Universidade Federal de Pernambuco
Centro de Informática

Graduação em Engenharia da Computação

Integração do protocolo *LoRaWAN* à meta plataforma *IoT KNoT* para coleta de dados em uma planta industrial

Luis Felipe Miranda da Silva

Trabalho de Graduação

Recife
20 de março de 2024

Universidade Federal de Pernambuco
Centro de Informática

Luis Felipe Miranda da Silva

Integração do protocolo *LoRaWAN* à meta plataforma *IoT KNoT* para coleta de dados em uma planta industrial

Trabalho apresentado ao Programa de Graduação em Engenharia da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Engenharia da Computação.

Orientador: *Prof. Abel Guilhermino da Silva Filho*

Recife
20 de março de 2024

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Silva, Luis Felipe Miranda da.
Integração do protocolo LoRaWAN à meta plataforma IoT KNoT para coleta de dados em uma planta industrial / Luis Felipe Miranda da Silva. - Recife, 2024.
48 : il., tab.

Orientador(a): Abel Guilhermino da Silva Filho
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Informática, Engenharia da Computação - Bacharelado, 2024.

1. Integração IoT. 2. KNoT IoT. 3. ChirpStack. 4. LoRaWAN. 5. Gerenciamento de Dispositivos. I. Silva Filho, Abel Guilhermino da. (Orientação). II. Título.

000 CDD (22.ed.)

LUIS FELIPE MIRANDA DA SILVA

Integração do protocolo LoRaWAN à meta plataforma IoT KNoT para coleta de dados em uma planta industrial

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de bacharel em Engenharia da Computação.

Aprovado em: 20 / 03 / 2024

BANCA EXAMINADORA

_____

Prof. Dr. Abel Guilhermino da Silva Filho (Orientador)

Universidade Federal de Pernambuco

_____

Prof. Dr. Divanilson Rodrigo de Sousa Campelo (Examinador Interno)

Universidade Federal de Pernambuco

Resumo

A Internet das Coisas visa conectar em larga escala objetos cotidianos sem capacidades de computação, de acordo com o *IEEE WORLD FORUM INTERNET OF THINGS* de 2014, era previsto mais de 50 bilhões de dispositivos conectados até 2020. Essa visão dependia do desenvolvimento de um ecossistema sustentável de empresas oferecendo uma variedade de aplicativos, produtos e serviços. O núcleo desse ecossistema seria uma plataforma de software compartilhada por várias aplicações e produtos. E a integração eficiente de dispositivos *IoT* (Internet das Coisas) em ambientes industriais e urbanos representa um desafio constante, exigindo soluções robustas para a coleta e gerenciamento de dados. Este trabalho aborda a colaboração entre duas plataformas proeminentes, *KNoT IoT* e *ChirpStack*, visando otimizar a comunicação e eficiência na coleta de dados em larga escala. Enquanto a *KNoT IoT* oferece uma abordagem abrangente para a gestão de dispositivos *IoT*, a *ChirpStack* destaca-se na manipulação eficiente de dispositivos *LoRaWAN (Long Range Wide Area Network)*. A sinergia entre essas plataformas promete fortalecer a infraestrutura para a gestão efetiva de dispositivos, desbloqueando possibilidades significativas para aplicações industriais e urbanas.

A metodologia abrangeu diversas fases, iniciando com a análise da eficiência conjunta da KNoT IoT e ChirpStack na gestão de dispositivos e protocolos diversos. A estratégia para desenvolver a integração das plataformas foi planejada utilizando *Golang* e *AMQP* a partir da máquina de estados do *KNoT Virtual Thing*, implementada em uma das integrações da *ChirpStack*.

Embora desafios tenham sido identificados, como na configuração para lidar com 1000 dispositivos no teste de estresse, o trabalho conclui que a integração pode ser aplicada efetivamente em plantas industriais com até 600 sensores. A análise metodológica destaca a necessidade de ajustes específicos para cenários de maior porte, ressaltando a importância de futuros desenvolvimentos, como a migração para a *ChirpStack* V4 e a simplificação da configuração para usuários finais.

Palavras-chave: Integração IoT, KNoT IoT, ChirpStack, LoRaWAN, Gerenciamento de Dispositivos, Comunicação Eficiente, Coleta de Dados, Ambientes Industriais, Ambientes Urbanos.

Abstract

The Internet of Things aims to connect everyday objects on a large scale without computing capabilities, according to the IEEE WORLD FORUM INTERNET OF THINGS in 2014. It was predicted that more than 50 billion devices would be connected by 2020. This vision relied on the development of a sustainable ecosystem of companies offering a variety of applications, products, and services. The core of this ecosystem would be a software platform shared by various applications and products. The efficient integration of Internet of Things (IoT) devices in industrial and urban environments represents a constant challenge, requiring robust solutions for data collection and management. This work addresses the collaboration between two prominent platforms, KNoT IoT and ChirpStack, aiming to optimize communication and efficiency in large-scale data collection. While KNoT IoT offers a comprehensive approach to IoT device management, ChirpStack excels in the efficient handling of LoRaWAN (Long Range Wide Area Network) devices. The synergy between these platforms promises to strengthen the infrastructure for effective device management, unlocking significant possibilities for industrial and urban applications.

The methodology covered various phases, starting with the analysis of the joint efficiency of KNoT IoT and ChirpStack in managing diverse devices and protocols. The strategy for developing the integration of the platforms was planned using Golang and AMQP from the state machine of the KNoT Virtual Thing, implemented in one of the ChirpStack integrations.

Although challenges were identified, such as configuration issues when dealing with 1000 devices in stress testing, the work concludes that the integration can be effectively applied in industrial plants with up to 600 sensors. The methodological analysis emphasizes the need for specific adjustments for larger scenarios, highlighting the importance of future developments, such as migrating to ChirpStack V4 and simplifying configuration for end-users.

Keywords: IoT Integration, KNoT IoT, ChirpStack, LoRaWAN, Device Management, Efficient Communication, Data Collection, Industrial Environments, Urban Environments.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

IoT Internet of Things

KNoT KNoT Network of Things

AMQP Advanced Message Queuing Protocol

MQTT Message Queuing Telemetry Transport

Wi-Fi Wireless Fidelity

LoRa Long Range

LoRaWAN Long Range Wide Area Network

MAC Media Access Control

ADR Adaptive Data Rate

AppSKey Application Session Key

NwkSKey Network Session Key

OTAA Over-the-Air Activation

ABP Activation by Personalization

gRPC google Remote Procedure Call

UDP User Datagram Protocol

ACK Acknowledgement

API Application Programming Interface

HTTP Hypertext Transfer Protocol

Go Go Programming Language

RF Radio Frequency

TCP-IP Transmission Control Protocol/Internet Protocol

ModBus Modular Digital Communication Protocol

REST Representational State Transfer

CESAR Centro de Estudos e Sistemas Avançados do Recife

SF Spread Factory

1 Introdução

A Internet das Coisas *(Internet of Things - IoT)* pode ser compreendida como uma rede ubíqua de coleta e processamento de dados de um ambiente local ou global a partir de sensores conectados à rede (Albertin and Albertin, 2017). Esses sensores são equipamentos eletrônicos que possuem a capacidade de interagir com alguma característica mensurável do meio analisado e, a partir de medições, conseguem traduzir essa leitura para a forma digital e enviar por meio de uma rede pública ou privada para uma central de análise, onde a informação pode ser compartilhada, e ações podem ou não ser tomadas (Patel et al., 2016).

Utilizando o IoT, a revolução industrial 4.0 procede uma sequência de outras revoluções que se deram pela forma de manipular uma fonte de energia ou conhecimento, sendo a primeira com o uso da energia do vapor, seguida pela manipulação da eletricidade para mover motores; para então se utilizar da eletrônica e tecnologia da informática para controle da produção na terceira revolução industrial. Agora, com a tecnologia de protocolos avançada e com a maior disponibilidade de comunicação via rádio com o avanço das comunicações móveis, se fez possível a presença de pequenos sensores espalhados por toda uma planta industrial para a coleta de dados, iniciando a era das indústrias inteligentes baseadas nas *Smart Enterprise* com a coleta e processamento de dados utilizando o IoT (Okano, 2017). A revolução industrial 4.0 já está acontecendo, e milhares de dados são coletados a cada segundo, ajudando a aumentar a eficiência de produção das plantas industriais ao redor do mundo ((Manavalan and Jayakrishna, 2019)). As redes IoT chegaram para ajudar nessa coleta e tornar mais eficiente a comunicação entre os equipamentos da planta industrial (Manavalan and Jayakrishna, 2019). Mas a diversidade de protocolos IoT é vasta e com diversas formas de se comunicar.

Uma das maiores diferenças entre um protocolo de internet comum e da internet das coisas é a preocupação com o consumo de energia dos equipamentos. Com a indisponibilidade de manutenção constante de milhares de sensores espalhados em um ambiente, é preciso que sua bateria tenha um baixo consumo, para poder durar muito tempo. E com um desses objetivos nasce o protocolo *LoRaWAN*, que é LPWAN (*Low-power Wide-area Network*), pensando para longas distâncias e baixo consumo de energia (Augustin et al., 2016), um ótimo custo-benefício econômico e ambiental pela longa duração das baterias para a coleta de dados (Sherazi et al., 2021) na área industrial e vem liderando o avanço para a indústria 5.0 (HINZ, 2023) ao redor do mundo. *LoRaWAN* é baseada em coletas pontuais de informações por sensores que passam a maior parte do tempo em *Stand-by* e acordam apenas para enviar a um aparelho central que está sempre ouvindo, chamado de

Gateway (HINZ, 2023). Por conta da sua modulação em forma de *chirp* (HINZ, 2023), aliada a um sistema adaptativo de fator de espalhamento, acaba conseguindo um grande alcance em áreas urbanas (Augustin et al., 2016).

Por conta da diversidade de protocolos voltados para o IoT, a necessidade de um ponto focal se fez necessário para a junção e compreensão de diferentes formas de receber dados dos mais diversos sensores. Para esse "nó"central, foi desenvolvido o *KNoT* pelo Centro de Estudos e Sistemas Avançados do Recife (C.E.S.A.R) (kno, 2019), com a ideia de ser uma "cola"para juntar todos os dados provenientes de diferentes protocolos *IoT*. O *KNoT* foi inicialmente projetado para incorporar protocolos de *IoT* para aplicações residenciais utilizando rádio frequência de curto alcance. Mas contendo em seu planejamento a incorporação de protocolos industriais para suprir as necessidades do mercado, como o *LoRaWAN* (de Souza, 2017).

Este projeto tem como intuito mostrar o estudo e o desenvolvimento da integração do protocolo *LoRaWAN* ao *KNoT*. Adicionalmente, este trabalho de conclusão visa o estudo do desempenho do *KNoT* em um cenário industrial recebendo os dados via uma nova integração. Por fim, obtendo um comparativo entre os dados coletados entre o *LoRaWAN* e outros protocolos já utilizados pelo *KNoT*.

2 Referencial Teórico

2.1 As Coisas

Coisas são objetos físicos que não têm como primeiro objetivo servir como ponto de comunicação em uma rede ou executar cálculos como um computador. As coisas são objetos comuns do dia a dia, como carros, TVs, máquinas de lavar ou até mesmo grandes equipamentos de geração de vapor, como caldeiras em uma fábrica (Patel et al., 2016). Com o avanço das comunicações móveis e a redução do tamanho dos equipamentos de transmissão, como antenas impressas nas placas com alto poder de transmissão de dados, permitiu a implementação de menores equipamentos de comunicação (Wang and Du, 2014). Isso permitiu que coisas do dia a dia pudessem ter em seu corpo instaladas algumas antenas para comunicar seus dados com alguma central.

2.2 A Internet das Coisas - *IoT*

A internet convencional é formada por diversos computadores que conversam entre si por meio de vários protocolos, compartilhando informações ou recursos com diversas finalidades (Ross, 2008). Diferentemente disso, a Internet das Coisas é composta por objetos do dia a dia que, por meio de sensores instalados e pela leitura de alguma característica física, traduzem a informação e a enviam para a rede utilizando algum protocolo de comunicação. Nessa rede, uma central irá receber os dados e, por meio de uma aplicação, eles serão analisados, podendo ou não resultar em ações (Patel et al., 2016).

Um exemplo da utilização da rede *IoT* pode ser observado em Minas Gerais, onde sensores foram instalados em diversas partes dos caminhões para o monitoramento da saúde do veículo, com os pneus, motor e cargas monitorados a todo momento, além de GPS de alta precisão para verificar a disponibilidade da frota e traçar a melhor rota ao destino (Albertin and Albertin, 2017). Podendo monitorar os equipamentos e uso dos carros, calculando custos e ganhos, os sensores fornecem informações valiosas que tornam a mina mais competitiva no mercado (Albertin and Albertin, 2017).

2.3 *LoRa*

Com a necessidade de novos paradigmas de comunicação, fez-se necessário também a evolução dos meios de comunicação. A comunicação móvel é um grande exemplo, com a constante evolução dos seus protocolos como o 3G, 4G e atualmente o 5G. Mas diferente da comunicação móvel comum, a Internet das Coisas não necessita de grandes volumes de dados constantes sendo transmitidos; os dados necessários são pontuais e com menor frequência de transmissão devido ao consumo da bateria. Os sensores instalados nos equipamentos precisam ter um tamanho pequeno e não perceptível para que possam apenas fazer as suas leituras sem interferir no funcionamento da coisa em si (Devalal and Karthikeyan, 2018).

A tecnologia *LoRa* como ilustrada na Figura 2.1, parte integrante da evolução avançada das comunicações sem fio na Internet das Coisas (*IoT*), destaca-se como uma *Low Power Wide Area Network* ou LPWAN chave. Desenvolvida pela *Semtech* em 2012, o *LoRa* é uma tecnologia proprietária que envolve duas camadas distintas: uma camada física utilizando a técnica de modulação *Chirp Spread Spectrum* ou CSS e um protocolo de camada MAC (*Media Access Control*) conhecido como *LoRaWAN*. A camada física *LoRa*, muitas vezes referida como *LoRa PHY*, descreve a técnica de modulação e seus diferentes parâmetros adotados para a comunicação *LoRa*, que incluem largura de banda BW, fatores de espalhamento SF e taxa de código CR. (Raychowdhury and Pramanik, 2020; Alliance, 2020). Já a camada MAC (*Media Access Control*) do *LoRaWAN* desempenha um papel crucial na gestão do acesso ao meio para dispositivos IoT que utilizam a tecnologia *LoRa*. Ela define as regras para a transmissão de dados e a comunicação eficiente entre os dispositivos e a rede. A camada MAC *LoRaWAN* é responsável por controlar o acesso ao canal de comunicação, garantindo um uso eficiente do espectro de frequência. E as classes dos dispositivos *LoRaWAN* desempenham papéis distintos na gestão de energia e na disponibilidade de comunicação. Sendo a Classe A mais eficiente em consumo de energia enquanto a Classe C prioriza disponibilidade sobre eficiência energética.

