	Raspberry Pi Family	K MAX6675
S7	SensorTag™ CC2650	-
S10	NXP LPC2148	LM35
S11	Arduino Family	DHT22
S12	PIC32MX130F064B	-
S13	Raspberry Pi Family	-
S14	-	PT-100
S18	STM32L4	-
S23	Arduino Family	-
S24	Arduino Family	-

Among the studies shown in Table 2, the most employed board family is Arduino (<https://www.arduino.cc>), mentioned in S11, S23, and S24. This open-source electronic platform abstracts low-level hardware programming and offers flexibility in prototyping, being widely adopted in the last decade [14]. It is worth mentioning the S18 cites the STM32L4 board, from STMicroelectronics (<https://www.st.com>), which is also compatible with Arduino and therefore can be classified as such.

The applications of S6 and S13 use Raspberry Pi

(<https://www.raspberrypi.org>), a single-board computer that runs Linux and provides a set of general-purpose input/output (GPIO). Together with Arduino, it is a highly popular platform for developing IoT applications [15].

NXP LPC2148, from NXP Semiconductors (<https://www.nxp.com>), is the microcontroller of the S10 application. It is small, has low power consumption, and is suitable for industrial control and medical systems. Finally, S3 and S7 embrace commercial solutions for continuous cold chain monitoring, including the data loggers Fridge-Tag 30 and SensorTag CC2650, respectively.

Only four sensor models were explicitly specified in the selected studies. The temperature sensors K MAX6675, DHT22, and PT-100 were employed in S6, S11, and S14 applications, respectively, while S10 used the LM35 sensor to track temperature and humidity.

**2086.3.2.1 Communication:** The network plays a crucial role in IoT by enabling communication between devices and back-end applications. In Fig. 3, are illustrated the wireless communication technologies extracted from the studies and their respective geographic scope, while they are grouped by the publication's year in Fig. 4.

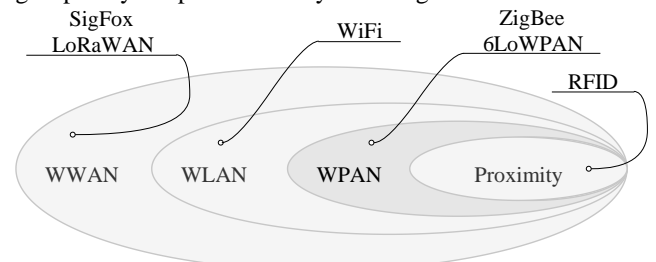


Fig. 3 Wireless technologies extracted and their geographic coverage. Adapted from [13]

SigFox, LoRaWAN, and mobile networks are Wireless Wide Area Networks (WWAN), with regional or even nationwide coverage. The Global System for Mobile Communications (GSM) and the following generations (i.e. 3G and 4G) are present in over 40% of the selected papers, a fact that can be explained by their dominant role in the IoT area given the existing cellular infrastructure [13]. SigFox and LoRaWAN cited once in S12 and S8 respectively, are said low power WAN due to their low energy characteristic despite the long range. The former requires a subscription to a licensed wireless provider, while the latter permits owning the communication infrastructure [16].

Cited in 15.9% of the chosen studies, WiFi, based on the IEEE 802.11 family [17], is the most widespread Wireless Local Area Network (WLAN), usually used within a home or a building. ZigBee and 6LoWPAN, correspondingly to S8 and S1, along with Bluetooth and its low-energy evolution (Bluetooth Low Energy - BLE) adopted by 21.9% of papers, are Wireless Personal Area Network (WPAN), which means they should work in the personal space from 10 up to 100 meters [13].

Radio-frequency identification (RFID), mentioned by

9.4% of the gathered IoT applications, is usually composed of an RFID tag which can store an identification through transponders. When it has its own power source, it is considered an active tag, otherwise, it is a passive tag and can be activated by an electromagnetic signal emitted by an RFID reader [18]. Passive RFID tags are inexpensive and, along with the capacity of identifying something, it is a flashy technology in the IoT area.

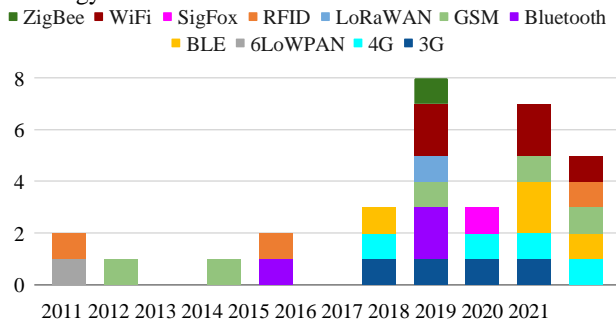


Fig. 4 - Protocols per year

Apart from the aforementioned Wireless Sensor Network (WSN) technologies, summarized by [13], the gathered manuscripts also mention the usage of the Hypertext Transfer Protocol (HTTP) and Message Queue Telemetry Transport (MQTT) as the application layer protocols [19]. S1, S3, S6, and S18-22 uses HTTP or its secure version (HTTPS) to exchange data between client and servers via its request and response model, while, S08, S14 and S22 employed the MQTT, a lightweight publish/subscribe messaging protocol, vastly adopted in the IoT industry [20].

**2086.3.2.1 Cloud:** The National Institute of Standards and Technology (NIST) defines [21] Cloud Computing as *a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources*. Its cost-based model enables end-to-end services for users to access applications on demand from anywhere.

Among the selected papers, two of them explicitly specify the usage of cloud providers. Without providing further implementation information, the study S07 relies on the IBM Watson IoT service to register the IoT device and exchange real-time collected data. The authors of the S22 study stated the Microsoft Azure IoT Hub was selected because it can be integrated with other used Azure services seamlessly. It supports MQTT, HTTPS, and other application protocols.

### 2086.3.3 Places of Selected Studies

The places stated in Table 1 were firstly defined whenever explicitly specified in the study, and secondly for being the leading country of the carried-out investigation. In Fig. 5, the countries painted in dark green speak for 2 studies each, while a light green painted country stands for 1 study. Asian countries were responsible for 56,5%, followed by 21,7% from Europe, 13% from South America, and 8,7% from Africa.

Regarding the remote area's issues, five papers explicitly cite the major challenges, which can be categorized as (a)

lack of network connectivity and (b) energy outages. In S2 (Albania), the solution relies on the mobile phone battery against the common transitory power supplies. S23 (Bangladesh) focus on the low-energy consumption of the proposed prototype as its local storage capability to overcome power problems. S14 (South Africa) uses solar panels as the power source for their solution to monitor the vaccine box. S9 (Congo) applies solar energy as well, but it goes further and experiments with LoRaWAN to create a sequence of wireless devices to form arbitrarily large mesh networks, disconnected from the Internet.

## 2086.4 Discussion

As presented in the results got with the SLR, a wide variety of technologies can be applied in cold chain vaccine logistics in different places throughout the world. Based on this information, this section aims to answer both research questions defined in this SLR.

### 2086.4.1 RQ1: What are the main IoT technologies used in cold chain vaccine logistics?

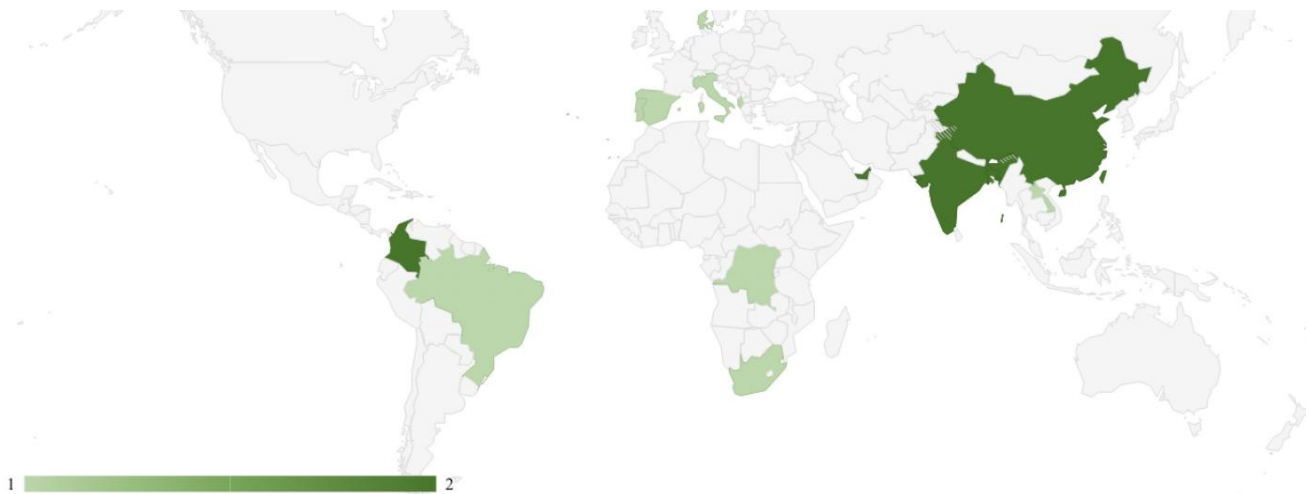
A few sensor types are employed to gather the environmental conditions, especially for temperature. The collected input is usually processed in loco by the board, highlighting the choice of Arduino and Raspberry Pi. Hence, the data is transferred through wireless communication to a gateway or directly to the cloud. Mobile networks, such as GSM/3G/4G, are heavily employed throughout the years. SigFox and LoRa(WAN) are next-generation technologies employed for long-distance communication. Meanwhile, RFID, Bluetooth/BLE, and ZigBee are used for shorter-range communication. Cloud services such as IBM Watson IoT and Azure IoT Hub offer further data processing and analysis, supporting different application protocols for integration as HTTP/S and MQTT.