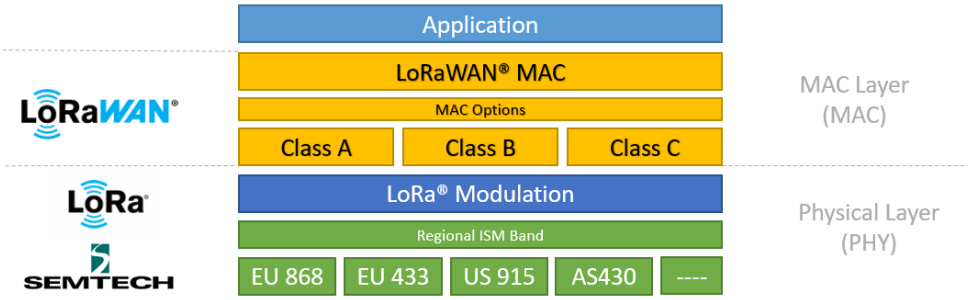

Figura 2.1 Camadas de uma "Aplicação"*LoRa* (semtech, 2019).

O protocolo *LoRa* proporciona flexibilidade, oferecendo diferentes taxas de dados e alcances, dependendo dos fatores de espalhamento múltiplo (de 7 a 12) utilizados. O *LoRaWAN*, desenvolvido pela *LoRa Alliance* em 2015, é um protocolo de camada MAC

específico para *LoRa*. A topologia de rede em estrela-estrela do *LoRa* permite comunicação entre dispositivos finais, utilizando o esquema de modulação *LoRa* (Haxhibeqiri et al., 2018; Alliance, 2020). Os diferentes fatores de espalhamento da Figura 2.1 são utilizados entre 7 e 12 para reduzir interferências e enfraquecimento por multipercursos. O sinal na comunicação *LoRa* torna-se mais longo ao usar diferentes fatores de espalhamento, e o comprimento do código de espalhamento é expresso numericamente como 2SF. Esses *"chirps"2SF* cobrem toda a faixa de frequência. Por meio dessa técnica, a taxa de modulação e a potência de transmissão podem variar entre os nós de acordo com o uso do fator de espalhamento. Um símbolo de informação é codificado por um *"chirp"* e a taxa de *"chirp"* novamente depende da largura de banda. Alterando o fator de espalhamento, diferentes parâmetros de transmissão são modificados (Raychowdhury and Pramanik, 2020; Alliance, 2020).

Spreading Factor (For UL at 125 KHz)	Bit Rate	Range (Depends on Terrain)	Time on Air for an 11-byte payload
SF10	980 bps	8 km	371 ms
SF9	1760 bps	6 km	185 ms
SF8	3125 bps	4 km	103 ms
SF7	5470 bps	2 km	61 ms

Figura 2.2 *LoRa Spreading Factors* (semtech, 2019).

O *LoRaWAN* opera em diversas regiões ao redor do mundo, adaptando-se às regulamentações de frequência locais. A Figura 2.3 mostra que na América do Norte, utiliza a faixa de 902 MHz a 928 MHz, enquanto na Europa, opera entre 863 MHz e 870 MHz. Na Ásia, é comum a faixa de 433 MHz, e na Austrália, a faixa varia de 915 MHz a 928 MHz, que é a utilizada aqui no Brasil. Essa diversidade de frequências permite a flexibilidade do *LoRaWAN* em atender a diferentes requisitos de comunicação para uma ampla gama de aplicações, como cidades inteligentes, monitoramento ambiental e agricultura conectada, respeitando as regulamentações específicas de cada região (Haxhibeqiri et al., 2018; Alliance, 2020; Kuan, 2019).

Lora-based Frequency MOKOSMART

Region	The Lora-based Frequency
Europe	863-870 MHz 433 MHz
US	902-928 MHz
China	470-510 MHz 779-787 MHz
Australian	915-928 MHz
Indian	865-867 MHz
Asia	433 MHz
North America	915 MHz

Figura 2.3 Frequências utilizadas (Kuan, 2019)

Para o controle de acesso ao meio na infraestrutura foi pensado o protocolo *LoRaWAN*, que é um protocolo *open-source* padronizado pela *LoRa Alliance* (HINZ, 2023) para a autenticação e controle do registro dos dispositivos na rede. A arquitetura da camada *Media Access Control (MAC) LoRaWAN* mostrada na Figura 2.4 facilita a comunicação entre dispositivos e *gateways* em uma arquitetura de rede em estrela, onde dispositivos finais se comunicam apenas com *gateways*, não entre si. Os *gateways*, conectados a um servidor de rede, encaminham pacotes brutos dos dispositivos finais, e o servidor de rede é responsável por enviar pacotes de descida e comandos *MAC*, encerrando a comunicação nos servidores de aplicativos, podendo ser de terceiros (Haxhibeqiri et al., 2018; Alliance, 2020).

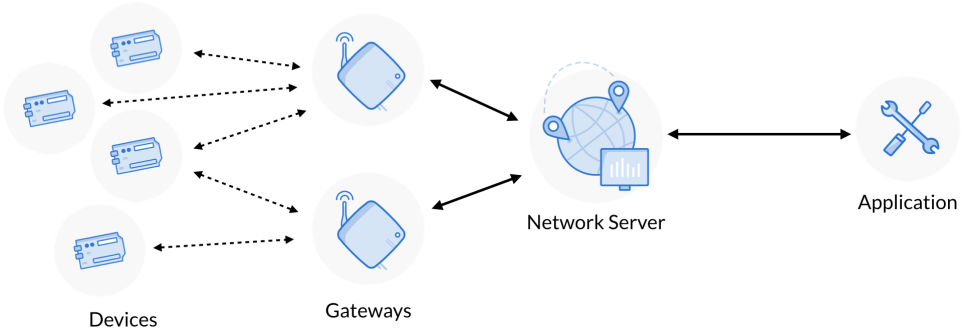

Figura 2.4 Topologia de rede *LoRaWAN* (thethingsindustries, 2020).

2.3.1 *Devices*

O padrão *LoRaWAN* estabelece três classes de dispositivos finais, A, B e C. Os dispositivos Classe A implementam conjuntos básicos de opções para ingressar em uma rede *LoRaWAN*. Para permitir a comunicação bidirecional, cada transmissão de subida (*uplink*) de um dispositivo Classe A é seguida por duas curtas janelas de recebimento de descida (*downlink*). A comunicação de descida é acionada pelo dispositivo final, aguardando por uma comunicação de subida. As janelas de recebimento de descida iniciam 1 e 2 segundos após o final da transmissão de subida. Se a transmissão de descida ocorrer na primeira janela, o mesmo canal da subida será utilizado. Se a segunda janela de recebimento for usada, um SF e canal fixos serão aplicados, normalmente o canal de 125 kHz centrado em 869.525 MHz usando SF12, com um ciclo de trabalho de 10% e alta potência de transmissão de 24 dBm. O agendamento preciso e controle de tempo são responsabilidades do servidor de rede. Dispositivos Classe A consomem menos energia, pois estão inativos a maior parte do tempo. Dispositivos Classe B consomem mais energia devido às janelas adicionais. Dispositivos Classe C mantêm janelas de recebimento contínuas, estando praticamente sempre disponíveis para tráfego de descida, exceto durante a transmissão. O protocolo *LoRaWAN* inclui mecanismos para garantir comunicação confiável e segura (Haxhibeqiri et al., 2018; Alliance, 2020).

O *device LoRaWAN* possui um mecanismo de Taxa de Dados Adaptativa (*Adaptive Data Rate (ADR)*) que dinamicamente administra os parâmetros de *link* para melhorar a entrega de pacotes. Esse mecanismo gerencia a taxa de dados e a potência de transmissão dos dispositivos finais. Os dispositivos podem escolher permitir que o servidor de rede gerencie seus parâmetros de transmissão, indicando o bit de *ADR* de subida em seus pacotes de comunicação de subida. Alternativamente, os dispositivos têm a opção de gerenciar seus próprios parâmetros de transmissão usando o mecanismo de *ADR* interno. Essas duas partes do *ADR* operam de maneira assíncrona entre o servidor de rede e o nó final (Haxhibeqiri et al., 2018; Alliance, 2020).

2.3.2 *Gateways*

Os *gateways LoRaWAN* são dispositivos cruciais na arquitetura da rede *LoRa*, desempenhando o papel de ponte entre os dispositivos *IoT* e a infraestrutura de rede. Responsáveis por receber dados transmitidos pelos dispositivos *LoRa*, os *gateways* encaminham essas informações para servidores na nuvem, permitindo a integração e monitoramento centralizado. Esses dispositivos são fundamentais para a expansão da cobertura e eficiência da rede *LoRaWAN*, facilitando a conectividade confiável e de longo alcance em ambientes diversos, desde áreas urbanas até locais remotos (Devalal and Karthikeyan, 2018).

2.3.3 *Network Server*

Enquanto a maioria das tecnologias adota uma única camada de segurança, a rede *LoRaWAN* diferencia-se ao incorporar duas camadas distintas: segurança de rede e

segurança de aplicativo. A segurança de rede desempenha um papel crucial na autenticação do nó na rede dentro da *Network Server*, enquanto a segurança de aplicativo visa resguardar os dados da aplicação do usuário final contra intervenções do operador de rede. Essa abordagem única da tecnologia *LoRaWAN* utiliza duas chaves específicas, conhecidas como *Network Session Key (NwkSKey)* (Chave de Sessão de Rede) e *Application Session Key (AppSKey)* (Chave de Sessão de Aplicativo), para garantir tanto a segurança quanto a autenticidade do sistema (Devalal and Karthikeyan, 2018).

O LoRaWAN define o processo pelo qual um dispositivo final pode se integrar à rede. Cada dispositivo que se conecta à rede precisa ser personalizado e ativado, podendo essa ativação ocorrer pelo ar (*Over-the-Air Activation* - Over-the-Air Activation (OTAA)) ou por personalização (*Activation by Personalization* - Activation by Personalization (ABP)). No OTAA, o dispositivo é personalizado com chaves e identificadores antes de enviar solicitações de junção, que incluem *Device EUI* (DevEUI), *JoinEUI* e *DevNonce*. O servidor de rede responde com mensagens de aceitação de junção, contendo *Device Address* ou DevAddr e JoinNonce, após verificar a integridade usando *Network Key* ou *NwkKey*. Este processo é ilustrado no fluxo de pacotes, proporcionando a integração segura de dispositivos à rede *LoRaWAN* (Haxhibeqiri et al., 2018; Alliance, 2020).

2.3.4 *Application*

A camada de aplicação (*Application*) no contexto *LoRaWAN* representa a interface final entre os dispositivos *IoT* e as aplicações específicas. Essa camada define os formatos de dados, protocolos e interações entre os dispositivos e as soluções de software que processam e utilizam essas informações. Ao possibilitar a personalização das aplicações conforme as necessidades específicas de cada caso de uso, a camada de aplicação no *LoRaWAN* desempenha um papel crucial na extração de valor a partir dos dados transmitidos pela rede, abrangendo uma variedade de setores, como agricultura, saúde, logística e cidades inteligentes.

2.4 *ChirpStack v3*

Uma *LoRaWAN Network Server Stack* gerencia toda a administração da rede, controle de acessos, autenticação e direcionamentos. A *ChirpStack* é uma plataforma *open-source* com todos os componentes para a implementação de um *Network Server*, permitindo a instalação de todo o sistema em uma única máquina ou de forma separada em ambientes diversos. Usa *google Remote Procedure Call (gRPC)* para comunicação entre os componentes, possibilitando a implementação de novos módulos ou outras formas de tratamento dos dados, como mostrado na Figura 2.5 (ChirpStack, 2021).

O *ChirpStack* é uma plataforma *open-source* que desempenha um papel crucial no desenvolvimento de redes *LoRaWAN*, facilitando a configuração e ativação de cada dispositivo. Este processo pode ocorrer através do *OTAA* ou *ABP*. Além disso, o destaque da *ChirpStack* não se limita apenas à integração de dispositivos, mas também à intera-

ção eficaz com *gateways*. Oferece uma ampla gama de possibilidades para aplicações, permitindo a construção de soluções personalizadas e inovadoras. Sua capacidade de integração com protocolos como *Advanced Message Queuing Protocol (AMQP)* e *Message Queuing Telemetry Transport (MQTT)* amplia ainda mais as opções, proporcionando uma flexibilidade significativa para o desenvolvimento e implementação de soluções baseadas em *LoRaWAN*. O *ChirpStack* se destaca como uma plataforma abrangente, catalisadora para o avanço e aprimoramento de redes *LoRaWAN* (ChirpStack, 2021).

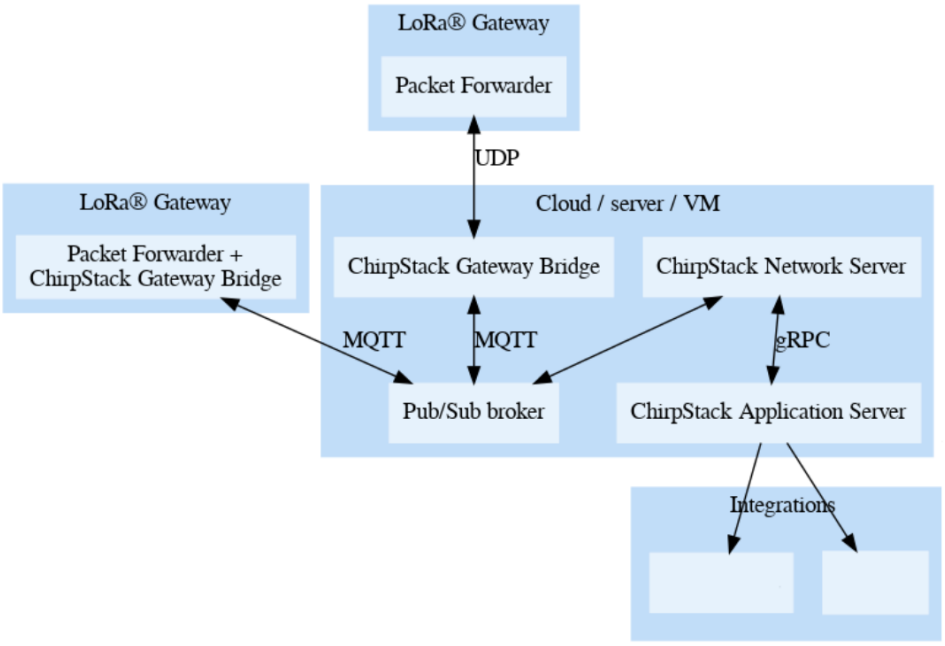

Figura 2.5 Estrutura *ChirpStack* (ChirpStack, 2021).

2.4.1 *ChirpStack Gateway Bridge*

O *ChirpStack Gateway Bridge* é o módulo da *ChirpStack* responsável por "traduzir"os dados que chegam do *gateway* para que a plataforma possa compreender (ChirpStack, 2021). Suporte aos mais populares programas de *package forwarding* no mercado como:

- *Semtech UDP* v1 e v2;
- *Basic Station*;
- *Chirpstack concentrator*:
  - Usa ZeroMQ, uma ferramenta para distribuir de forma otimizada as mensagens para os programas de consumo (ChirpStack, 2021).

Podendo ser utilizado em vários locais, o *ChirpStack Gateway Bridge* é um serviço modular, podendo ser instalado diretamente em *gateways* comerciais, permitindo o envio dos pacotes via MQTT para a plataforma ou qualquer outro *broker*. Suporte para Azure e *Google Cloud* (ChirpStack, 2021).

2.4.2 *ChirpStack Network Server*

O serviço da *ChirpStack Network Server* faz a administração dos dispositivos, autenticação dos aparelhos e pacotes. Não tem acesso às mensagens, apenas repassando as mesmas ainda criptografadas para o *Application Server* (ChirpStack, 2021).

- Controle da pilha de *downlink e Acknowledgement (ACK)*, repasse do *uplink*;
- Usa *PostgreSQL*;
- Suporta *OTAA* e *ABP*.

2.4.3 *ChirpStack Application Server*

A *ChirpStack Application Server* é responsável por administrar as informações dos dispositivos, distribui os dados para as integrações e grava todos as mensagens de *uplink e downlink* em um banco *InfluxDB*. Contém as chaves para descriptografia das mensagens (apenas o *Application Server* obtém acesso aos dados dos sensores). Permite o controle de usuário, controle das permissões que cada usuário pode ter, possibilitando uma hierarquia administrativa. Permite o controle do *Application Server* por *gRPC* e *RESTful API* com o mesmo controle que teria na interface web. *Embedded RESTful HTTP API to gRPC proxy* (mais lento por conta da tradução das mensagens). *ChirpStack SDK* nas linguagens *Go, Python, Rust e JavaScript*. Exemplos do uso da *Application Programming Interface (API)* para controle do *Application Server* (ChirpStack, 2021).

2.4.4 Integrations

A integração *MQTT* expõe todos os eventos do dispositivo. O *uplink* contém informações da aplicação, dispositivo, metadados sobre a rede e local do *gateway*, assim como o *payload*. O *AMQP/RabbitMQ* permite a publicação de todos os eventos em uma chave de roteamento. Criando uma ou várias ligações a uma ou várias filas. Sendo possível assinar todos os dados ou apenas um subconjunto. Usando *Hypertext Transfer Protocol (HTTP)*, pode-se receber todos os eventos por um *endpoint HTTP*. A *ChirpStack* fornece exemplos de implementações de *endpoints* com *Go Programming Language (Go) e Python*. Caso esse *endpoint* esteja no mesmo *host* que o *Application Server*, o *endpoint* será http://localhost:8090. Bancos de dados como *PostgreSQL* e *InfluxDB* podem ser utilizados para armazenar os eventos (ChirpStack, 2021).

2.5 Plataforma KNoT

Para suprir as necessidades do mercado, a plataforma *IoT* precisa atender as necessidades de várias partes, como o usuário final, fabricantes, aplicações web e provedor da infraestrutura, como mostra a Figura 2.6 onde o *Device Vendor* ou Fornecedor dos dispositivos precisa de aplicações e uma infraestrutura para basear seus ecossistemas, podendo utilizar essa plataforma como meio de chegar ao usuário *(User)*.

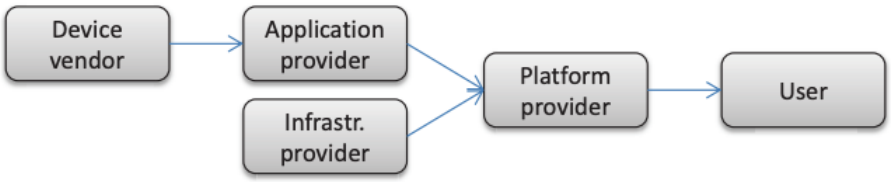

Figura 2.6 Arquitetura de uma plataforma *IoT* (Mazhelis and Tyrväinen, 2014).

Diversas propostas de *middleware* para *IoT* foram desenvolvidas para atender a requisitos específicos, especialmente na comunicação entre dispositivos heterogêneos. Embora existam plataformas recentes voltadas para ambientes de *IoT*, elas ainda não atingiram maturidade. São necessários esforços adicionais de pesquisa, incluindo a construção de infraestruturas robustas, o gerenciamento de incertezas e conflitos, e o suporte à adaptação de aplicações em ambientes dinâmicos (Pires et al., 2015).

O *KNoT* visa integrar plataformas de hardware e software para *IoT*, com ênfase em dispositivos acessíveis. Sua arquitetura, denominada *KNoT* 1.0, conecta dispositivos (*Things*) a uma nuvem por meio de um *Gateway*. Os *Things* consistem em microcontrolador, fonte de alimentação e módulo RF, permitindo a conexão de sensores e atuadores. O *Gateway* traduz os protocolos dos *Things* para *JSON/XML* e inclui módulos Radio Frequency (RF) variados. A nuvem *KNoT* coleta dados do *Gateway*, servindo como ponto de entrada para aplicações e roteando mensagens. Essa abordagem busca facilitar a criação de soluções *IoT* acessíveis e eficientes (CESAR, 2021).

A Figura 2.7 mostra a arquitetura 1.0 do *KNoT*, que pode ser dividida em três partes:

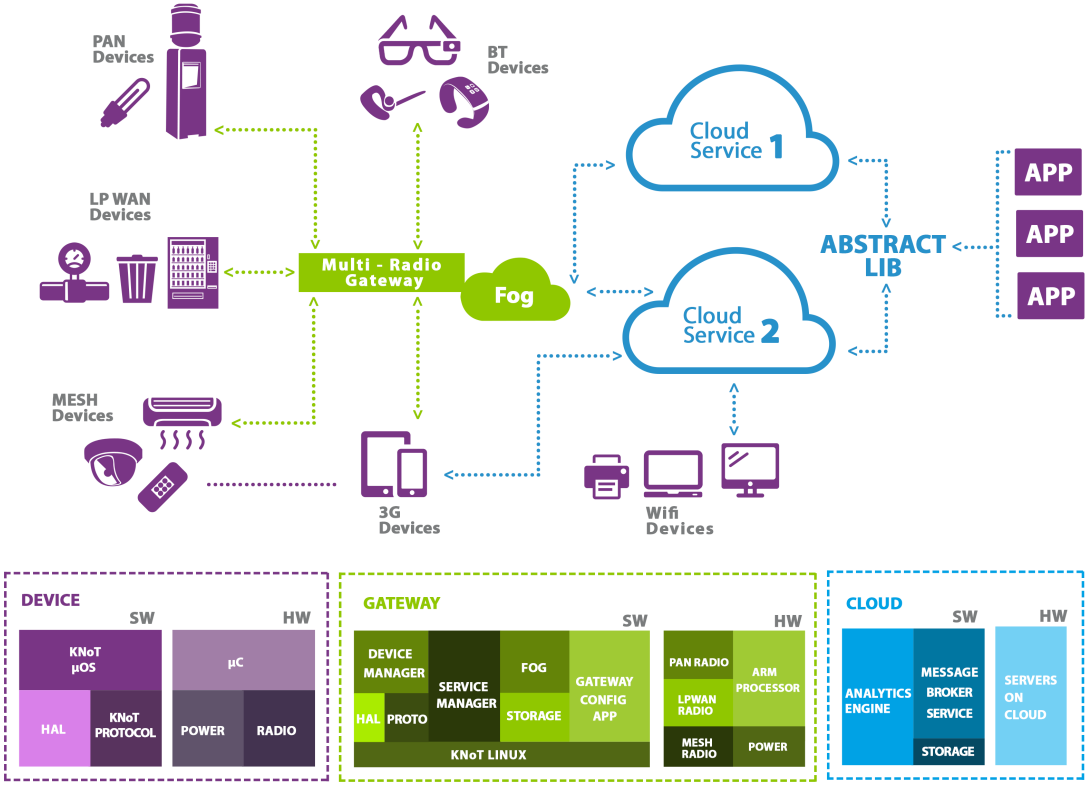

Figura 2.7 Arquitetura *KNoT* (CESAR, 2021).

- ***Device***: É a forma como o *KNoT* descreve as coisas, sendo que cada uma delas podem ter vários sensores, também levado em consideração pela meta-plataforma. O *KNoT* permite ao desenvolvedor uma série de ferramentas para implementar seu dispositivo inteligente a partir de um kit inicial contendo software e hardware (Neto et al., 2017).
- ***Gateway***: Meio do caminho, responsável pelo endereçamento dos dados na internet e envio para a *Cloud* no formato JSON ou XML. Também conta com a *Fog* que é uma versão simplificada da plataforma *KNoT Cloud* que conta com várias ferramentas, mas a mais importante sendo o sistema de *backup* responsável pela estabilidade da conexão, podendo armazenar uma boa quantidade de dados caso a conexão seja perdida por algum tempo (Neto et al., 2017).
- ***Cloud***: Dispõe de vários serviços para autenticar conexões de aplicações e dispositivos, coletar e armazenar o que é recebido em um banco de dados, gerenciar dispositivos, disponibilizar os dados via API e outras conexões como AMQP para outros serviços externos, além de uma *engine* para indexar e mostrar os dados armazenados (Neto et al., 2017).

2.5.1 *KNoT Virtual Thing*

Junto ao *KNoT*, o Centro de Estudos e Sistemas Avançados do Recife (CESAR) também desenvolveu uma ferramenta para facilitar a integração de outros protocolos à meta-plataforma, tais como *AMQP,MQTT e Modular Digital Communication Protocol (ModBus)*, entre outros. O *KNoT Virtual Thing* é instalado em um ambiente Linux e faz a coleta do dados de algum ponto do sistema desejado como a coleta de dados de um banco de dados *SQL* local, podendo ser via API Representational State Transfer (REST) ou ModBus Transmission Control Protocol/Internet Protocol (TCP-IP). Após configurado com as credenciais corretas geradas pela *Cloud*, o *KNoT Virtual Thing* utiliza uma conexão AMQP diretamente com a *Cloud* ou a Fog para enviar os dados coletados a partir do algoritmo de autenticação, que utiliza a configuração (*schema*) do *device* com seus sensores e um *token* de autenticação gerado pela *Cloud* para estabelecer uma conexão *AMQP*. Utilizando o diagrama demonstrada na Figura 2.8, configura o *device* na *Cloud* a partir do pedido de autenticação do dispositivo, caso o mesmo não esteja registrado um pedido de registro é feito e então é autenticado, depois sua configuração é enviada para a *KNoT Cloud*, com todos os dados configurados de forma correta, o dispositivo pode enviar os dados coletados pelo sensor para a meta-plataforma (CESAR, 2024).

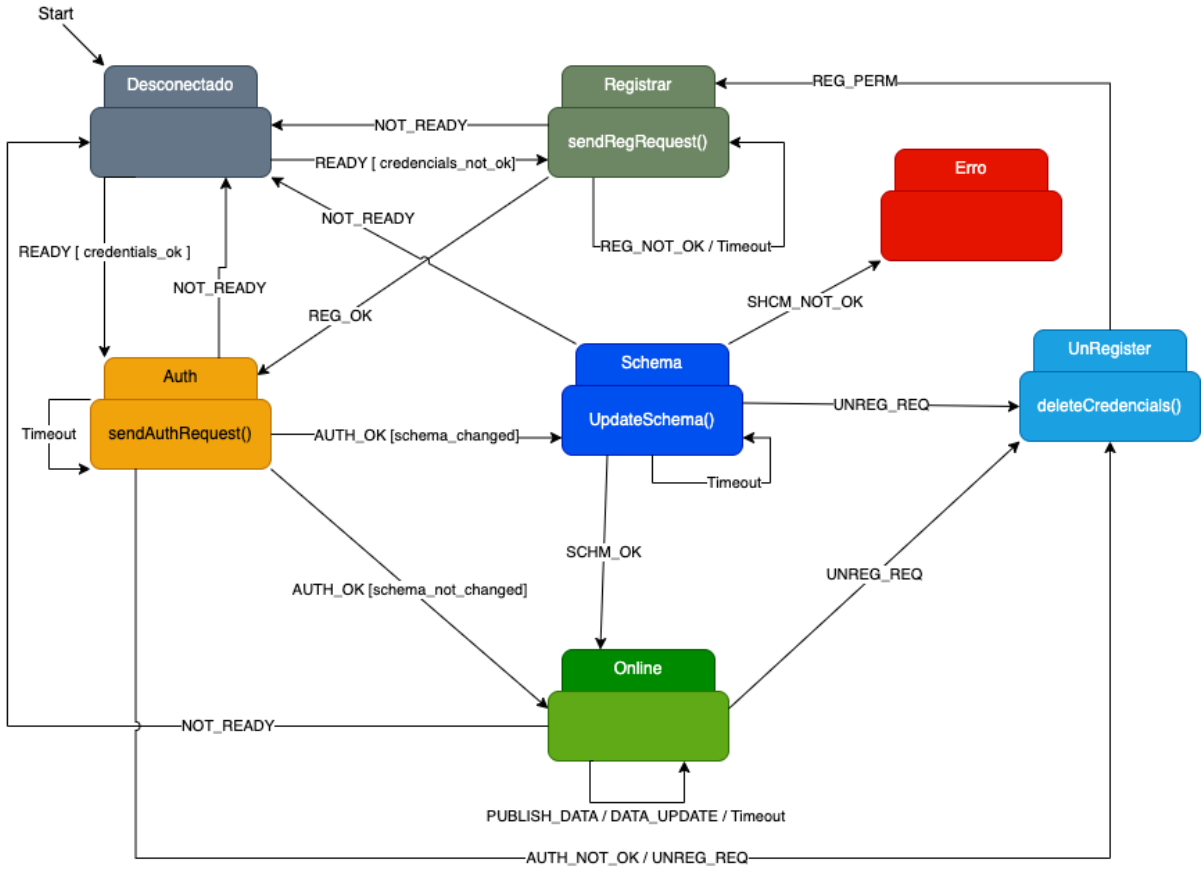

Figura 2.8 Estados de Autenticação *KNoT Virtual Thing* (CESAR, 2024).

3 Trabalhos Relacionados

No estudo de de Souza (2017), analisaram-se as necessidades de uma plataforma *IoT* de maneira abrangente, buscando atender às demandas da maioria dos usuários. Uma das qualidades mais destacadas pelo autor foi a disponibilidade da plataforma para se conectar a diferentes tipos de protocolos, possibilitando a recepção de dados de diversos dispositivos *IoT* e contando com a maior variedade possível de protocolos e plataformas. Exemplos como *realTime.io*, *SensorCloud* e *TempoDB* não apresentaram essa flexibilidade. Outro ponto abordado foi a licença de uso da plataforma, sendo a mais desejada a *open-source*, proporcionando maior liberdade aos desenvolvedores para aprimorar as ferramentas e adequá-las aos seus problemas. Nenhuma das plataformas já citadas permite esse desenvolvimento, assim como *ThingWorx*, *IFTTT*, *LinkSmart* e *Node-RED*, prendendo o desenvolvedor a soluções prontas e rígidas, embora possam ser suficientes para usuários comuns. O controle de acesso aos dados também é fundamental para que o usuário possa gerenciar suas aplicações, mas muitas dessas plataformas não oferecem essa liberdade. A *ThinkSpeak*, por exemplo, permite ao usuário um pequeno controle com uma interface para determinar permissões simples de escrita e leitura, enquanto outras, como *OpenRemote* e *LinkSmart*, deixam isso a cargo do desenvolvedor de aplicações (de Souza, 2017).

No entanto, o aspecto fundamental do trabalho de de Souza (2017) foi a análise das lacunas com foco no *KNoT*, onde o autor descreve que, apesar de ser uma plataforma heterogênea com diversos protocolos disponíveis para a coleta de dados, o protocolo *LoRaWAN*, explicado em detalhes pelo autor e indicado como um dos mais importantes LPWAN disponíveis, não está integrado ao *KNoT*, contando apenas com uma promessa de desenvolvimento. É destacado que, mesmo que o *KNoT* tenha, naquele momento, a deficiência de integração com protocolos, a arquitetura *open-source* da plataforma permite que novos protocolos sejam desenvolvidos e integrados, como o *LoRaWAN*, MQTT, etc (de Souza, 2017).