### 2086.4.2 RQ2: Which of these technologies are applied in remote areas?

In rural or remote places, even nowadays cellular networks may be intermittent or do not exist at all [22]. In such scenarios, a hybrid solution using the device's local storage and WSN disconnect from the Internet is commonly applied, like SigFox and LoRa(WAN) for long-distance communication. Meanwhile, RFID, Bluetooth/BLE, and ZigBee are used for shorter-range link. Other key impact is the non-reliable power supply in such areas, which could be mitigated by employing low-power devices and protocols, besides batteries or sustainable power providers as solar panels.

## 2086.5 Conclusion

This paper is focused on cold chain vaccine logistics by leveraging IoT technologies. The aim was to understand the key technologies and their correlation with remote places. In the elaborated SLR, the last decade's primary studies which addressed this problem via IoT applications were gathered and scrutinized via defined criteria.



**Fig. 5** Studies conducted worldwide: countries painted in dark green published 2 studies, while countries in light green contributed to 1 study each.

It was possible to answer the RQ1 and RQ2 from the selected studies' information. Technologies-wise, following the three-layer IoT architecture, we highlighted the role of the IoT devices such as sensors, microcontrollers, and boards, a wide variety of wireless communication means, and cloud services when applied.

For remote areas' concerns, the lack of connectivity and the energy consumption of the solution were shown as the fundamental challenges. Strategies like local storage help to dodge the former, while next-generation and low-energy communications protocols address the former.

Policymakers can make decisions based on technologies and issues highlighted by this study, along with understanding the available vaccine's characteristics. Vaccines that require fine-grained temperature control may be suitable for urban areas, where power supply and connectivity are much more reliable. On the other hand, near real-time IoT solutions would improve the distribution of such goods in rural areas, mitigating the waste of resources and, most importantly, saving lives.

The presented SLR was performed during the Problem Investigation phase of the Master's degree project "System based on the Internet of Things for Intelligent Tracking applied to Cold Chain Logistics", under development by the first author of this paper. The co-author has guided the entire writing and reviewing processes for this article.

## References

- [1] W. W. D.-G. Opening, "Remarks at the media briefing on covid-19—11 march 2020," mar 2020. [Online]. Available: <https://www.who.int/director-general/speeches/detail/who-director-general-sopening-remarks-at-the-media-briefing-on-covid-19---11-march-2020>
- [2] Y. Yan, Y. Pang, Z. Lyu, R. Wang, X. Wu, C. You, H. Zhao, S. Manickam, E. Lester, T. Wu *et al.*, "The covid-19 vaccines: recent development, challenges and prospects," *Vaccines*, vol. 9, no. 4, p. 349, 2021.
- [3] O. J. Wouters, K. C. Shadlen, M. Salcher-Konrad, A. J. Pollard, H. J. Larson, Y. Teerawattananon, and M. Jit, "Challenges in ensuring global access to covid-19 vaccines: production, affordability, allocation, and deployment," *The Lancet*, vol. 397, no. 10278, pp. 1023–1034, 2021.
- [4] C. Zheng, W. Shao, X. Chen, B. Zhang, G. Wang, and W. Zhang, "Realworld effectiveness of covid-19 vaccines: a literature review and metaanalysis," *International Journal of Infectious Diseases*, vol. 114, pp. 252–260, 2022.
- [5] W. H. Organization *et al.*, "Controlled temperature chain: strategic roadmap for priority vaccines 2017-2020," World Health Organization, Tech. Rep., 2017.
- [6] Y. B. Yu, K. T. Briggs, M. B. Taraban, R. G. Brinson, and J. P. Marino, "Grand challenges in pharmaceutical research series: ridding the cold chain for biologics," *Pharmaceutical Research*, vol. 38, no. 1, pp. 3–7, 2021.
- [7] H. W. Fan and W. M. Monteiro, "History and perspectives on how to ensure antivenom accessibility in the most remote areas in brazil," *Toxicol*, vol. 151, pp. 15–23, 2018.
- [8] R. Badia-Melis, U. Mc Carthy, L. Ruiz-Garcia, J. Garcia-Hierro, and J. R. Villalba, "New trends in cold chain monitoring applications-a review," *Food Control*, vol. 86, pp. 170–182, 2018.
- [9] K. Ashton *et al.*, "That 'internet of things' thing," *RFID journal*, vol. 22, no. 7, pp. 97–114, 2009.
- [10] J. Wang, M. K. Lim, C. Wang, and M.-L. Tseng, "The evolution of the internet of things (iot) over the past 20 years," *Computers & Industrial Engineering*, vol. 155, p. 107174, 2021.
- [11] S. Keele *et al.*, "Guidelines for performing systematic literature reviews in software engineering," Technical report, ver. 2.3 ebse technical report. ebse, Tech. Rep., 2007.
- [12] C. Kohl, E. J. McIntosh, S. Unger, N. R. Haddaway, S. Kecke, J. Schiemann, and R. Wilhelm, "Online tools supporting the conduct and reporting of systematic reviews and systematic maps: a case study on cadima and review of existing tools," *Environmental Evidence*, vol. 7, no. 1, pp. 1–17, 2018.
- [13] W. Kassab and K. A. Darabkh, "A-z survey of internet of things: Architectures, protocols, applications, recent advances, future directions and recommendations," *Journal of Network and Computer Applications*, vol. 163, p. 102663, 2020.
- [14] H. K. Kondaveeti, N. K. Kumaravelu, S. D. Vanambathina, S. E. Mathe, and S. Vappangi, "A systematic literature review on prototyping with arduino: Applications, challenges, advantages, and limitations," *Computer Science Review*, vol. 40, p. 100364, 2021.
- [15] A. Polianytzia, O. Starkova, and K. Herasymenko, "Survey of hardware iot platforms," in *2016 Third International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T)*. Kharkiv, Ukraine: IEEE, 2016, pp. 152–153.
- [16] P. W. Lawrence, T. M. Phippard, G. S. Ramachandran, and D. Hughes, "Developing the iot to support the health sector: A case study from kikwit, dr congo," in *International Conference on Emerging Technologies for Developing Countries*. Springer, 2017, pp. 45–56.

- 
- [17] B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai, "Ieee 802.11 wireless local area networks," *IEEE Communications magazine*, vol. 35, no. 9, pp. 116–126, 1997.
  - [18] R. Weinstein, "Rfid: a technical overview and its application to the enterprise," *IT professional*, vol. 7, no. 3, pp. 27–33, 2005.
  - [19] H. Zimmermann, "Osi reference model-the iso model of architecture for open systems interconnection," *IEEE Transactions on communications*, vol. 28, no. 4, pp. 425–432, 1980.
  - [20] T. Yokotani and Y. Sasaki, "Comparison with http and mqtt on required network resources for iot," in *2016 international conference on control, electronics, renewable energy and communications (ICCEREC)*. IEEE, 2016, pp. 1–6.
  - [21] P. Mell, T. Grance *et al.*, "The nist definition of cloud computing," 2011.
  - [22] A. M. Cavalcante, M. V. Marquezini, L. Mendes, and C. S. Moreno, "5g for remote areas: Challenges, opportunities and business modeling for brazil," *IEEE Access*, vol. 9, pp. 10829–10843, 2021.

---

## ANEXO B

# Artigo regular aceito pela ICEIS 2025

---

O projeto e desenvolvimento do SBIdC-RALCF foi tema do manuscrito *A LoRaWAN Multi-Network Server Application for Smart Cold Chain Tracking in Remote Areas*, aceito como artigo regular pela conferência *27th International Conference on Enterprise Information Systems (ICEIS 2025)*.