Nas considerações finais, o autor de de Souza (2017) descreve problemas do *KNoT*, como a camada de segurança e a falta de suporte à busca de composição de *streams* de dados. Conclui, no entanto, que a meta-plataforma é robusta e ainda está em desenvolvimento, mas possui todas as características desejadas para um desenvolvimento completo (de Souza, 2017).

No estudo de Oliveira and Silva (2020) apresenta uma análise detalhada das principais plataformas IoT baseadas em nuvem. Utilizando critérios de pontuação, a plataforma *Microsoft Azure IoT* foi considerada a mais satisfatória, superando a pontuação da plataforma *IBM Watson IoT* por uma margem estreita. *IBM Watson IoT, AWS IoT e*

Oracle Cloud IoT seguiram com pontuações próximas, seguidas pela solução do *Google* (Oliveira and Silva, 2020).

O estudo destaca que as plataformas *open-source* ocupam posições inferiores no ranking, com exceção da plataforma *Cisco Kinetic*, que não é *open-source*. O trabalho enfatiza que a posição no ranking não implica necessariamente na inadequação de uma plataforma para uma aplicação específica, e que até mesmo as plataformas classificadas mais baixas atendem à maioria dos critérios avaliativos (Oliveira and Silva, 2020).

Uma análise comparativa com trabalhos relacionados revela que o estudo do (Oliveira and Silva, 2020) se destaca pela quantidade de critérios avaliativos utilizados na comparação. Além disso, as plataformas *open-source* como *SiteWhere*, *KNoT* e *Macchina.io* não foram consideradas em outros trabalhos correlatos, evidenciando uma abordagem única. O estudo também ressalta a importância da data de publicação, indicando que pesquisas mais recentes são essenciais para refletir as características atuais das plataformas *IoT*, que estão em constante evolução. Assim, sendo o mais recente entre trabalhos relacionados, este estudo proporciona informações e tendências mais atualizadas (Oliveira and Silva, 2020).

No trabalho Batista et al. (2018), o autor aborda a plataforma *KNoT* e como ela oferece uma solução prática para integrar dispositivos físicos heterogêneos sem comunicação nativa via IP. Propõe complementar essa característica, tornando-a mais flexível e integrada ao ambiente *FIWARE*, permitindo que a *KNoT* acesse um ecossistema rico em ferramentas para o desenvolvimento de aplicações variadas. No entanto, a integração requer adaptações nas camadas inferiores da *KNoT* para interagir com o *Orion*, componente central do *FIWARE*. A Figura 3.1 ilustra a arquitetura da integração, destacando a importância da *KNoT Thing Library* e do *knotd* (Batista et al., 2018).

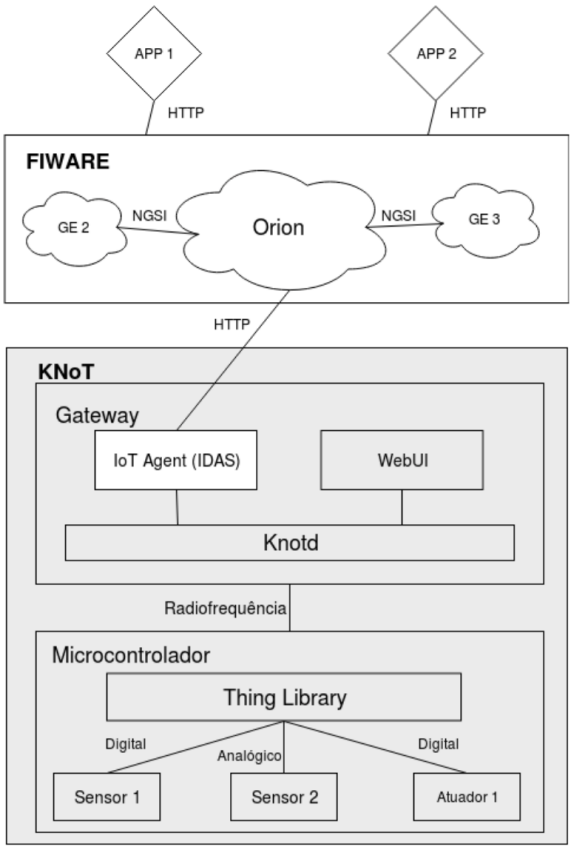

Figura 3.1 Arquitetura da integração entre *KNoT* e FIWARE (Batista et al., 2018).

O *KNoT*, executado no *Gateway*, foi modificado para suportar a integração com a *FIWARE*, traduzindo dados para o formato *JSON* antes de enviá-los ao *IoT Agent*, componente do IDAS (*Intelligence Data Advanced Solution*) da *FIWARE*. O *IoT Agent*, instanciado no *Gateway*, realiza funções de cadastro, atualização, exclusão e leitura para criar um contexto compatível com NGSI, desempenhando papel crucial na segurança e na comunicação entre o Orion e a *KNoT* (Batista et al., 2018).

Na validação da solução do Batista et al. (2018) inclui o desenvolvimento da aplicação LightsOut, que controla a iluminação de uma sala. A aplicação exemplifica a integração, modelando uma lâmpada como entidade no Orion e utilizando o *STH Comet* e *Cygnus* do Cosmos do *FIWARE* para dados históricos e séries temporais. A *LightsOut* ilustra o potencial de incremento que o ecossistema *FIWARE* oferece para desenvolvedores criarem novas aplicações de *IoT* (Batista et al., 2018).

4 Metodologia

A metodologia utilizada neste trabalho inicia-se com um estudo bibliográfico sobre o funcionamento da meta plataforma *KNoT* e *LoRaWAN*, detalhado no capítulo 2, assim como aplicações *IoT* na área industrial. A partir da análise de artigos sobre os temas, concluiu-se que seria de grande vantagem para a meta plataforma *KNoT* a adição de um protocolo *LPWAN*, sendo que o mesmo já estava no planejamento de desenvolvimento industrial. Por isso, todo esse projeto foi executado em uma planta industrial real de produção de sabão de uma grande marca situada na região metropolitana do Recife, e por questões legais não será diretamente referenciada, apenas chamada pelo codinome Fábrica Recife.

O objetivo deste trabalho é, a partir da infraestrutura *LoRaWAN*, realizar a coleta de dados dos sensores dispersos na planta industrial de produção de sabão da empresa Fábrica Recife e disponibilizar esses dados na *KNoT Cloud*. Para isso, o trabalho adotou a seguinte metodologia:

1. Definição da Arquitetura do Sistema;
2. Implementação de *Firmware* dos nós Lora;
3. Teste de viabilidade do *Gateway*;
4. Iniciar o *KNoT Cloud* em uma máquina local;
5. Conectar a chirpstak com devices (nós);
6. Integração do *KNoT* à *Chirpstack*;
7. Testes da plataforma integrada;
8. *Deploy* no Cliente;

4.1 Definição da Arquitetura do Sistema

A ideia principal é permitir com que o *KNoT* possa receber os dados dos sensores que contem comunicação *LoRa*. Como visto no estudo sobre o protocolo *LoRaWAN*, na infraestrutura do protocolo é exigido um servidor para o gerenciamento e segurança dos dispositivos, trabalho esse realizado pela *ChirpStack* que é um servidor *LoRaWAN*

de código aberto, permitindo a adição de código e adição de *features*. Também foi demonstrado que o *KNoT* recebe os dados a partir de uma conexão *AMQP* e tem em seu escopo de ferramentas desenvolvidas o *KNoT Virtual Thing* que executa uma máquina de estados de autenticação para poder enviar os dados para a *Cloud*. Assim, a partir das ferramentas apresentadas, foi pensada a seguinte arquitetura demonstrada na Figura 4.1 onde temos os dispositivos fazendo as leituras da rede elétrica na planta industrial e enviando para o *gateway LoRaWAN* via rádio com modulação *LoRa*, que por sua vez envia as informações sem alteração para a *ChirpStack* via UDP pelo *packet forwarder* instalado no *Gateway*, sendo recebida pelo *Gateway Bridge* e repassando para a *Network server*, onde é descriptografado a partir das chaves de segurança dos dispositivos para então seguir para o *Application Server* que é o local de *Decoder* das mensagens e traduzidas a partir das informações fornecidas pelo fabricante dos medidores de consumo elétrico. Isso resulta em um *JSON* que contém todos os dados das leituras e segue para a **Integração KNoT** para ser direcionada ao *KNoT Cloud*. A **Integração KNoT** é o resultado desse projeto, um módulo criado para permitir a comunicação entre os dispositivos *LoRaWAN* e a *KNoT Cloud*.

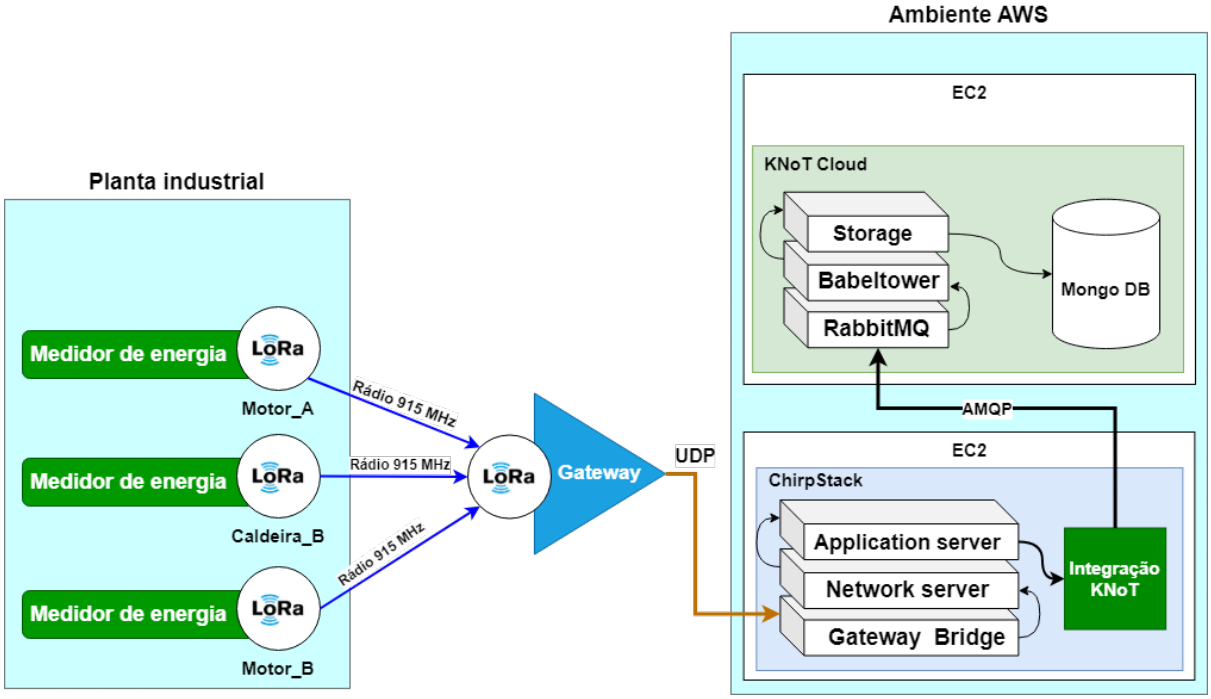

Figura 4.1 Arquitetura geral do sistema.

4.1.1 Integração *KNoT*

O sistema de medição de qualidade da rede elétrica trifásico é enriquecido pela integração com a plataforma *ChirpStack*, que oferece uma arquitetura modular para otimizar a gestão de dispositivos *LoRaWAN* que é ilustrada na 4.2 onde os três principais componentes da plataforma - *ChirpStack Gateway Bridge*, *ChirpStack Network Server*

e *ChirpStack Application Server* - desempenham papéis distintos, contribuindo para a eficiência e escalabilidade da solução.

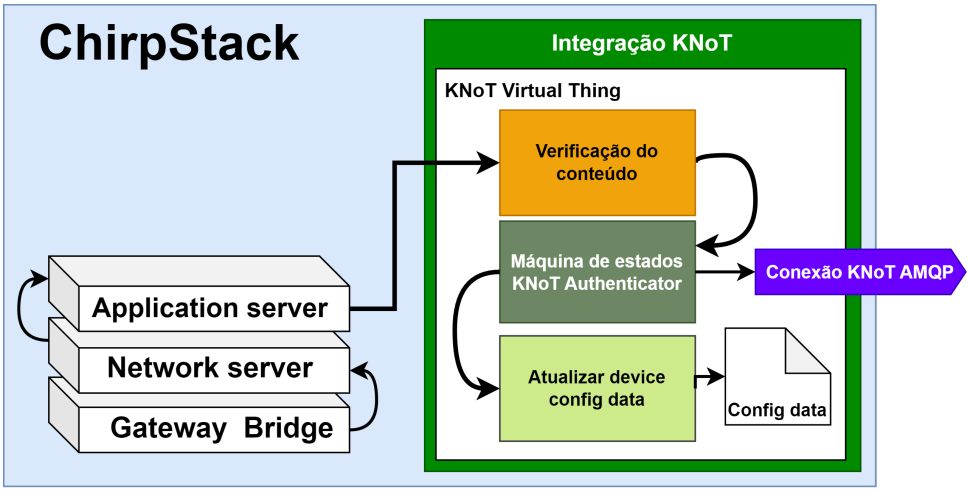

Figura 4.2 Integração *KNoT*.

O *ChirpStack Gateway* Bridge é responsável por receber as mensagens provenientes do *gateway LoRaWAN* de 16 canais, como mencionado anteriormente. Ele atua como uma ponte para traduzir e encaminhar esses dados para o *ChirpStack Network Server*. Este último, por sua vez, gerencia a conectividade com os dispositivos, controlando os aspectos de autenticação e autorização, garantindo a integridade dos dados e coordenando as transmissões na rede *LoRaWAN*.

O *ChirpStack Application Server* representa a camada final, concentrando-se na manipulação dos dados recebidos pelos dispositivos. Ele fornece uma interface para desenvolvedores e usuários finais, permitindo a integração de aplicativos personalizados. Essa modularidade da plataforma *ChirpStack* possibilita uma configuração flexível, adaptável às necessidades específicas do projeto.

A integração com *ChirpStack* também destaca a capacidade de conectar-se a outras ferramentas e serviços externos, ampliando as funcionalidades do sistema. Essa flexibilidade oferece oportunidades para integrações personalizadas, seja com sistemas de gerenciamento de energia, plataformas de análise de dados ou outras ferramentas relevantes para otimizar o monitoramento remoto da rede elétrica trifásica.

Nesse projeto, foi elaborada uma integração personalizada entre o *ChirpStack Application Server* e a plataforma *KNoT IoT*, com o intuito de fortalecer a interoperabilidade e ampliar as capacidades de comunicação do sistema de monitoramento de qualidade de rede elétrica. Essa nova integração proporciona uma rota eficiente para a transmissão de dados coletados pelos dispositivos, passando por uma série de processos para assegurar a qualidade e a formatação adequada das informações.

Os dados recebidos pelo *ChirpStack Application Server* são submetidos a uma verificação, onde são eliminados valores ausentes e formatos numéricos não adequados são corrigidos. Essa etapa visa garantir a integridade dos dados antes de serem encaminhados

para a plataforma KNoT IoT. Posteriormente, é implementada uma máquina de estados dedicada que desempenha um papel crucial na autenticação com a plataforma *KNoT IoT*.

A máquina de estados não apenas autentica os dados, mas também realiza a formatação necessária para uma transmissão eficaz para a plataforma *KNoT IoT*. Além disso, ela desempenha um papel fundamental na atualização do documento de configurações associado a cada dispositivo, conhecido como *KNoT Device*. Esse documento de configurações é vital para garantir a coesão e a consistência no gerenciamento de cada dispositivo conectado, refletindo diretamente na qualidade e na confiabilidade do monitoramento remoto da rede elétrica trifásica.

Essa nova integração representa não apenas um avanço técnico, mas também a capacidade de personalizar e estender a funcionalidade do sistema, proporcionando uma solução mais completa e adaptável às necessidades específicas do projeto em questão.

4.1.2 Medidor de Energia

O medidor trifásico utilizado foi o ITE 11LI da Figura 4.3 que opera por meio da integração de sensores precisos de tensão, corrente, energia ativa, reativa e fator de potência. Esses sensores coletam dados em tempo real, que são processados por um microcontrolador embarcado. O microcontrolador realiza a análise dos parâmetros elétricos e empacota as informações em pacotes *LoRaWAN*. Utilizando um módulo de comunicação *LoRaWAN* a 915MHz, o dispositivo transmite esses pacotes para um *gateway* central. A tecnologia *LoRaWAN* assegura uma transmissão de dados eficiente em longas distâncias, facilitando o monitoramento remoto da qualidade da rede elétrica.

Figura 4.3 Medidor de consumo e qualidade elétrica ITE 11LI (khomp)

A comunicação *LoRaWAN* a 915MHz é fundamental para viabilizar a conectividade sem fio entre o medidor e o *gateway* como lista a Figura 4.4. Este último recebe os pacotes de dados, realiza a decodificação e encaminha as informações para um sistema centralizado de monitoramento. Dessa forma, o medidor proporciona uma solução avançada e acessível para o monitoramento remoto da qualidade da rede elétrica, apresentando potencial

aplicação em diversos setores industriais e comerciais.

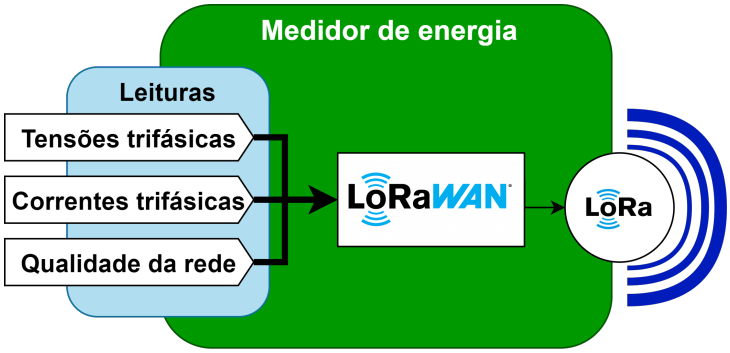

Figura 4.4 Medidor da qualidade elétrica.

4.1.3 *Gateway LoRaWAN*

O medidor de qualidade de rede elétrica trifásico proposto é aprimorado pela integração com um *gateway LoRaWAN IP65 Outdoor Gateway for LoRaWAN RAK7240* da Figura 4.5 de 16 canais, elevando a eficiência e a capacidade de gerenciamento da comunicação sem fio. Este *gateway* atua como ponto de acesso central para coletar dados provenientes do medidor, permitindo uma comunicação mais robusta e eficiente em ambientes complexos. Além disso esse modelo conta com um *buffer* especial para armazenar mensagens em caso de queda de rede, além de 3 interfaces diferentes para acesso a internet como 4G, *Wireless Fidelity (Wi-Fi)* e *ethernet* podendo fazer o chaveamento entre elas em caso de perda de conexão.

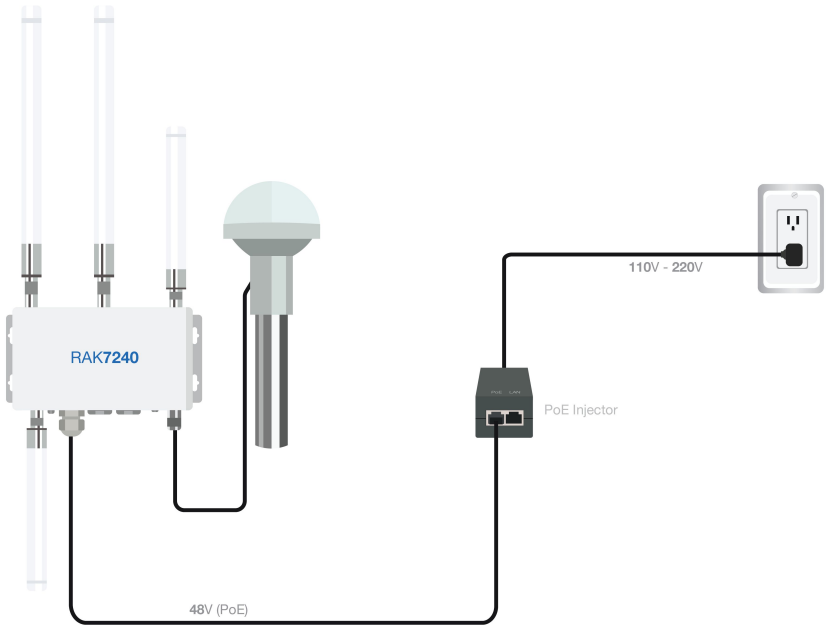

Figura 4.5 *Gateway LoRaWAN RAK7240* (RAK).

A Figura 4.6 mostra o modelo do dispositivo que utiliza um software de pacote conhecido como *"packet forwarder"* para direcionar as mensagens *LoRaWAN* de forma adequada. Esse componente desempenha um papel crucial no encaminhamento dos pacotes de dados coletados pelo medidor para os canais disponíveis no *gateway*. A utilização de um *gateway* de 16 canais proporciona maior largura de banda e capacidade de processamento, permitindo a gestão simultânea de múltiplos dispositivos e minimizando a interferência entre canais.

Além disso, o medidor emprega funcionalidades avançadas do *LoRaWAN*, como a capacidade de otimização de tempo de atividade (*Duty Cycle*), para garantir o cumprimento das regulamentações locais e maximizar a eficiência da transmissão. A implementação dessas funcionalidades avançadas não apenas melhora a confiabilidade da comunicação, mas também contribui para a eficácia global do sistema, tornando-o uma solução robusta para o monitoramento remoto da qualidade da rede elétrica em ambientes diversos.

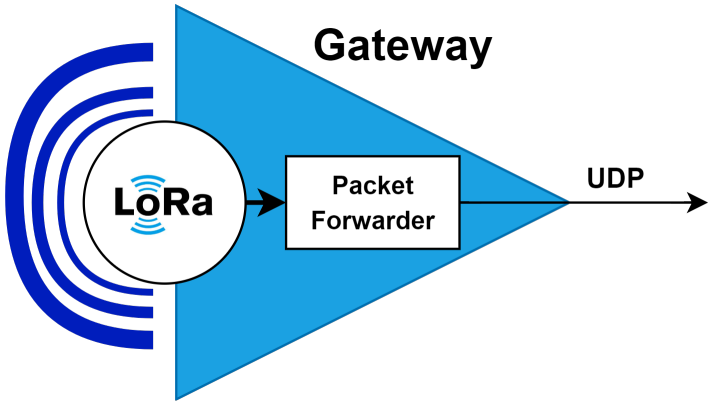

Figura 4.6 *Gateway LoRaWAN.*

4.2 Implementação de *Firmware* dos nós Lora

No contexto deste projeto, foi importante realizar uma validação abrangente da comunicação *LoRaWAN*, bem como a análise e configuração adequada para a integração na planta industrial do cliente. Para facilitar esse processo, desenvolveu-se um *firmware* específico destinado a um dispositivo final *LoRaWAN*, permitindo um controle detalhado da comunicação durante os testes.

O *firmware* foi programado em linguagem C e compilado pela IDE do *Arduino* para o microcontrolador *ESP32*, que incorpora um rádio *LoRa*. Esse dispositivo final foi equipado com um pequeno monitor LCD para proporcionar uma interface visual como mostra a Figura 4.7, exibindo informações cruciais sobre as configurações e dados da comunicação. Além disso, a inclusão de botões no sistema permitiu a manipulação direta, oferecendo funcionalidades como a inicialização de uma nova comunicação, a recepção de *acknowledgments* da rede e a configuração dinâmica de parâmetros essenciais, como potência de transmissão e modulação.

Figura 4.7 End Device Esp32 LoRa Oled Heltec.

A implementação do *firmware* não apenas simplificou os testes de comunicação, mas também permitiu uma avaliação precisa das distâncias alcançadas pela rede *LoRaWAN* na planta industrial. A capacidade de manipular diferentes configurações em tempo real e visualizar as informações relevantes no monitor LCD ofereceu uma abordagem prática para otimizar o desempenho da rede em ambientes industriais desafiadores.

Essa etapa de validação não apenas certificou a eficácia da comunicação *LoRaWAN* na planta industrial, mas também proporcionou uma base sólida para ajustes finos e personalizações necessárias para atender às necessidades específicas do ambiente operacional do cliente.

4.3 Teste de viabilidade do *Gateway*

Para testar a viabilidade do projeto, foi inicialmente preparada uma primeira averiguação com o objetivo de ver a viabilidade do protocolo *LoRaWAN* na planta industrial Fábrica Recife, que apresenta uma planta industrial com as seguintes dimensões: 81.000 m^2 e raio de 200 m. O objetivo geral desse primeiro teste desdobra-se nos objetivos específicos descritos abaixo:

- Avaliar a viabilidade do uso de equipamentos *LoRaWAN* para a coleta de informações em longas distâncias;
- Mensurar a distância média de captura de dados usando um *gateway indoor* e sensores de temperatura e umidade.

Equipamentos utilizados:

- *Gateway LoRaWAN* Dragino LPS8 Indoor 915 MHz da Figura 4.8;
- Antena Móvel UHF 5/8 de onda WHIP 900 MHz da Figura 4.9;
- *End Device Esp32 LoRa Oled Heltec* (ESP com *firmware* de teste de conexão, simulando um *end device* OTAA Figura 4.10.

Para controle e análise da rede *LoRaWAN*, foi utilizada a plataforma *ChirpStack* rodando localmente em um notebook. Todas as configurações foram feitas para direcionar os dados a ela. O *gateway LPS8v2* da Figura 4.8 foi configurado conforme o manual para o padrão Australiano, que é o mesmo utilizado no Brasil, repassando os pacotes via *Semtech UDP (User Datagram Protocol) packet forwarding protocol* para a plataforma. Assim, todos os equipamentos estavam na mesma rede Wi-Fi disponibilizada por um *smartphone* para a comunicação dos dados via User Datagram Protocol (UDP).

Figura 4.8 *LPS8v2 – Indoor LoRaWAN Gateway* Dragino Technology Co.

O sensor foi configurado para o padrão *OTAA* e intervalos de 1 minuto para maior velocidade nas confirmações de recebimento de mensagens. A Figura 4.9 mostra a antena do *gateway* posicionada a 3,5 metros do solo, em cima de uma pequena sala de operação. Os testes foram realizados a partir da movimentação do dispositivo com *firmware* modificado da Figura 4.10 e leitura das respostas que chegavam e eram mostradas no pequeno monitor LCD do dispositivo, assim como mensagens de time-out de 1 minuto. O primeiro local do teste foi na região do *Reator Slurry* (tipo de reator químico que envolve a suspensão de sólidos em um líquido) com o sensor a 2 m do solo, na frente do galpão, a 178 m do *gateway*; em seguida, o mesmo foi realocado para o piso superior na sala com os painéis de controle, dentro do galpão e ao lado do motor no piso mais alto; e, por fim, na região de líquidos.

Figura 4.9 Antena do *gateway LoRaWAN*.

Figura 4.10 Teste ao lado do motor no piso superior do *slurry*.

Durante o teste, as configurações do dispositivo foram modificadas para se obter um melhor sinal no ponto analisado. Assim, uma das principais configurações testadas foi o *SF* (*Spread Factory*), variando de 7 a 12, sendo o 12 utilizado para longas distâncias ou locais com maior interferência de sinais ou ambiente. O teste foi bem-sucedido; a conexão do sensor junto ao *gateway* (ponto azul) foi estabelecida em todos os pontos testados (em verde), como mostra a Figura 4.11. O resultado é mais detalhado na seção 5.

Figura 4.11 Pontos de testes com o dispositivo.

4.4 Iniciar o *KNoT Cloud* em uma máquina local

A ideia é ter o *KNoT Cloud* em um notebook local para que se possa desenvolver a nova integração na *Chirpstack*. O repositório do *KNoT Cloud* foi então clonado e instalado, juntamente com todas as suas dependências, no notebook. A *Cloud* mostrou-se bastante interessante, pois é baseada em microsserviços, assim como a *ChirpStack*, com a utilização de vários desses serviços em *containers dockerizados*. Isso ajudou na instalação posterior da *ChirpStack*, pois as duas ferramentas são baseadas em *Docker*. A Figura 4.12 mostra o *log* da *babeltower*, que é o orquestrador das comunicações do *KNoT Cloud*. Nesse caso,

estão sendo filtrados os endereços de um dispositivo para ver as mensagens que chegaram do mesmo. Na última linha é possível notar os dados chegando de um dos *Devices* da rede comum do *KNoT* com *SensorId* 1 e nome "Peso Total".

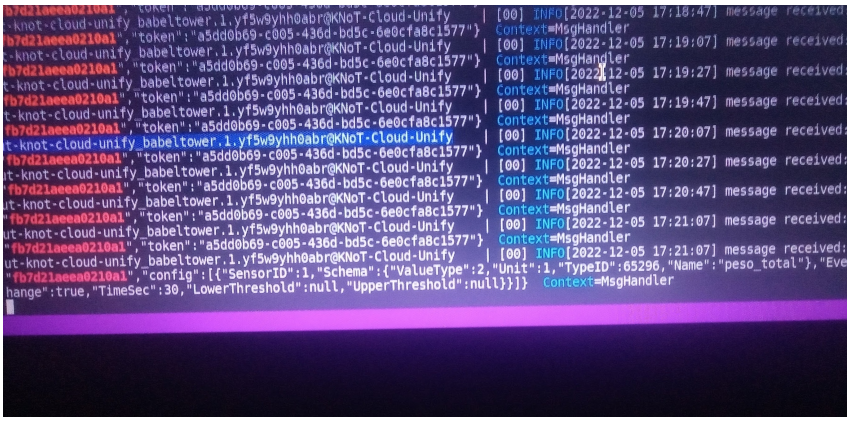

Figura 4.12 *KNot Cloud* em máquina local com o serviço *Babeltower*.

4.5 Conectar a *ChirpStack* com *devices* (nós)

A *ChirpStack* assim como o *KNoT Cloud*, é baseada em microsserviços em *containers dockerizados*. O mais importante para esse trabalho é o *Chirpstack Application server*, camada responsável pelas integrações com outras plataformas. Depois de instalar todas as dependências, foi só inicializar o arquivo do *docker compose*, a Figura 4.13 mostra o log de inicialização do *docker compose* onde cada serviço é inicializado sem erros pela indicação do *"done"*, todos os serviços foram inicializados sem erros.