# A LoRaWAN Multi-Network Server Application for Smart Cold Chain Tracking in Remote Areas

Alex Fabiano Garcia<sup>1</sup><sup>a</sup>, Wanderley Lopes de Souza<sup>1</sup><sup>b</sup> and Luís Ferreira Pires<sup>2</sup><sup>c</sup>

<sup>1</sup>*Ubiquitous Computing Group, Graduate Program in Computer Science, Computing Department, Federal University of São Carlos, P.O. Box 676, São Carlos, Brazil*

<sup>2</sup>*Semantics, Cybersecurity & Services Group, Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, P.O. Box 217, Enschede, The Netherlands*  
*alex.garcia@estudante.ufscar.br, desouza@ufscar.br, lferreirapires@utwente.nl*

**Keywords:** Internet of Things, LoRaWAN, Transport, GPS-free, Multilateration, Cold Chain, Perishable Goods.

**Abstract:** This paper describes the design and implementation of a multi-network server application for smartly tracking cold chain based on the LoRaWAN technology. The system aims to solve logistics challenges in remote areas by using IoT technologies to monitor and manage the conditions of perishable goods during transportation, ensuring their quality and safety. The proposed solution encompasses various sensor types and integrates multiple network providers to improve coverage, aiming to support decision-makers in public-private partnerships when addressing social issues in outlying regions. The study compares the simulation of antenna coverage with the effective distance from the gateway that receives the signal. Field tests in the Netherlands demonstrate the system's effectiveness in real-world scenarios, showing features such as GPS-free geolocation by multilateration, long-range communication, and the potential for applying our solution in other domains beyond cold chain logistics.

## 1 INTRODUCTION

The term Internet of Things (IoT), coined by Kevin Ashton in 1999 (Ashton et al., 2009), refers to a network of pervasive devices embedded with sensors, software, and other technologies to collect and exchange data. IoT significantly transformed various application domains, such as smart homes, industrial automation, healthcare, and environmental monitoring, by enabling unprecedented levels of efficiency, automation, and data-driven decision-making (Atzori et al., 2010).


One critical application domain of IoT is cold chain logistics, where IoT solutions help ensure the integrity and safety of temperature-sensitive products such as pharmaceuticals, food, and chemicals. IoT enhances cold chain management by providing real-time monitoring and data analytics, which in turn helps maintain product quality and comply with regulatory standards (Badia-Melis et al., 2018; Aung and Chang, 2014).


The cold chain logistics sector faces significant


challenges, particularly in remote areas where maintaining the required temperature conditions for sensitive products is arduous due to energy restrictions and lack of network connectivity (Badia-Melis et al., 2018). For example, the antivenom distribution in the Amazon region (Fan and Monteiro, 2018) is a concrete case in which cellular connectivity is often sporadic or non-existent, hindering real-time monitoring and data transmission, and leading to delays in identifying and addressing temperature oscillations. The Internet penetration in Brazilian rural areas is around 34%, compared to 65% of the urban regions, mainly due to the high investment per covered inhabitant and revenue uncertainty from Mobile Network Operators (MNO) (Cavalcante et al., 2021). Addressing these challenges requires innovative solutions that operate efficiently under energy constraints and provide reliable communication in areas with limited network infrastructure.

Designed for low-power wide-area communication, LoRaWAN (Long Range Wide Area Network) is a network protocol that enables long-range data transmission with minimal energy consumption, making it appropriate for IoT applications in locations with limited access to power and connectivity (Centenaro

---

<sup>a</sup> <https://orcid.org/0000-0002-6202-1849>

<sup>b</sup> <https://orcid.org/0000-0001-8645-4393>

<sup>c</sup> <https://orcid.org/0000-0001-7432-7653>

et al., 2016). Its robust network architecture operating in a license-free band allows for scalable deployment, accommodating thousands of connected devices and facilitating comprehensive environmental monitoring in remote cold chain logistics (Vangelista et al., 2015). Additionally, the total daily cost for an increasing number of devices is lower than that of cellular protocol alternatives due to their subscription costs and private infrastructure (Frangoudis et al., 2021).

In our research, we identified gaps in related designs that employ LoRaWAN in cold chain logistics scenarios, developed a multi-network server application that aims to fill some of these breaches in remote locations, and deployed and evaluated a prototype of this application in the field. In our multi-network approach, in which several LoRaWAN network servers are applied, we sought to enhance network coverage by leveraging existing infrastructure provided by the public LoRaWAN Network Operator (LoRa Alliance, 2024) and private networks. Moreover, we applied the LoRaWAN geolocation capabilities without using Global Positioning Systems (GPS).

This paper is further structured as follows: Section 2 details our research methodology; Section 3 describes the most relevant related work; Section 4 presents an overview of the leading technologies employed in this work; Section 5 describes the design of the system developed in this study; Section 6 presents tests and validation of our system; Section 7 concludes the paper by discussing our contributions, the remaining challenges, and some topics for future work.

## 2 RESEARCH METHODOLOGY

In this research we followed the Design Science Methodology (DSM) (Wieringa, 2014). Figure 1 shows the framework used, which considers two main activities: (a) designing an artefact and (b) empirically investigating its effects in a problem context. By building a multi-network LoRaWAN application, we aim to improve cold chain tracking while satisfying long-range communication, condition breach detection, and GPS-free geolocation. Our application aims to help governments establish public-private partnerships (PPP), investing in private LoRaWAN deployments while taking advantage of existing infrastructure from different network providers, enabling continuous cold chain tracking in remote areas with a cost-effective technology (Frangoudis et al., 2021).

Our artefact has been conceived in a Design Cycle consisting of three tasks: 1. *Problem Investigation*: to capture stakeholder objectives and identify

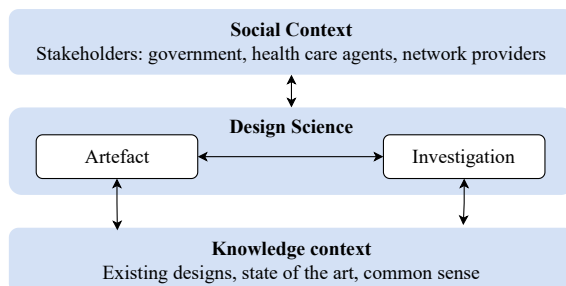


Figure 1: A Framework for Design Science (adapted from (Wieringa, 2014)).

the challenges faced in tracking the cold chain in remote regions; 2. *Treatment Design*: to design the LoRaWAN application to improve cold chain tracking in remote areas with sparse network coverage; and 3. *Treatment Validation*: to evaluate the effects caused by our LoRaWAN application in the context of cold chain tracking. The Treatment Validation task could trigger a new iteration of the Design Cycle, as shown in Figure 2. We used an agile development approach to gradually design, implement, and evaluate our application throughout the design cycle.

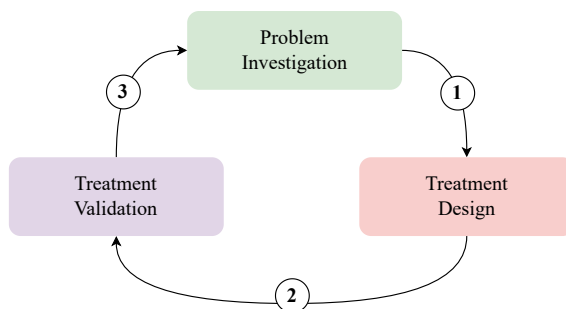


Figure 2: The Design Cycle tasks.

According to (Wieringa, 2014), the Design Cycle is part of a more comprehensive flow called the Engineering Cycle, which also encompasses the additional Treatment Implementation and Treatment Evaluation tasks to validate the application of the artefact in a real-world scenario and use and evaluate it with the interested parties. Since we only validated our application in experimental settings, these tasks were out of the scope of our research project.

In the Problem Investigation task, we first conducted a Systematic Literature Review (SLR) (Garcia and de Souza, 2023) of IoT applications and technologies for cold chain tracking. In this SLR, we identified the lack of network infrastructure that supports the distribution of cold chain goods, especially due to the low Return on Investment (ROI) for telecom companies, considering the population density in remote areas. Moreover, we also learned how low-power

wide-area network (LPWAN) technologies such as LoRaWAN offer appropriate capabilities in resource-constrained contexts. Next, we conducted a comprehensive literature review of this technology to identify its potential role as part of a solution. Although many applications employ this technology, they do not exploit some aspects, especially the hybrid networking capability.

In the Treatment Design task, we developed a multi-network LoRaWAN application named Cold Chain Tracking (CCT), which includes hardware prototypes, the communication layer, and a web application. The Treatment Validation comprises a Single-Case Mechanism and a Scale-Up Experiment (Wieringa, 2014). We first tested our application in the field with different routes and transport modes, validating the fulfilment of the requirements defined before. Finally, we scaled up to more realistic conditions via load tests to assess the robustness of the application.

### 3 RELATED WORK

Our previous study (Garcia and de Souza, 2023) highlights several IoT applications and technologies for cold chain vaccine tracking through an SLR. Among them, in (Lawrence et al., 2018) LoRaWAN and IoT are applied to support the health sector in Kikwit, Democratic Republic of the Congo. The authors implement a mesh network in a real-world application, demonstrating the suitability of the protocol in resource-limited contexts and areas with poor infrastructure. They mention the difficulties in finding suitable high-altitude locations to install gateways in an ad-hoc manner and the lack of financial support at national or regional levels to expedite their deployment.

In (Zinas et al., 2017), the authors report on a protocol and application based on the LoRaWAN technology for cattle tracking, implemented and tested in Pogoniani, Greece. The device prototype includes sensors, a GPS module, a LoRa transceiver, and a microcontroller enclosed in a box and placed in the cows' bell collar. During their experiments, the authors could send data over LoRaWAN to a distance of 6 km.

Focusing on maintaining optimal temperature conditions, in (Enriko et al., 2022) an IoT-based approach to monitoring vaccine quality during distribution in Indonesia is presented. The approach integrates various IoT devices and sensors, ensuring continuous monitoring and immediate alerts in case of deviations from the required conditions. Due to the network coverage, the proposed architecture employs

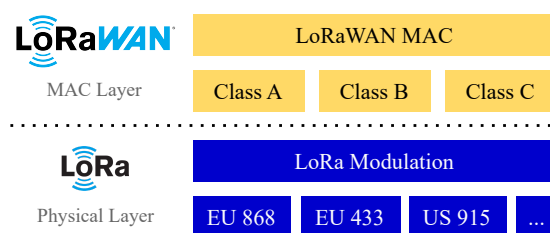


Figure 3: LoRa and LoRaWAN network protocol layers.

LoRaWAN only for at-rest nodes while using cellular technologies for in-transit devices.

In (Bose et al., 2022), a LoRaWAN-based vaccine cold storage monitoring system is proposed that comprises an end node with a GPS module, LoRaWAN gateway, network server, and a web-based user application interface. The authors tested their design with a stationary gateway placed at a high altitude in Bangalore, India. Their presented results indicate that the protocol is noise-resistant for long-range communications in urban areas.

Our paper addresses some of the gaps from the related works by enabling a multi-LoRaWAN network to enhance the coverage, showing a method for antenna placement rather than ad-hoc attempts, and using the cheaper LoRaWAN GPS-free geolocation by multilateration (Fargas and Petersen, 2017).

## 4 IOT-RELATED TECHNOLOGIES

This section introduces some aspects of LoRa (Long Range), LoRaWAN, and MQTT (Message Queuing Telemetry Transport), which are the most relevant IoT-related technologies we used to develop our application.

### 4.1 LoRa and LoRaWAN

LoRa and LoRaWAN complement each other to define a Low-Power Wide Area Networks (LPWAN) protocol. Figure 3 shows that LoRa refers to the physical layer, which applies a wireless modulation technique based on Chirp Spread Spectrum (CSS) that enables long-range communication with minimal power consumption. This modulation allows for robust data transmission over several kilometres and operates on the license-free sub-Gigahertz bands, such as 915 MHz, 868 MHz, and 433 MHz, depending on the region (Augustin et al., 2016).

On its turn, LoRaWAN is the Media Access Control (MAC) layer protocol that sits on top of LoRa. It defines the communication protocol and system ar-

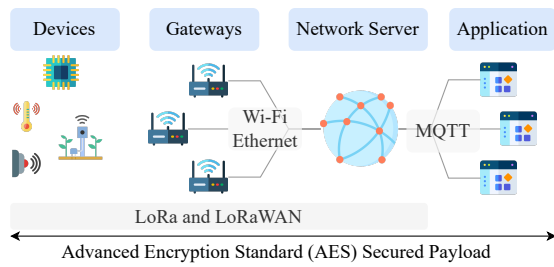


Figure 4: LoRaWAN architecture layers and technologies.

chitecture for orchestrating the network operation, including how devices connect and transmit data, ensuring security and scalability (Cattani et al., 2017). Since the protocol operates on regional license-free bands, governments often regulate the duty cycle of radio devices, which indicates the fraction of time a resource is allowed to be busy.

LoRaWAN defines three classes of devices to address different application needs regarding communication latency and energy efficiency: Class A devices have the lowest power consumption, are designed for battery-operated sensors, and allow two short receive windows following each uplink transmission; Class B devices add scheduled receive slots, using a synchronised beacon from the gateway to open additional receive windows at set times, balancing power consumption with more frequent downlink opportunities; Class C devices are nearly always listening for downlinks, offering minimal latency at the cost of higher power consumption, suitable for applications where immediate response is critical like control systems or actuators (Augustin et al., 2016; Cattani et al., 2017).

## 4.2 LoRaWAN Architecture

Figure 4 illustrates a typical LoRaWAN architecture, which is composed of end devices, gateways, network servers, and applications. The end devices contain sensors or actuators and send and receive LoRa-modulated wireless messages to the gateways, which are connected to the Network Server through a backhaul such as Wi-Fi, Ethernet, cellular networks or satellite. The Network Server manages all the LoRaWAN network components and interfaces the data exchange with the application where the business logic resides, usually through MQTT brokers.

The Network Service includes the software responsible for handling device activation in the network, the Join Server, which allows the end nodes to connect using either Over-The-Air-Activation (OTAA) or Activation By Personalisation (ABP) (Augustin et al., 2016). The former is a preferred method, where the device sends a join request to the network server, which then responds with session keys used

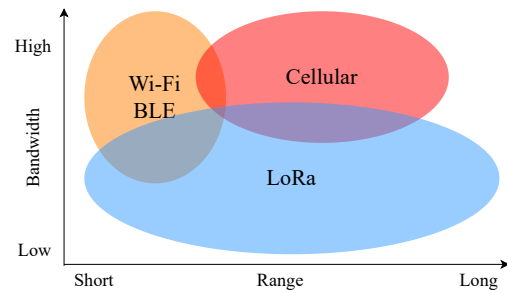


Figure 5: LoRaWAN bandwidth and range compared to other wireless networks (adapted from The Things Network<sup>1</sup>).

to encrypt future communications, allowing dynamic rejoining and better security. In contrast, the latter involves pre-configuring devices with network parameters like session keys and device addresses before deployment, enabling immediate communication but lacking the dynamic security and flexibility of OTAA.

Enterprise Network Servers such as The Things Stack (TTS)<sup>1</sup> offer plug-and-play capabilities for custom devices and gateways to operate alongside each other. Telecom providers usually include gateways as part of the solution, with features built on top of that such as GPS-free geolocation via triangulation, trilateration, and multilateration (Fargas and Petersen, 2017). In contrast, open-source alternatives like ChirpStack<sup>2</sup> enable complete private network operations attending the LoRaWAN specifications.

## 4.3 Bandwidth and Range

Figure 5 shows that wireless networks such as Wi-Fi and Bluetooth (including its Low Energy version BLE) have short range but high bandwidth, making them ideal for video and voice package transmissions. Cellular technologies are intended for mission-critical outdoor use cases and demand more power. More IoT-oriented cellular protocols, such as NB-IoT (Narrowband Internet of Things) and LTE-M (Long-Term Evolution for Machines), are energy-efficient in favourable conditions, however, they still require additional chip investment and mobile infrastructure in licensed spectrum (Labdaoui et al., 2023). In contrast, LoRa modulation aims to achieve long-range with low power consumption with the cost of low bandwidth, which is suitable for sensors and actuators that do not require big payload messaging. LoRa modulation has six Spreading Factors (SF), from SF7 to SF12, which influence data rate, time-on-air, battery life, and receiver sensitivity (Augustin et al., 2016).

<sup>1</sup><https://www.thethingsindustries.com>

<sup>2</sup><https://www.chirpstack.io>

## 4.4 Line of sight

LoRaWAN performance is significantly impacted by line-of-sight (LoS), as the technology relies on low-power, long-range radio frequency communications. With a clear distance between the transmitter and the receiving gateway, LoRaWAN can achieve long range and reliability, often extending several kilometres in rural areas, but can be blocked by obstacles such as buildings and trees, causing attenuation in the communication channel. Urban area deployments require multiple gateways or a routing mechanism to ensure robust network coverage and data transmission due to the constrained LoS (Saban et al., 2021).

Figure 6a has been taken from TTN Mapper<sup>3</sup> and shows the signal heatmap measured in decibels per milliwatt (dBm) of a gateway placed at a 136m-high tower in Hilversum, the Netherlands. Figure 6b shows a simulation of a 902 MHz antenna placed at the same point and altitude using the Radio Mobile Online tool (Roger Coudé VE2DBE, 2024). The correlation between the expected coverage and the effective transmission of the gateway indicates the simulation as a mapping alternative to ad-hoc antenna placement (Lawrence et al., 2018) and how this point of access benefits from the LoS due to the flat topography of the Netherlands.