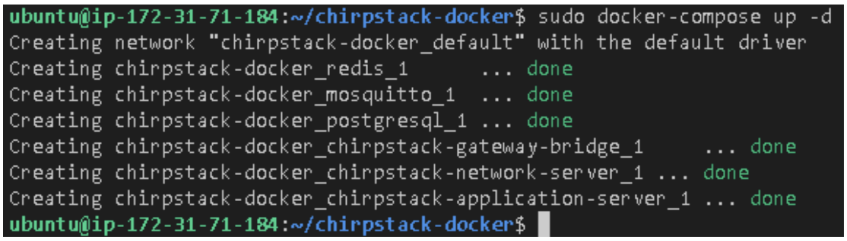

Figura 4.13 *ChirpStack Docker Compose up*.

4.5.1 *ChirpStack Application Server*

Essa camada de serviço da *ChirpStack* é responsável pela gerência dos dispositivos, criptografia dos *payloads* e integrações para envio de dados para outras aplicações, utilizando meios como *MQTT, Kafka, AWS SNS*, além de poder enviar requisições para gravar dados em bancos de dados como *PostgreSQL* e *MongoDB*. Todas essas integrações são escritas em linguagem de programação *Golang*.

4.6 Integração do *KNoT* à *Chirpstack*

A ideia principal para criar o link entre as plataformas é utilizar o protocolo *AMQP*, comum aos dois, para que os dados possam fluir. A Figura 4.14 ilustra de modo simples a ideia geral do projeto, a partir da leitura de dados de um medidor de energia, para o *gateway* que é gerenciado pela *ChirpStack*, para então chegar ao *KNoT*.

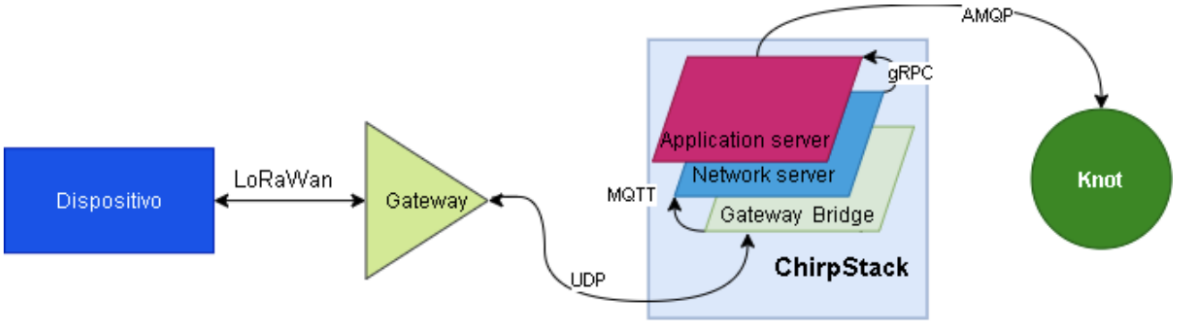

Figura 4.14 Fluxo dos dados.

Para que haja a comunicação com o *KNoT*, é preciso seguir o protocolo de autenticação da comunicação da plataforma. Assim, foi utilizada a máquina de estados do *KNoT VIrtual Thing* da Figura 2.8 como inspiração, sendo implementada uma máquina de estados com alguns passos a menos, mas com todo o contexto importante para a segurança e comunicação dos dados, como mostra a Figura 4.15. Dessa forma, os dados não são perdidos, pois não há o *UnRegister*. Se um dispositivo com *DevEUI* já cadastrado pode ser recadastrado, pois internamente será gerado um ID para cada registro feito no *KNoT* pela *ChirpStack*.

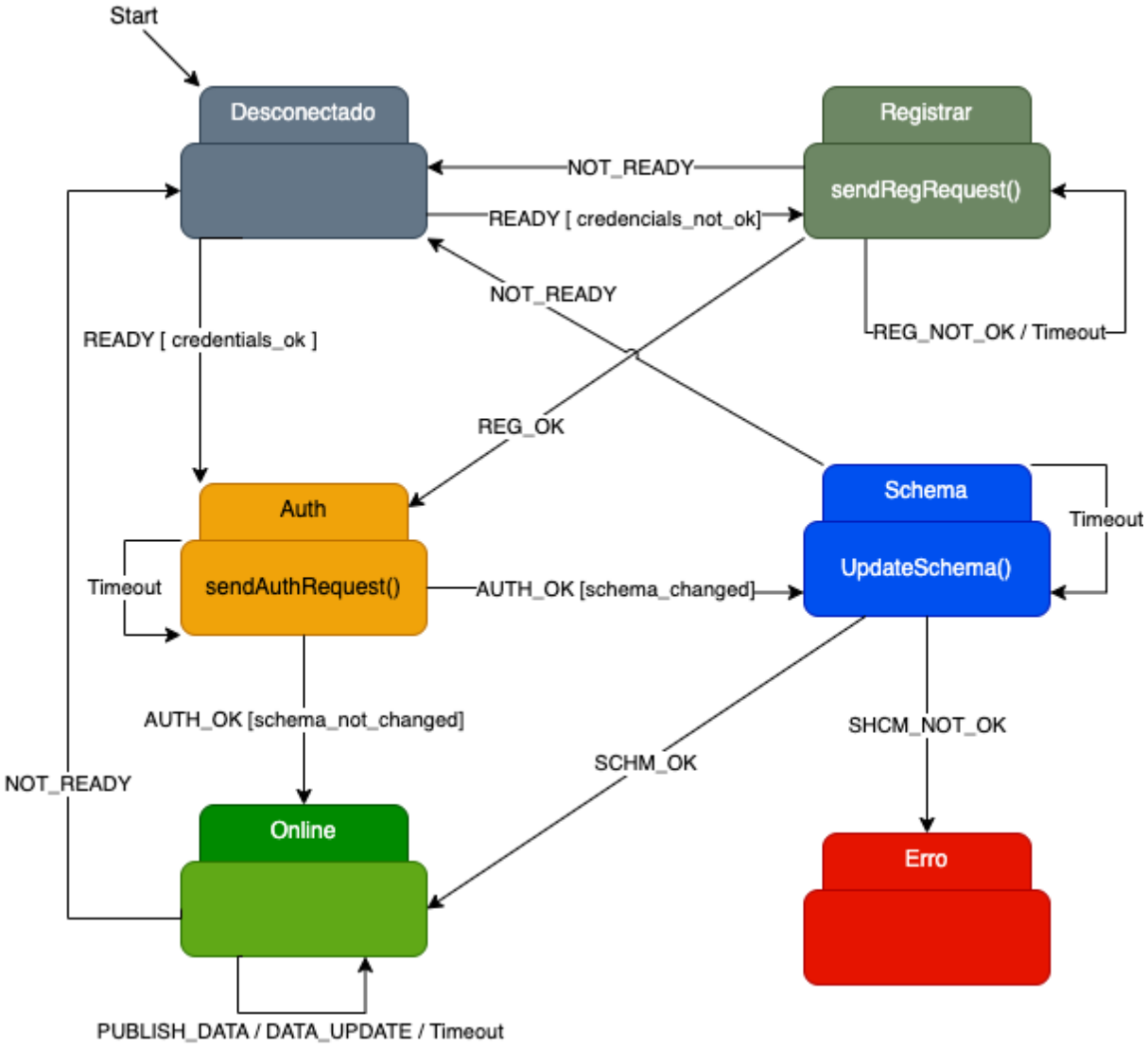

Figura 4.15 Máquina de estados para comunicar com o *KNoT*.

Foi implementado o protocolo de autenticação *KNoT* em *Golang* na *Chirpstack* seguindo a representação da Figura 4.15, onde, por meio do *AMQP*, estabeleceu-se o link, permitindo assim que a camada de aplicação da *ChirpStack* pudesse enviar os dados dos sensores. O código 1 mostra uma das funções da máquina de estados implementada. A partir da linha 7 é feita uma verificação do estado do dispositivo, caso seja novo, como na linha 10, ele publica um pedido de registro na *KNoT Cloud* enviando via AMQP, em um outro ponto do código a resposta será tratada, caso toda a sua configuração esteja correta a resposta será positiva; na linha 14 o *device* publica um pedido de autenticação, reenviando seus dados e recebendo um *device token* de autenticação; que em seguida é utilizado para a autenticação, esse *device token* deve ser armazenado de forma permanente em um documento de configuração. Depois de autenticado o *device* vai estar no estado *KNoTOK* e poderá enviar mensagens de forma constantes para a *KNoT Cloud*. Mas caso

haja um erro que não tenha sido tratado antes, segue para a linha 25 onde é armazenado e mostrado na tela do *log*. E caso o dispositivo esteja desativado, fica sempre na linha 32, preso e sem ações.

```
1  // Control device paths.
2  func knotStateMachineHandler(deviceChan chan entities.Device, p *protocol) {
3          for device := range deviceChan {
4
5                  device = p.validateKnotDevice(device)
6
7                  switch device.State {
8
9                  // If the device status is new, request a device registration
10                 case values.KNoTNew:
11                         p.sendKnotRequests(deviceChan, device.State,
                           ↪  values.KNoTWaitReg, device)
12
13                 // If the device is already registered, ask for device
                   ↪  authentication
14                 case values.KNoTRegistered:
15                         p.sendKnotRequests(deviceChan, device.State,
                           ↪  values.KNoTWaitAuth, device)
16
17                 // Now the device has a token, make a new request for
                   ↪  authentication.
18                 case values.KNoTAuth:
19                         p.sendKnotRequests(deviceChan, device.State,
                           ↪  values.KNoTWaitConfig, device)
20
21                 // Send the new data that comes from the device to Knot Cloud
22                 case values.KNoTOk:
23                         p.publishData(device)
24
25                 // Handle errors
26                 case values.KNoTError:
27                         log.WithFields(log.Fields{"knot":
                           ↪  values.KNoTError}).Error("ERROR WITHOUT HANDLER: " +
                           ↪  device.Error)
28                         device.State = values.KNoTError
29                         device.Error = values.UnhandledError
30                         p.updateDevice(device)
31
32                 case values.KNoTOff:
33
34                 }
35         }
36 }
```

Listing 1: Código em Golang.

O protocolo *LoRaWAN* exige uma quantidade de informações do dispositivo para que o mesmo possa ingressar na rede, e a *ChirpStack* conta com uma interface web intuitiva para a configuração dessas informações, como *DevEUI*, chaves de segurança e dados relevantes à comunicação da rede. O *KNoT* não é diferente. O *KNoT thing* tem sua estrutura própria com definição de *schemas* para a leitura de sensores presentes no dispositivo, assim como ID, nome e *token*. Para comunicar com o *KNoT*, a *ChirpStack* precisa ter registrados esses dados, assim como as chaves de autenticação requeridas pelo *KNoT*. Para escrever essas configurações do dispositivo *KNoT*, foi optada uma solução mais rápida para o desenvolvimento, a edição de arquivos.

Aproveitando um arquivo de configuração já utilizado pela *ChirpStack* para editar quais integrações serão ativadas e suas configurações, foi criada a estrutura do *Device KNot* no arquivo de configuração *chirpstack-application-server.tom* como mostra o Código 2 que é um arquivo de configuração padrão da *ChirpStack* no formato *TOM*. A linha 1 é como uma seção da configuração assim como as linhas 4, 9, 14, 17, 23 e 27. Na linha 2 é ativada a integração *KNoT* e nas linhas 5, 6 e 7 são informadas as configurações necessárias para estabelecer uma comunicação com o *KNoT Cloud* como endereço do *broker AMQP* e *user token* fornecido ao usuário da *KNoT Cloud* para autenticação. Em seguida na linha 10 é informado o DevEUI do primeiro dispositivo LoRaWAN, sendo esse valor único para capara aparelho e é fornecido pelo fabricante. O *token* da linha 11 será recebido após o registro do dispositivo na *Cloud*, e o nome do dispositivo. Na linha 14 se inicia a configuração de cada sensor desse dispositivo, podendo ter N sensores. Na linha 15 é informado o ID do primeiro sensor e na linha 17 se inicia as características desse sensor, como o tipo de leitura na linha 18, unidade na linha 19 e outras informações encontradas na tabela do CESAR. Na linha 23 e 24 são encontradas variáveis internas ao protocolo *KNoT* para verificação das configurações.

```
[application_server.integration]
enabled     = ["knot"]

[application_server.integration.knot]
url         = "amqp://knot:knot@192.168.0.100:5672"
name        = "knot lora"
user_token  = "eyJhbGciOiJIUZIlNiIsInR5CCI6IkpXVCJ9.eyJleHAiOjE"

[[application_server.integration.knot.devices]]
deveui      = "af502ca3ca41a841"
token       = ""
name        = "loral"

[[application_server.integration.knot.devices.config]]
sensorId = 1

[application_server.integration. knot.devices.config.schema]
valueType   = 2
unit        = 0
typeId      = 65296
name        = "sensor_teste"

[application_server.integration. knot.devices.config.event]
change      = true
timeSec     = 12

[[application_server.integration. knot.devices ]]
deveui  = "1d37d3ab3f9f71cd"
token = ""
name = "lora2"
```

Listing 2: Configuração Application Server.

4.7 Testes da plataforma integrada

O primeiro teste de conexão utilizando apenas um *Thing KNoT* validou a comunicação entre as duas plataformas, conforme ilustrado as áreas destacadas da Figura 4.16, onde a comunicação foi estabelecida e autenticada a partir do *token* gerado pela *KNoT Cloud*, e o registro do *device* também ocorreu com sucesso, como mostra a Figura 4.17.

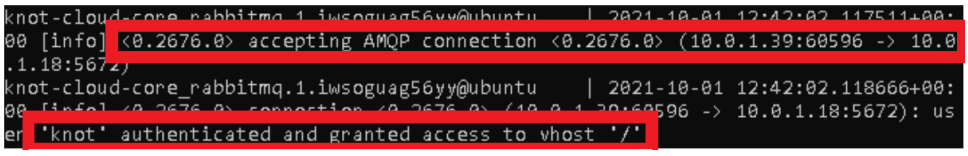

Figura 4.16 *KNoT BabelTower* conectada à *ChirpStack*.

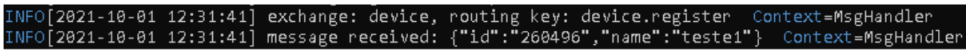

Figura 4.17 *KNoT BabelTower* registrando *Device*.

A *ChirpStack* conseguiu gerenciar as mensagens recebidas e executar o cadastro com o armazenamento do *Thing Token* proveniente da *Cloud*, conforme apresentado a região em destaque na Figura 4.18 onde foi mostrado no log um primeiro teste de registro, e como nessa versão antiga havia uma opção de *unregister*, o mesmo foi testado também assim como os outros canais de comunicação como autenticação e *update* de configurações. Assim como a verificação de novas mensagens, destacado em um retângulo amarelo logo abaixo da região maior.