## 4.5 MQTT

MQTT is an application layer protocol over TCP (Transmission Control Protocol) specified to be simple to implement, bandwidth-efficient, and provide different levels of Quality of Service (QoS) data delivery (Mishra and Kertesz, 2020). The protocol uses the publish-and-subscribe model, allowing bidirectional communication between millions of clients via an intermediate broker. After a client establishes an acknowledged connection with the MQTT broker, it can publish messages on topics, which are represented by hierarchical keys containing multiple levels separated by slashes. Clients can connect and subscribe to these topics on the broker, which tracks subscribers and forwards messages to them.

MQTT defines three levels of QoS: Level 0 means "at most one delivery", which is useful when occasional data loss due to the lack of confirmation is acceptable; Level 1 should be used if the message requires confirmation but allows duplication since it provides "at least one delivery"; Level 2 has the highest quality level and provides "exactly once" message delivery, which is recommended for critical applications where data loss and duplication are unaccept-

<sup>3</sup><https://ttnmapper.org>

able. Higher QoS levels increase resource consumption, such as storage and network traffic, due to the additional steps necessary for confirmation. The protocol also provides features to enhance reliability and security, such as persistent messages and encryption through TLS (Transport Layer Security).

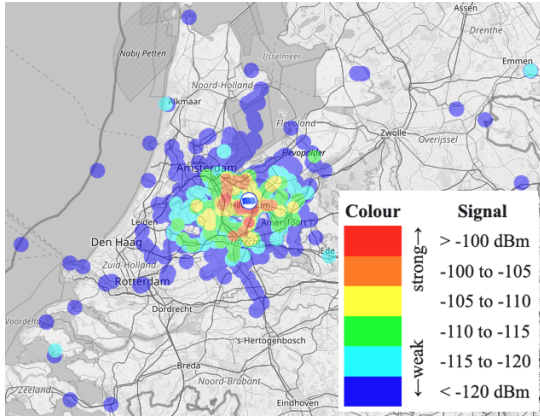
## 5 DESIGN

In this section, we discuss the design of a multi-network server LoRaWAN artefact, which we called Cold Chain Tracking (CCT), to address the challenges imposed by the cold chain context while tracking perishable goods in remote areas. The stakeholders can check the products' measured conditions since the end device constantly monitors them during freight. The system also provides essential notifications in case of violations, even if the carrier is not connected to the Internet.

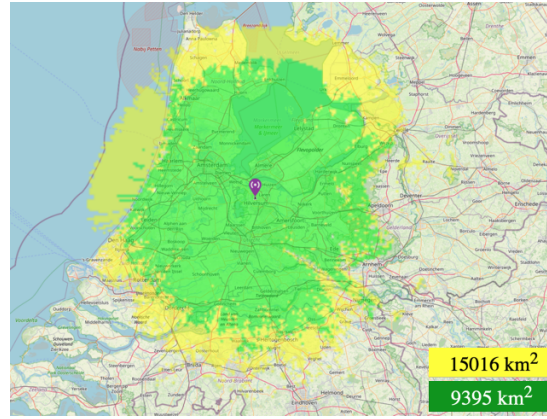
The most valuable novel aspect of this proposal is the support to multiple Network Servers. Our solution expands the coverage area and is beneficial in large-scale deployments in remote zones, ensuring that all devices remain within range of a network server. This potential redundancy helps maintain consistent data transmission and reception, which is critical for applications like cold chain logistics. On top of that, we guarantee the interoperability of the end nodes and how to decouple the application layer from different networks. With these capabilities, this approach offers an alternative for governments and decision-makers to opt for a mixed solution to public-private partnerships (PPP). It enables partnerships with established telecommunications providers to be leveraged and employ private networks in areas where the coverage is not profitable for them.

### 5.1 Capabilities

Figure 7 illustrates the leading design capability groups of our approach: management, monitoring, alerts, tracking, and security. The management capability encompasses CRUD (Create, Read, Update, and Delete) operations of the main application entities, including measurement types, devices, carriers, products, and freights. Conditions measured by the device sensors, such as temperature and humidity, are decoded, stored, and monitored by the application. CCT identifies and emits alerts for any product condition violation. The tracking capability is based on the LoRaWAN location using the multilateration feature when available. Apart from the encrypted data exchange between the system components, the secu-



(a) Signal heatmap of gateway transmission from TTN Mapper.  
Figure 6: Heatmap of effective transmissions and coverage simulation for gateway placed at a 136 m tower in Hilversum, NL.



(b) 902 MHz antenna coverage simulation.

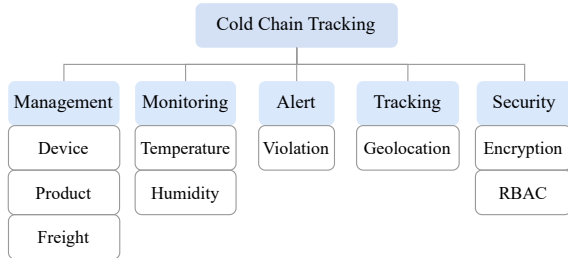


Figure 7: Cold Chain Tracking main capabilities.

ity capability defines the role-based access control (RBAC) for the two main actors (Administrator and Carrier).

Figure 8 shows how the CCT components interact to monitor a freight and alert condition violations. Assuming devices are registered to the Network Server and the CCT application has subscribed to their measurements, the administrator creates a freight by setting up the product details and condition limits, the carrier who will deliver it, the available device to monitor the product conditions, and the origin and destination. The Carriers can list their available freights, choose one of them to start, and turn on the device, which immediately tries to join the server. The device then sends the measurement data to the gateways via LoRa wireless modulation, which forward the received messages to their respective Network Servers. The CCT application then listens to and stores these events, verifying any thresholds violated and sending a downlink in case of a violation. The device then issues an alert to the carrier.

## 5.2 Architecture

According to the DSM, an architecture structure is a conceptual framework in which each system is an entity that can be decomposed into components

that interact to produce overall system behaviour. It supports case-based research, in which we investigate individual cases, study their architecture, identify mechanisms by which overall system-level phenomena are produced, and generalise case by case.

The architecture presented here is similar to the typical LoRaWAN architecture: the CCT system comprises end devices that send uplink messages data using LoRa wireless modulation to gateways connected through backhaul to the Network Server. Figure 9 shows that we adopted a multi-server model instead of a single network server to better align with our design problem. Each provider exposes the device messages to CCT via MQTT topics and provides an HTTPS (Hypertext Transfer Protocol Secure) API (Application Programming Interface) that allows bidirectional communication. Finally, the web application subscribes to the devices' messages via the MQTT broker, storing important information in a PostgreSQL database. The web application uses the Network Server's APIs to send downlink commands to the end nodes when a violation occurs.

We opted for using two network servers that allow the registration and retrieval of devices in the Netherlands, where we carried out this study. KPN Things<sup>4</sup>, which is an IoT enterprise solution from the Dutch telecommunications company KPN, was chosen as Network Server due to its high coverage and its device geolocation capability using multilateration (KPN, 2024c). The Things Network (TTN)<sup>5</sup>, which is a community network operated by TTS, was chosen to simulate a private deployment since it contains community gateways, thus less coverage, and allows the addition of custom gateways. TTN applies a Fair Use Policy (FUP), which limits the uplink airtime to

<sup>4</sup><https://portal.kpnthings.com>

<sup>5</sup><https://www.thingsnetwork.org>

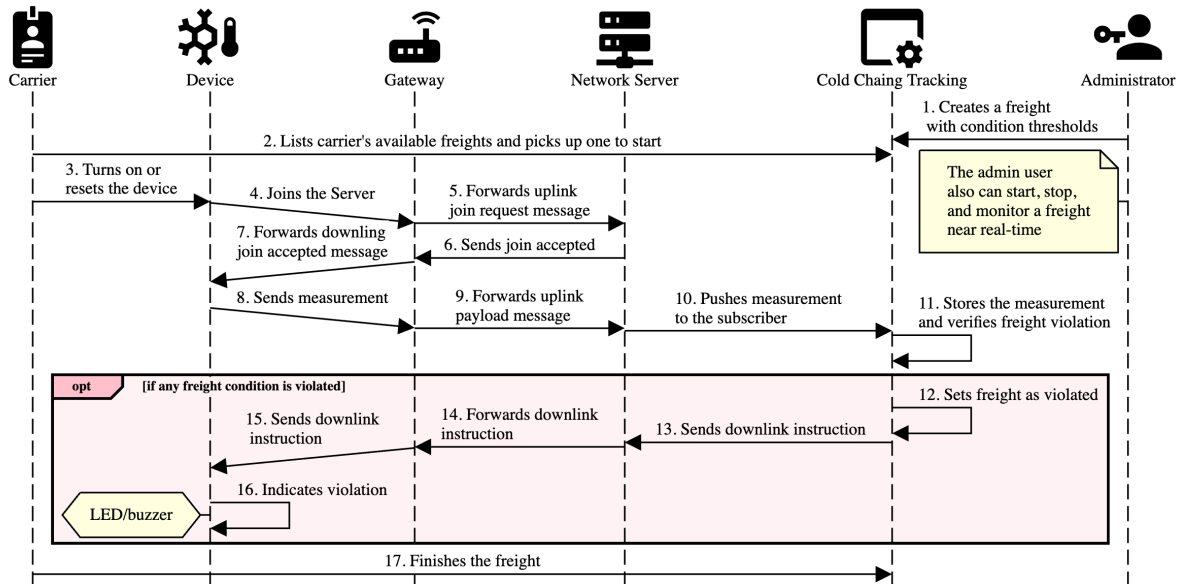


Figure 8: Sequence diagram of the freight feature.

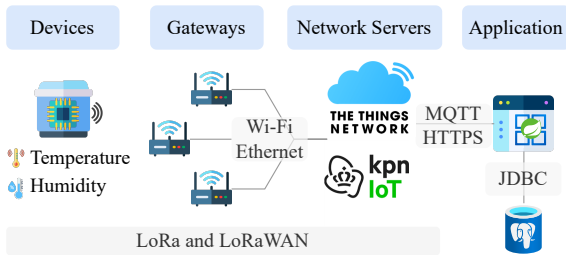


Figure 9: Cold Chain Tracking Architecture diagram.

30 seconds per day per node and the downlink messages to 10 messages per day per node.

In DSM, an application of an artefact in a context is called a *treatment*. In the sequel, we describe the technologies involved in the treatment of our problem for each architecture component.

### 5.3 End Devices

LoRaWAN encompasses various end nodes, including commercial solutions with out-of-the-box sensing capabilities. To allow full customisation and familiarise ourselves with the protocol, we chose to use Az-Delivery NodeMCU v3 (ESP8266) and Heltec WiFi LoRa 32 (ESP32) developer boards with general-purpose input/output (GPIO) pins. We wired up both with a DHT22 module to collect environment temperature (from -40 to +80 °C) and humidity (0 to 100% Relative Humidity) data. For the former, a LoRa SX1276 module, including an antenna, is wired up along with the sensor, as presented in Figure 10, while the latter already contains it onboarded. Table 1 shows the detailed pin mapping for the NodeMCU v3

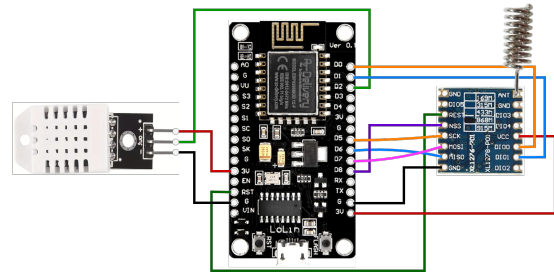


Figure 10: DHT22 and SX1276 modules mapped to NodeMCU.

board and the wired LoRa and sensor modules. The mapping is more straightforward for the WiFi Lora 32 board since it already contains the LoRa transceiver (only the DHT22 module OUT (DATA) pin is wired to the WiFi Lora 32 D2 pin).

We used Arduino IDE (Integrated Development Environment) to upload the source code<sup>6</sup> to both boards. The source code uses the DHT (Digital Temperature And Humidity) sensor library<sup>7</sup> for reading the sensor digital input, and the LMIC (LoRaWAN-MAC-in-C) library<sup>8</sup>, a LoRaWAN Class A and Class B implementation, configured for the European region (863-870 MHz). The board-specific parameters are defined at the beginning of the source code, such as the device keys to join the Network Servers and the pin mapping for the built-in LED (Light-Emitting Diode), LoRa and sensor modules.

<sup>6</sup><https://github.com/alexfabgarcia/cold-chain-tracking>

<sup>7</sup><https://github.com/Khuuxuannoc/DHT-sensor-library>

<sup>8</sup><https://github.com/mcci-catena/arduino-lmic>

Table 1: NodeMCU wired to SX1276 and DHT22 modules.

NodeMCU v3	SX1276	DHT22
D8 (15)	NSS	
D7 (13)	MOSI	
D6 (12)	MISO	
D5 (14)	SCK	
RST	RST	
D1 (5)	DIO0	
D0 (16)	DIO1	
3.3V	VCC	3.3V (+)
GND	GND	GND (-)
D2 (4)		OUT (DATA)

The `setUp` function configures the DHT22 and built-in LED pins, followed by the LMIC initialisation and the first LoRa package sending job, which automatically triggers the join request. The `loop` function only calls the `os_runloop_once` LMIC procedure, which is crucial in the event-driven system that handles LoRaWAN communication. It ensures that operations are handled promptly without blocking the main program flow, such as joining the network, sending data, and receiving acknowledgements.

The constant `TX_INTERVAL` defines the transmission interval, which might become longer due to duty cycle limitations. The `readTemperatureAndHumidity` function reads temperature and humidity through the sensor in the LMIC-encoded format, and then LMIC prepares the upstream data for the next possible transmission time. This invocation occurs whenever the transmission concluded event (`EV_TXCOMPLETE`) is received, and when there is downstream payload, the function `handleConditionViolation` is called to handle the message coming from the Cold Chain Tracking application. The implementation turns on the built-in LED to indicate that a violation has happened.

## 5.4 Network Servers

KPN Things allows us to provision a set of LoRaWAN nodes for a registered application, including a programmable LoRa device suitable for our purposes, showing the OTAA credentials during registration. The messages that devices send are encoded in the Sensor Measurement List (SenML) format (Jennings et al., 2018) and can be forwarded to multiple destination types, such as custom MQTT brokers or HTTPS endpoints. We used the EMQ free public MQTT broker (EMQ Technologies Inc, 2024) because the network server does not provide a default

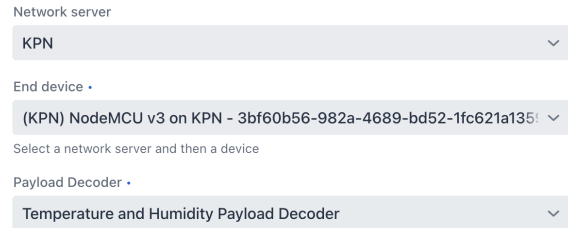


Figure 11: Payload decoder definition for the KPN device.

broker. Finally, KPN Things provides a REST (Representational State Transfer) API that enables external applications to query for devices and send downlink payload to them (KPN, 2024a).

The TTN configuration is similar to the KPN Things configuration. After setting up the application for the desired region (Europe), we registered the end devices and used the OTAA credentials generated for them. TTN provides an MQTT Broker version 3.1.1 (QoS 0 only) and a REST API (The Things Industries, 2024) for seamless integration. Additionally, TTN allows custom LoRaWAN gateways to be registered. Afterwards, TTN Mapper can be used to show their coverage in a graphic similar to Figure 6a.

## 5.5 Cold Chain Tracking application

The CCT web application has been implemented as a Java Spring Boot 3 microservice. Using the Spring Integration framework, it receives data from the MQTT brokers by subscribing message consumers that handle the Network Server’s technical details. Vaadin is the platform we used to quickly generate the web interface, including Pro-version components such as Map and CRUD to expedite the development. The Spring Security framework ensures authentication and RBAC authorisation in the web application. A PostgreSQL database stores all the application data, including the device measurements received. Google Maps API (Google Maps Platform, 2024) is used to get the geolocation from the addresses searched.

We leverage the Spring Framework dependency injection mechanism to support multi-network servers with implementations of interfaces such as `NetworkProviderService`. An Administrator user can list the devices from both providers and assign to each of them an available payload decoder, defined by the interface `DevicePayloadDecoder` for extensibility, as displayed in Figure 11. The “Temperature and Humidity Payload Decoder” is the default implementation and decodes the LMIC payload into a key-pair Java Map interface implementation.

Figure 12 illustrates the setup of the Novavax COVID-19 Vaccine product on CTT, along with the desirable conditions for its safety. The administra-

tor previously configured the measurement types with their unit (°C or °F). This feature can generalise the design for use in cases other than cold chain since the user can configure multiple measurement types and assign them to product types accordingly.

Name •  
Novavax COVID-19

Category •  
VACCINE

Measurement Type  
Temperature X

Measurement Type	Minimum	Maximum
Temperature (°C)	2.0	8.0

Figure 12: Novavax vaccine restrictions managed on CCT.

After defining all basic entities, the CCT administrator creates a freight by defining the product, the end device, the carrier, a description, the origin and the destination. Figure 13 shows a freight configuration for a Novavax COVID-19 vaccine freight between two vaccination centres.

Product •  
Novavax COVID-19

Device •  
NodeMCU v3 on KPN

Carrier •  
Alex Garcia

Description  
Batch 12345

Origin •  
Burgemeester de Bordesstraat 80, 1404

Destination •  
Albert Schweitzerlaan 31, 7334 DZ Apeldoorn

Figure 13: Vaccine freight between two vaccination centres.