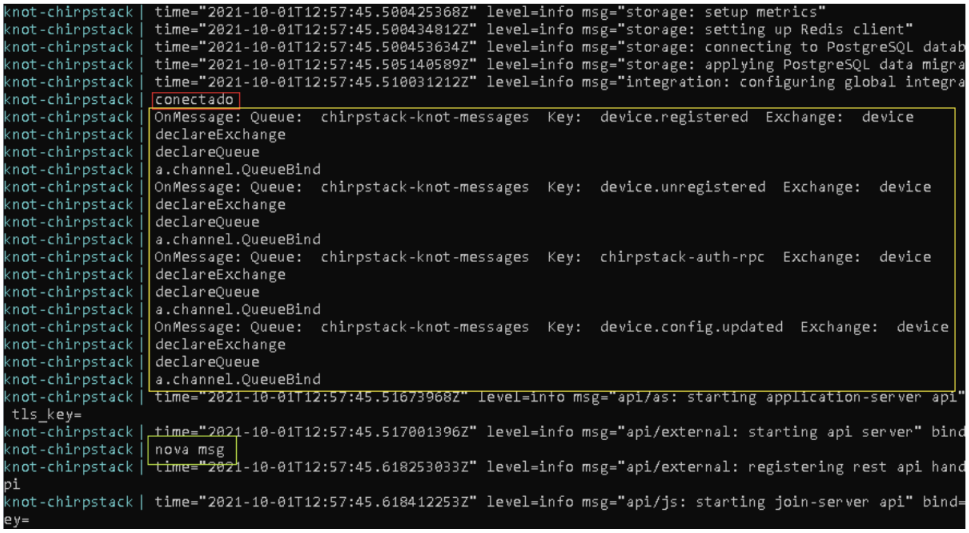

Figura 4.18 *ChirpStack* registrando *Device*.

De forma resumida, antes de ativar o *ChirpStack*, o arquivo *chirpstack-application-server.toml* foi configurado com as informações de ativação e autenticação para a integração do *KNoT*. Foi necessário ter as informações de conexão do *Broker*, um nome qualquer e o *token* de usuário gerado diretamente na *KNoT Cloud*. As informações dos dispositivos *LoRaWAN* configurados na *ChirpStack* também precisam ser adicionadas a esse documento, contendo as descrições de cada sensor de leitura feita pelo aparelho, onde cada sensor do *KNoT Thing* é uma leitura de dado, como temperatura, umidade, etc. Foi utilizado o padrão *KNoT* para indexar o tipo de leitura a partir do *schema* próprio CESAR. Dessa forma, ao iniciar a aplicação, estabeleceu a conexão com a *KNoT Cloud* e iniciou o processo de registro e autenticação dos dispositivos, ficando no aguardo pela chegada dos dados.

4.8 *Deploy* no Cliente

Um medidor de consumo ITE 11L1 (Khomp) foi instalado em uma das caldeiras da Fábrica Recife para realizar a coleta de 16 dados diferentes na rede trifásica. Os dados medidos em cada fase incluíram tensão, corrente, frequência, energia ativa, reativa e fator de potência, além da temperatura local. O sensor foi instalado no painel do equipamento, conforme mostrado na Figura 4.19.

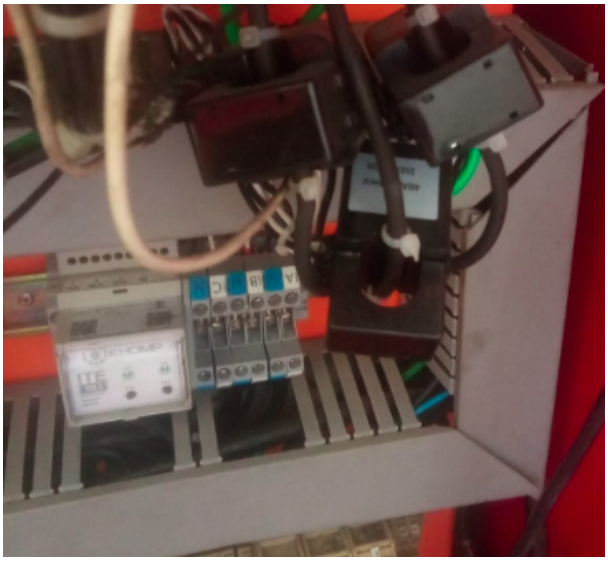

Figura 4.19 Sensor de consumo instalado no painel.

A 20 metros do *Gateway LoRaWAN*. O medidor suporta a rede *LoRaWAN* e foi configurado na rede com ativação ABP, conforme indicado no manual, e descrito como um *KNoT Thing* contendo 16 sensores no arquivo de configuração na *ChirpStack Application Server*.

A Figura 4.20 mostra as leituras do sensor da qualidade da rede na *ChirpStack Application Server* após serem descriptografadas e encaminhadas para a *KNoT Cloud*. É possível ver as leituras de tensão da fase A com 241,6 Volts, frequência geral da rede em 60 Hz, temperatura dentro do painel e corrente, mas nesse momento as leituras de corrente estavam com um defeito, os técnicos que instalaram o medidor acabaram curto-circuitando os leitores de corrente, não houve danos, mas após a descoberta tudo voltou ao normal e as leituras de corrente voltaram ao valores normais.

Figura 4.20 Leituras do sensor de consumo.

A Figura 4.21 mostra a recepção de uma mensagem registrada na *BabelTower* proveniente de outros sensores. É possível ver o ID do sensor e o valor da leitura na segunda linha na *label value* é possível encontrar o valor da temperatura como 37,9.

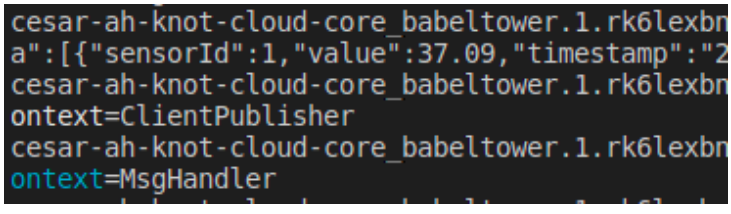

Figura 4.21 Dados recebidos de outro *Device*.

A comunicação ocorreu sem problemas; o medidor conseguiu enviar os dados com um intervalo de 5 minutos para a *KNoT Cloud* de forma contínua. Entretanto, entre o medidor e o *Gateway*, houve alguns problemas de duplicidade nas mensagens. Para garantir a obtenção de todos os dados, a configuração de *ACK* foi utilizada. Algumas mensagens enviadas pelo medidor não eram confirmadas a tempo, levando ao acúmulo de várias mensagens duplicadas. No total foram instalados 7 dispositivos, sendo 4 deles medidores de qualidade da rede e 3 medidores de vibração e temperatura.

5 Resultados

5.1 Ambiente experimental

O ambiente experimental para o teste do projeto ocorreu em uma planta industrial de fabricação de sabão localizada em Pernambuco. A instalação industrial apresentava um layout típico de uma fábrica de médio porte, com uma variedade de equipamentos industriais, motores elétricos e caldeiras de vapor distribuídos ao longo do espaço.

5.1.1 Características do Ambiente

1. **Equipamentos Industriais:** A planta possuía uma série de equipamentos industriais, incluindo misturadores, extrusoras, secadores e embaladoras, cada um impulsionado por motores elétricos de diferentes capacidades.

2. **Caldeiras de Vapor:** O processo de fabricação envolvia o uso de caldeiras de vapor para diversas etapas, como a secagem dos ingredientes e o aquecimento de misturas. Estas caldeiras eram parte integrante do sistema, contribuindo para a complexidade e diversidade do ambiente.

3. **Infraestrutura Elétrica:** A planta estava equipada com uma infraestrutura elétrica robusta, composta por painéis de distribuição, quadros elétricos e fiação que alimentava os motores elétricos e outros dispositivos presentes na instalação.

4. **Gateway LoRaWAN:** Para realizar a coleta de dados, foram instalados pequenos leitores de consumo de energia em vários equipamentos industriais. Os dados coletados eram transmitidos através de um gateway LoRaWAN, que atuava como o ponto de conexão entre os dispositivos de leitura e a rede local da planta.

5. **Conectividade com a Nuvem:** A planta possuía uma rede local com acesso à internet para possibilitar a transmissão dos dados coletados para a nuvem. Isso permitia o monitoramento remoto e a análise dos dados de consumo de energia em tempo real.

5.1.2 Equipamentos Utilizados nos Testes

- Motores de médio porte de corrente inferior a 50 amperes.

- Medidores de qualidade de energia ITE 11LI equipados com rádios *LoRa* para transmitir os dados para um *gateway LoRaWAN*.
- *IP65 Outdoor Gateway for LoRaWAN* conectado à rede local.

5.1.3 Objetivo dos Testes

O principal objetivo dos testes era avaliar a eficácia da instalação dos leitores de consumo de energia nos diversos equipamentos industriais. Além disso, a utilização da tecnologia *LoRaWAN* e a transmissão dos dados para a nuvem foram testadas para garantir a eficiência do sistema de monitoramento remoto.

5.1.4 Resultado Esperado

Esperava-se que os pequenos leitores de consumo de energia fornecessem dados precisos e confiáveis, permitindo uma análise detalhada do consumo energético em cada equipamento. A transmissão dos dados para a nuvem por meio do *gateway LoRaWAN* deveria ocorrer de maneira estável, proporcionando uma visão abrangente do desempenho energético da planta industrial.

5.2 Cenários de Teste

A busca contínua por eficiência energética e monitoramento inteligente tem impulsionado o desenvolvimento de soluções inovadoras para otimizar processos industriais. No contexto da fabricação de sabão em pó na planta industrial de Pernambuco, a implementação de pequenos leitores de consumo de energia, integrados a uma rede *LoRaWAN*, oferece uma abordagem promissora para a gestão eficaz de recursos.

- **Teste de Viabilidade de Comunicação:** Avaliou-se a robustez da comunicação entre os leitores de consumo de energia e o *gateway LoRaWAN*. Este teste buscou verificar a estabilidade e a confiabilidade da transmissão de dados em um ambiente industrial dinâmico.
- **Teste de Estresse:** O sistema foi submetido a condições de estresse, simulando situações de demanda intensificada e ambientes operacionais desafiadores. O objetivo era verificar a resposta do sistema sob pressão e identificar possíveis pontos de melhoria.

Esses cenários de teste abrangentes foram projetados para validar a eficácia e a confiabilidade do sistema de monitoramento energético em um ambiente industrial real, contribuindo para aprimorar a eficiência operacional e promover práticas sustentáveis na produção de sabão em pó.

5.2.1 Teste de Viabilidade de Comunicação

No primeiro teste de validação da comunicação entre o dispositivo e o *gateway*, foi considerado o estabelecimento da conexão entre o *end device LoRaWAN* e a plataforma *ChirpStack* por *OTAA*, com alterações no SF (*Spreading Factor*). O dispositivo obteve conexão em todos os pontos testados. Na Figura 5.1 o ponto azul representa a posição do *gateway*, e os pontos verdes representam os testes de conexão e envio de pacotes, nos quais pelo menos um pacote foi enviado, e uma mensagem de confirmação foi recebida pelo dispositivo. Também na Figura 5.1 são destacados os SF utilizados, sendo a área amarela com SF7, amarela com SF8 e vermelha com SF9. Sendo que o SF9 precisa de maior potência para poder chegar ao dispositivo.

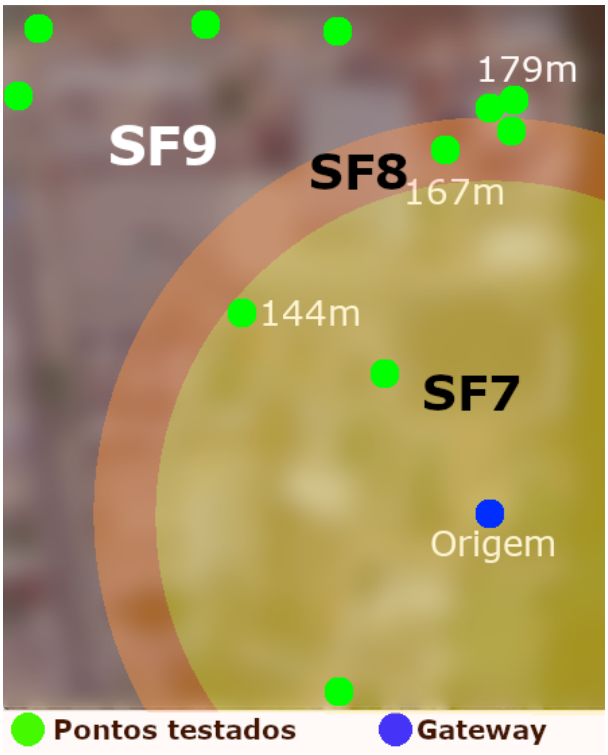

Figura 5.1 Mapeamento do SF.

A maior distância testada com êxito foi de 300 metros em relação ao *gateway*, conforme mostrado na Figura 5.2. Algumas áreas exigiram um SF maior (*Spreading Factor*), aumentando o gasto de energia para enviar os dados e apresentando maior dificuldade no envio e recepção.

Figura 5.2 Maior distância testada com êxito.

5.2.2 Teste de Estresse

A comunicação com 8 dispositivos reais mostrou-se estável e operante sem problemas no envio de dados da *ChirpStack* para a *KNoT Cloud* durante a implementação da leitura dos medidores na planta industrial da Fábrica. No entanto, para uma avaliação mais aprofundada da implementação, foram realizados testes de estresse utilizando um simulador de dispositivos (Brocaar) para criar um ambiente com 100 dispositivos configurados e enviando dados.

Três testes foram realizados com o simulador, conforme apresentado na Tabela 5.1. Nos dois primeiros testes, tudo ocorreu conforme o esperado, com os dispositivos sendo registrados, autenticados e seus dados armazenados na *KNoT Cloud* sem problemas. Todas as mensagens enviadas foram recebidas e armazenadas, e no segundo teste, foram contabilizadas 39 mensagens de cada um dos 100 dispositivos, sem nenhuma delas perdida. Com 600 dispositivos, foram observados problemas no registro dos dispositivos, mas, após reiniciar a aplicação, todos foram registrados conforme o esperado. Entretanto, no teste com 1000 dispositivos, ocorreu um problema durante o registro. A *ChirpStack*, baseada em *Golang*, uma linguagem de programação que trabalha com programação paralela utilizando várias *threads*, iniciou *threads* para o tratamento de novas mensagens. Na hora de registrar os 1000 dispositivos, essas *threads* entraram em conflito ao acessar o documento de armazenamento interno para salvar os dados de autenticação, devido ao atraso causado pela quantidade absurda de mensagens, comprometendo a aplicação e

inviabilizando a recepção de novos dados.

Qtd *Device*	tempo em min	registro	qtd msg p/ *device*	% msg perdidas
1	5	sim	4	0
100	40	sim	39	0
600	30	sim	18000	0
1000	40	não	0	100

Tabela 5.1 Testes feitos com o simulador

5.3 Comparação com Outros Protocolos

A Fábrica Recife precisava obter dados de outras fontes em sua planta, utilizando protocolos distintos para cada fonte. Essa coleta de dados foi aproveitada neste trabalho como uma oportunidade para obter novos dados comparativos com o *LoRaWAN*. Utilizando a máquina de estados do *KNoT Virtual Thing*, foram desenvolvidas aplicações para coleta de dados utilizando três protocolos: *OPC (PLC)*, *ModBus* e *HTTP (API)*. O gráfico da Figura 5.3 mostra a quantidade de sensores por protocolo. No contexto do *LoRaWAN* foram contabilizados 97 sensores, sendo que cada medidor de qualidade da rede elétrica envia 17 dados diferentes, e em todos os protocolos, cada leitura é considerada um sensor. Além dos outros medidores de vibração e temperatura *LoRaWAN* não abordados nesse trabalho.