The Carrier users can log in to CCT through the end device and start a freight assigned to them. The application MQTT subscribers constantly listen to and handle the device’s events with the associated freights. Apart from storing the measurements and location, a downlink message is sent to the end devices via the Network Server’s APIs when a violation of the product conditions occurs.

## 6 TESTS AND VALIDATION

The treatment prototypes were first placed in stationary indoor and outdoor positions to verify the activation of both end devices. Listing 1 shows a sample of the device serial log while handling an EV\_JOINED containing the network session and application ses-

sion keys from the OTAA method. Listing 2 shows the log of a device receiving a downlink violation message and turning its LED on.

```
730507: EV_JOINED
devAddr: 15AE(...)
AppSKey: 67-F4-EC-(...)
NwkSKey: AC-AC-6A-(...)
```

Listing 1: OTAA joining serial monitor output sample.

```
819200: EV_TXCOMPLETE (inc. RX waiting)
Condition violated. Turning LED on...
```

Listing 2: Condition violated serial monitor output sample.

We performed JUnit<sup>9</sup> automated tests to assess the integration and capabilities developed in the application layer. With a stable treatment, we carried out a Single-Case Mechanism and a Scale-Up Experiment for the DSM validation phase.

### 6.1 Single-Case Mechanism

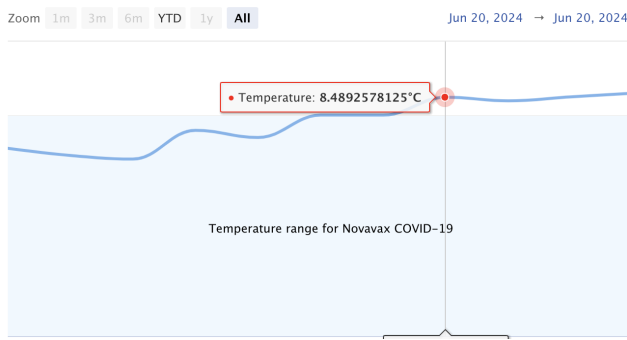
We performed a single-case mechanism validation with several routes and different transport modes to assess the effects caused by the artefact in the context. Figure 14 illustrates the CCT measurements for a freight simulated between Bussum and Soest (two cities in the Netherlands). The temperature metrics shown in Figure 14a match our intentional condition violation, which flagged the freight as violated in CCT and turned the device’s LED on. For this same route, Figure 14b shows the approximate device location received from KPN (KPN, 2024c). During our experiments, 90% of the location data received deviate less than 100 meters from the route effectively taken, which is sufficient for applications that do not require high accuracy, so that there is no need to acquire GPS modules.

Figure 15 depicts the transmissions of the NodeMCU v3 connected to the TTN during a sequential route between Soest and Bussum. The blue line indicates the effective route taken, and using the TTN Mapper app connected to the TTN’s MQTT broker, we could collect all the metadata and plot the gateways and points for each transmission on the map. The Hilversum Media Park tower gateway received multiple packages, including one from Amersfoort from a distance of 15,6 km, which evidences the impact of the LoS.

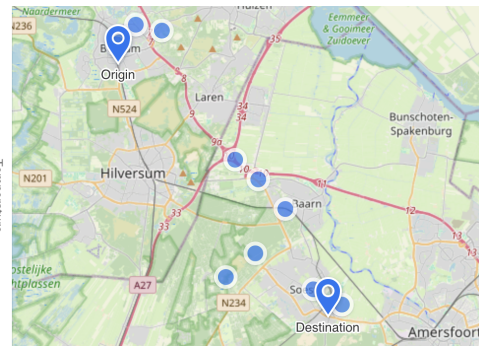
### 6.2 Scale-Up Experiment

We employed load tests to validate the robustness of our artefact. Being a community network, TTN can-

<sup>9</sup><https://junit.org>



(a) Intentional temperature violation.



(b) Approximated location via multilateration.

Figure 14: CCT displaying measurements for freight between Bussum and Soest with WiFi LoRa v3 using KPN.

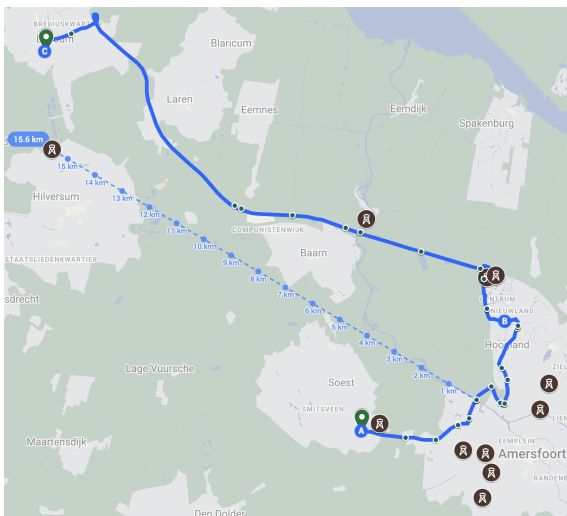


Figure 15: Gateways and longest transmission path mapped using TTN in a route from Soest to Bussum.

not be used for stress tests. At the same time, the KPN Freemium plan only supports up to three devices, even though the Standard enterprise plan supports over one thousand devices (KPN, 2024b). For these reasons, in our performance validation we focused on the LoRaWAN application in response to the stimulus simulated on the MQTT broker.

In an Apache JMeter<sup>10</sup> test plan, we defined 1000 devices for each network server, with 30 minutes of duration and a ramp-up period of 60 seconds, respecting the rate limit imposed by the EMQ public broker. Each device sends approximately one message per minute to the device-specific MQTT topic. The payload sent is a fixed measurement containing the temperature and humidity data, with a 2% chance of being violated temperature conditions, randomly above 8°C.

Besides storing the data received from the end nodes, CCT exposes metrics using the OpenTeleme-

<sup>10</sup><https://jmeter.apache.org>

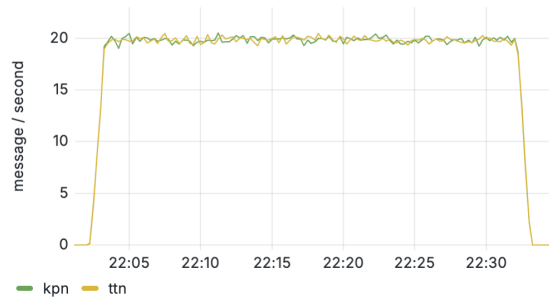


Figure 16: Device measurements throughput per Network Server.

try<sup>11</sup> standard, which helps validate the CCT performance. We have used Prometheus<sup>12</sup> and Grafana<sup>13</sup>, which are powerful open-source tools, to collect and visualise the metrics, respectively.

Figure 16 shows a Grafana panel for the throughput of messages handled per second for each network service in the CCT application. The data plotted is compatible with the throughput rate stated in the JMETER test results, namely, 19.8 messages per second. A single application instance achieved this throughput, processing the measurement message in less than ten milliseconds on average.

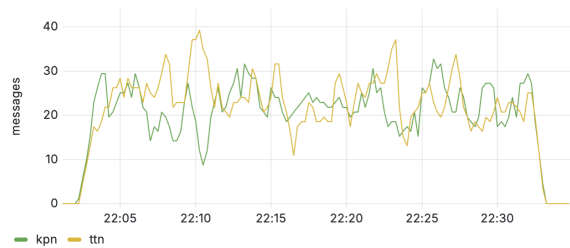


Figure 17: Condition violated metric per Network Server.

Figure 17 displays the violated messages metric

<sup>11</sup><https://opentelemetry.io>

<sup>12</sup><https://prometheus.io>

<sup>13</sup><https://grafana.com>

exposed by the CCT application per Network Server. Considering that the application processed around 2400 messages per minute, it is possible to correlate the error rate proposed in the test plan with the 25 violations identified per minute.

## 7 FINAL REMARKS

This paper reported on the design, implementation, and in-field tests of Cold Chain Tracking, a multi-network server LoRaWAN application designed to support cold chain logistics. Our study shows that multiple network servers can be used in combination with minimum parameter configuration in the end device due to the LoRaWAN OTAA mechanism and protocol interoperability. In addition, a decoupled and scalable microservice abstracts the source of the device information through multiple MQTT subscribers. This setup offers insights for PPP in various domains where long-range low-power communication excels, allowing new IoT applications to be added by leveraging the same infrastructure and maximising RoI.

The paper also showed how simulation tools can offer a proper balance between antennas' coverage and the LoRa modulation transmissions received by the Network Servers, which can be employed to private gateways deployment as an alternative to ad-hoc attempts. On top of that, the exposed geolocation by multilateration feature is an affordable GPS-free alternative for setups in case a highly accurate location is not required, which means investing more in gateway deployment can pay off compared to high-power end device costs with GPS modules.

Following best object-oriented development practices, the application is suitable and extensible for different network providers, measurement types, and payload decoders to work with the end device's sensing capabilities. This makes this approach in principle appropriate to build systems for other real-world scenarios than cold chain, especially in challenging remote areas.

During the development of our application, we encountered several challenges. Although the LMAC library implements and abstracts most of the particularities of the LoRaWAN protocol for end nodes, finding the correct pin mapping configuration for both boards demanded considerable investigation. Additionally, after wiring the LoRa SX1276 to the NodeMCU v3, not many general-purpose IO pins were left for the sensor itself, and proper mapping also took time. The TTN network does not provide the geolocation by multilateration feature out-of-the-box since it depends

on the gateways deployed by the community users. We circumvented this limitation during the test phase by using the TTN Mapper app and a mobile device connected to the Internet. For production deployments, modern LoRa transceivers, proper gateways, and network server configuration must be in place. Further, the TTN MQTT broker version 3.1.1 only supports QoS 0 level, which implies that messages are not guaranteed to be delivered, and the shared subscriptions capability is not supported. This MQTT 5 feature ensures that the broker pushes a message to only one client in a group of subscribers, improving load balancing and fault tolerance while consuming the messages from one topic by the CCT microservice. The TTN webhook integration could be used to forward messages to an MQTT 5 server instead.

We conducted field tests in the Netherlands, where the topography and infrastructure favour the LoRaWAN protocol. Future work could exploit other conditions by applying Technical Action Research (Wieringa, 2014), for example, by a) mapping antennas' coverage using tools such as Roger Coudé's Radio Mobile Online (Roger Coudé VE2DBE, 2024) for remote areas like the Amazon forest for antivenom distribution scenario and b) deploying gateways accordingly in a private network to validate the artefact in more adverse conditions.

Regarding the features implemented in our application, some useful improvements could be to a) automate device registration in the CCT application by employing technologies such as NFC (Near Field Communication); b) send freight thresholds to devices right after they joined the server; c) use the device local storage to handle thresholds when communication is not possible; and d) predict when the conditions will reach thresholds to prevent violations.

## ACKNOWLEDGEMENTS

This study was partly financed by the "Coordenação de Aperfeiçoamento de Pessoal de Nível Superior" - Brazil (CAPES) - Finance Code 001. We also thank the Brazilian National Council of Technological and Scientific Development (CNPq) and the São Paulo Research Foundation (FAPESP) for sponsoring our research in the context of the Brazilian National Institute of Science and Technology in Medicine Assisted by Scientific Computing (INCT-MACC).

Regarding software licences, we acknowledge KPN for the KPN Things freemium project, The Things Industries for The Things Stack Sandbox, and Vaadin for the Pro license. We also appreciated the support from the TTN community in the forum.

## REFERENCES

- Ashton, K. et al. (2009). That ‘internet of things’ thing. *RFID journal*, 22(7):97–114.
- Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things: A survey. *Computer networks*, 54(15):2787–2805.
- Augustin, A., Yi, J., Clausen, T., and Townsley, W. M. (2016). A study of LoRa: Long range & low power networks for the Internet of Things. *Sensors*, 16(9):1466.
- Aung, M. M. and Chang, Y. S. (2014). Temperature management for the quality assurance of a perishable food supply chain. *Food Control*, 40:198–207.
- Badia-Melis, R., Mc Carthy, U., Ruiz-Garcia, L., Garcia-Hierro, J., and Villalba, J. R. (2018). New trends in cold chain monitoring applications - A review. *Food Control*, 86:170–182.
- Bose, A., Aithal, A. K., and Rajeshwari, B. (2022). Vaccine cold storage monitoring and tracking using LoRaWAN. In *2022 2nd International Conference on Intelligent Technologies (CONIT)*, pages 1–6. IEEE.
- Cattani, M., Boano, C. A., and Römer, K. (2017). An experimental evaluation of the reliability of LoRa long-range low-power wireless communication. *Journal of Sensor and Actuator Networks*, 6(2):7.
- Cavalcante, A. M., Marquezini, M. V., Mendes, L., and Moreno, C. S. (2021). 5G for remote areas: Challenges, opportunities and business modeling for Brazil. *IEEE Access*, 9:10829–10843.
- Centenaro, M., Vangelista, L., Zanella, A., and Zorzi, M. (2016). Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios. *IEEE Wireless Communications*, 23(5):60–67.
- EMQ Technologies Inc (2024). Free public MQTT Broker. <https://www.emqx.com/en/mqtt/public-mqtt5-broker>. Accessed: 2024-06-20.
- Enriko, I., Alemuda, F., and Adrianto, D. (2022). An Approach to Monitor Vaccine Quality During Distribution Using Internet of Things. *Journal of Information Science & Engineering*, 38(5).
- Fan, H. W. and Monteiro, W. M. (2018). History and perspectives on how to ensure antivenom accessibility in the most remote areas in Brazil. *Toxicon*, 151:15–23.
- Fargas, B. C. and Petersen, M. N. (2017). GPS-free geolocation using LoRa in low-power WANs. In *2017 global internet of things summit (Giots)*, pages 1–6. IEEE.
- Frangoudis, P. A., Tsigkanos, C., and Dustdar, S. (2021). Connectivity technology selection and deployment strategies for IoT service provision over LPWAN. *IEEE Internet Computing*, 25(1):61–70.
- Garcia, A. F. and de Souza, W. L. (2023). Internet of Things Applications for Cold Chain Vaccine Tracking: A Systematic Literature Review. In *International Conference on Information Technology-New Generations*, pages 323–330. Springer.
- Google Maps Platform (2024). Geolocation API Overview. <https://developers.google.com/maps/documentation/geolocation/overview>. Accessed: 2024-06-20.
- Jennings, C., Shelby, Z., Arkko, J., Keranen, A., and Bormann, C. (2018). Sensor measurement lists (SenML). Technical report, Internet Engineering Task Force (IETF).
- KPN (2024a). KPN Things Developer Manual - API Access. <https://docs.kpnthings.com/dm/concepts/api>. Accessed: 2024-06-20.
- KPN (2024b). KPN Things Explorer. <https://www.kpn.com/zakelijk/internet-of-things/kpn-things/explorer.htm>. Accessed: 2024-06-20.
- KPN (2024c). KPN Things LoRa Geolocation. <https://docs.kpnthings.com/lora/geolocation/intro>. Accessed: 2024-06-20.
- Labdaoui, N., Nouvel, F., and Dutertre, S. (2023). Energy-efficient IoT Communications: A Comparative Study of Long-Term Evolution for Machines (LTE-M) and Narrowband Internet of Things (NB-IoT) Technologies. In *2023 IEEE Symposium on Computers and Communications (ISCC)*, pages 823–830. IEEE Computer Society.
- Lawrence, P. W., Phippard, T. M., Ramachandran, G. S., and Hughes, D. (2018). Developing the IoT to support the health sector: A case study from Kikwit, DR Congo. In *Emerging Technologies for Developing Countries: First International EAI Conference, AFRICATEK 2017, Marrakech, Morocco, March 27-28, 2017 Proceedings 1st*, pages 45–56. Springer.
- LoRa Alliance (2024). LoRaWAN Network Coverage - LoRa Alliance. <https://loro-alliance.org/lorawan-network-coverage>. Accessed: 2024-06-20.
- Mishra, B. and Kertesz, A. (2020). The use of MQTT in M2M and IoT systems: A survey. *Ieee Access*, 8:201071–201086.
- Roger Coudé VE2DBE (2024). Radio Mobile Online. <https://www.ve2dbe.com/rmonline.s.asp>. Accessed: 2024-06-20.
- Saban, M., Aghzout, O., Medus, L. D., and Rosado, A. (2021). Experimental analysis of IoT networks based on LoRa/LoRaWAN under indoor and outdoor environments: Performance and limitations. *IFAC-PapersOnLine*, 54(4):159–164.
- The Things Industries (2024). The Things Stack HTTP (REST) API. <https://www.thethingsindustries.com/docs/api/reference/http/routes/>. Accessed: 2024-06-20.
- Vangelista, L., Zanella, A., and Zorzi, M. (2015). Long-range IoT technologies: The dawn of LoRa™. In *Future Access Enablers for Ubiquitous and Intelligent Infrastructures: First International Conference, FABULOUS 2015, Ohrid, Republic of Macedonia, September 23-25, 2015*, pages 51–58. Springer.
- Wieringa, R. J. (2014). *Design science methodology for information systems and software engineering*. Springer.
- Zinas, N., Kontogiannis, S., Kokkonis, G., Valsamidis, S., and Kazanidis, I. (2017). Proposed open source architecture for Long Range monitoring: the case study of cattle tracking at Pogoniani. In *Proceedings of the 21st Pan-Hellenic Conference on Informatics*, pages 1–6.