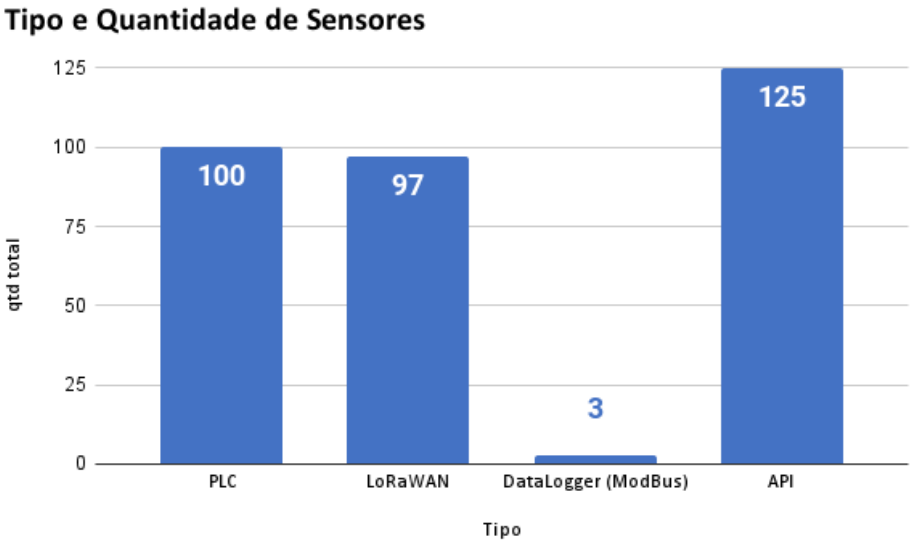

Figura 5.3 Tipo e quantidade de sensores.

Dentre os sensores, existiam diferentes intervalos de envio de dados, mesmo em um único meio coletado, como a *API* na Figura 5.4, que possui 3 conjuntos de sensores com intervalos de envio.

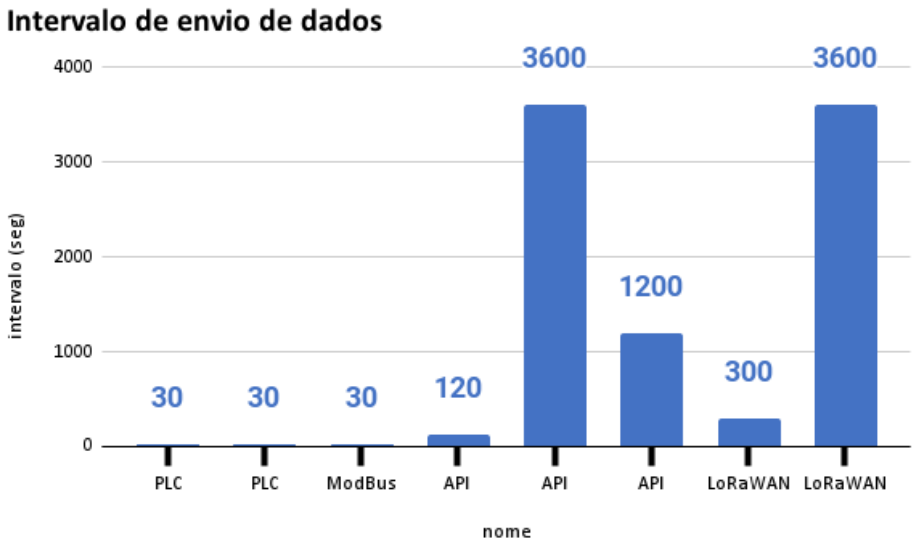

Figura 5.4 Intervalo de envio de dados.

O menor intervalo de envio é de 120 segundos, conforme mostrado na Figura 5.5, que agrupa os meios coletados e informa apenas o menor intervalo de envio para evidenciar um possível congestionamento na recepção de dados pela *KNoT Cloud*.

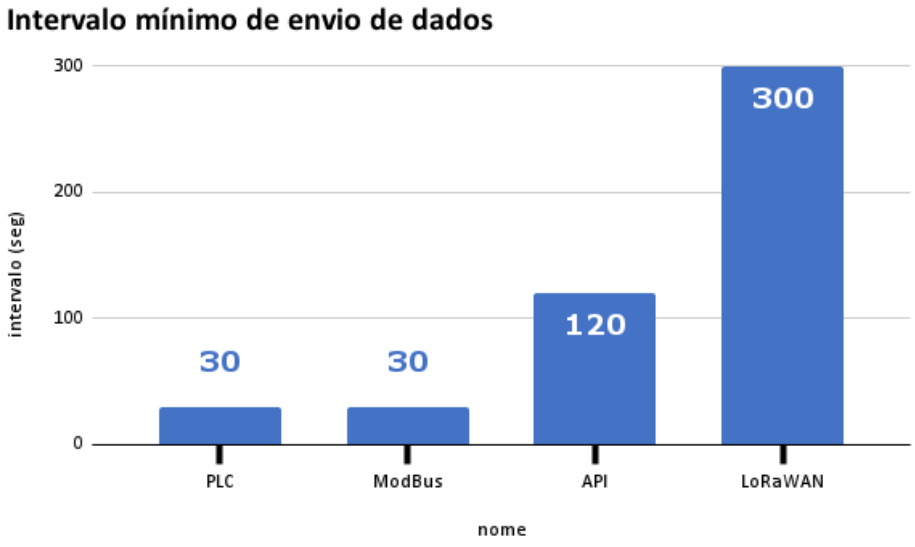

Figura 5.5 Intervalo mínimo de envio de dados.

O gráfico da Figura 5.6 revela a taxa de mensagens por minuto, sendo a maior de cada meio coletado mostrada na Figura 5.7. Durante os testes, mesmo o *LoRaWAN* tendo um número de sensores semelhante ao do *PLC* e da *API*, o *PLC* foi responsável pela maior

taxa de *throughput* de dados. Mesmo assim, a *KNoT Cloud* conseguiu suportar todos os dados recebidos e armazená-los sem problemas.

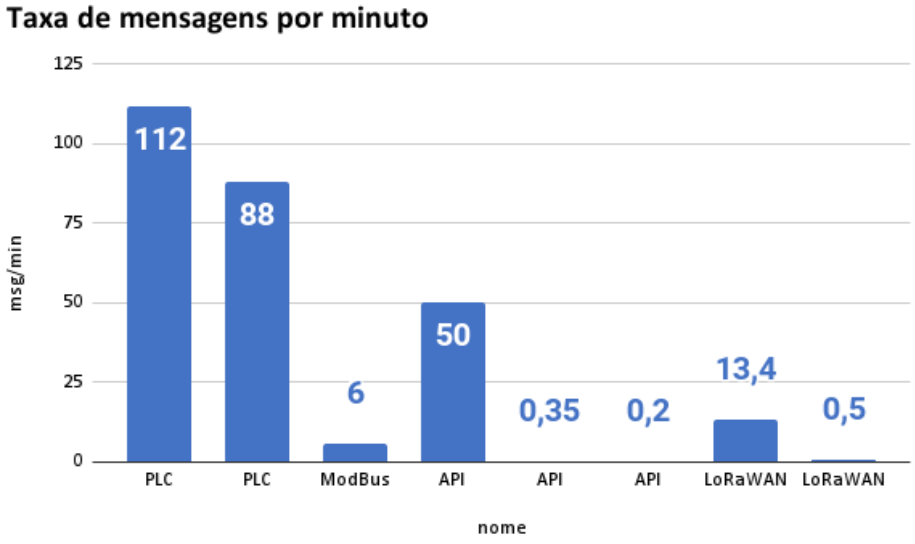

Figura 5.6 Taxa de mensagens por minuto.

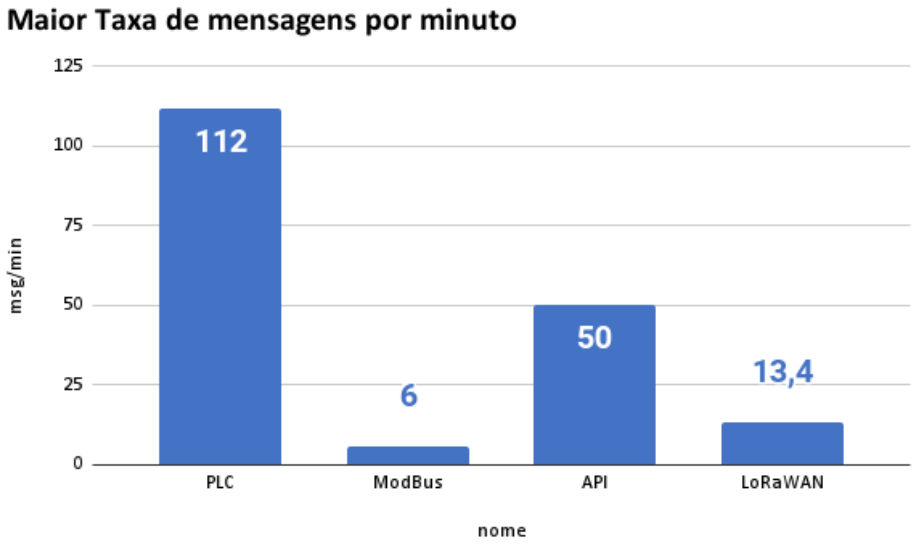

Figura 5.7 Maior taxa de mensagens por minuto.

6 Conclusão e Trabalhos Futuros

A meta-plataforma *KNoT IoT* atende ao seu propósito e oferece diversas ferramentas para lidar com os desafios de ter vários protocolos enviando dados simultaneamente. Além disso, ela permite a integração de novos protocolos de forma simples pelos desenvolvedores. Da mesma forma, a *ChirpStack* demonstra eficiência ao lidar com a rede *LoRaWAN* e compartilhar dados com diversas integrações implementadas e futuras. A combinação dessas duas plataformas traz benefícios significativos para a coleta e tratamento de dados em uma planta industrial, tornando-as mais competitivas e abrangentes.

A integração das duas plataformas se mostrou possível e bem-sucedida na captura de dados em grandes distâncias em uma planta industrial, com 100 dispositivos conectados usando uma configuração simples tanto do dispositivo *LoRaWAN* quanto do *KNoT* na aplicação da *ChirpStack*. Apesar de enfrentar desafios no manuseio de 1000 dispositivos, a solução é promissora, visto que com 600 dispositivos simulados, houve um bom desempenho, com apenas um pequeno problema no momento do cadastro, mas a recepção dos dados ocorreu com sucesso.

A análise abrangente dos protocolos de coleta de dados destacou a eficiência do *LoRaWAN* como um meio confiável e escalável para a transmissão de dados em ambientes industriais. Apesar de lidar com uma quantidade semelhante de sensores em comparação com outros protocolos, como o PLC e a *API*, o *LoRaWAN* demonstrou ser uma opção viável, oferecendo uma taxa de mensagens por minuto consistente e confiável. Sua capacidade de transmitir dados em distâncias consideráveis, juntamente com a capacidade da *KNoT Cloud* de gerenciar e armazenar os dados recebidos, ressalta a robustez e a eficácia do *LoRaWAN* como parte integrante de soluções de *IoT* em ambientes industriais.

Dessa forma, a integração funciona e pode ser aplicada em plantas industriais com até 600 sensores conectados, mas precisa de ajustes e desenvolvimentos para aplicações de maior porte. Os dados foram captados e transmitidos para o *KNoT Cloud* sem problemas na integração nos testes com dispositivos reais.

6.1 Trabalhos Futuros

Durante o desenvolvimento deste trabalho, observou-se que essa integração seria a forma mais rápida de conectar o protocolo *LoRaWAN* à meta-plataforma *KNoT Cloud*. No entanto, para trabalhos futuros, é possível explorar outras abordagens, como incorporar o código da *ChirpStack* diretamente em um dos serviços da *KNoT Cloud*. Isso permitiria

um controle mais granular sobre a rede e possibilitaria uma configuração mais simplificada para o usuário final. Outra alternativa seria adicionar telas na interface web da *ChirpStack* para a edição dos *KNoT Things*, facilitando a configuração e gerenciamento da integração.

Além disso, em futuros desenvolvimentos, seria interessante atualizar essa integração para a *ChirpStack V4*, aproveitando as ferramentas e melhorias mais recentes. Isso proporcionaria uma maior abrangência na coleta de dados, garantindo compatibilidade com as versões mais recentes das plataformas envolvidas. Poderia ser considerada também a proposta de um *Pull Request* para o repositório original da *ChirpStack*, contribuindo para a comunidade e possibilitando uma maior disseminação da integração.

Referências Bibliográficas

Introduction — knot documentation. https://knot-devel.cesar.org.br/doc/general/introduction.html, 2019. Acesso em: 6 nov. 2023.

Alberto Luiz Albertin and Rosa Maria de Moura Albertin. A internet das coisas irá muito além das coisas. 2017. Disponível em <https://repositorio.fgv.br/server/api/core/bitstreams/aef53964-fff1-42d4-b7b7-d6d7e62f6829/content> Acesso: 9, 1 de 2024.

LoRa Alliance. Lorawan 1.0.4 specification package. https://resources.lora-alliance.org/technical-specifications/ts001-1-0-4-lorawan-l2-1-0-4-specification, 2020. Acesso em 22 ago. 2023.

Aloÿs Augustin, Jiazi Yi, Thomas Clausen, and William Mark Townsley. A study of lora: Long range & low power networks for the internet of things. *Sensors*, 16(9), 2016. ISSN 1424-8220. URL https://www.mdpi.com/1424-8220/16/9/1466.

César Perdigão Batista, Pedro Victor Silva, Thais Batista, and Everton Cavalcante. Integrando as plataformas fiware e knot para o desenvolvimento de aplicações de internet das coisas. In *Anais do X Simposio Brasileiro de Computacao Ubiqua e Pervasiva*. SBC, 2018.

Brocaar. Chirpstack simulator repository. https://github.com/brocaar/chirpstack-simulator. Acesso em 21 ago. 2023.

CESAR. Knot documentation - unit type value. https://knot-devel.cesar.org.br/doc/thing/unit-type-value.html. Acesso em 21 ago. 2023.

CESAR. Knot - connectivity platform for iot. https://knot.cesar.org.br/, 2021. Acesso em 21 ago. 2023.

CESAR. Knot virtualthing repository. https://github.com/CESARBR/knot-virtualthing, 2024. Acesso em 21 ago. 2023.

ChirpStack. Open-source lorawan network server stack. https://www.chirpstack.io, 2021. Acesso em 21 ago. 2023.

Danilo Alfredo Marinho de Souza. Estudo de lacunas da meta-plataforma knot para iot. 2017. Disponível em https://www.cin.ufpe.br/~tg/2017-1/dams_tg.pdf, acessado em 9 de janeiro de 2024.

Agência Nacional de Telecomunicações (ANATEL). Resolução 680 da anatel. https://informacoes.anatel.gov.br/legislacao/resolucoes/2017/936-resolucao-680, 2017. Acesso em 21 ago. 2023.

Shilpa Devalal and A. Karthikeyan. Lora technology - an overview. In *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 284–290, 2018. doi: 10.1109/ICECA.2018.8474715.

LTD. Dragino Technology Co. Lps8v2 – indoor lorawan gateway. https://www.dragino.com/products/lora-lorawan-gateway/item/228-lps8v2.html. Acessado em 6 de março de 2024.

Jetmir Haxhibeqiri, Eli De Poorter, Ingrid Moerman, and Jeroen Hoebeke. A survey of lorawan for iot: From technology to application. *Sensors*, 18(11):3995, 2018. ISSN 1424-8220. doi: 10.3390/s18113995. URL https://www.mdpi.com/1424-8220/18/11/3995.

B HINZ. Lorawan® leads the global industrial evolution and drives industry 5.0's sustainability. 2023. Disponível em https://lora-alliance.org/lora-alliance-press-release/lorawan-leads-the-global-industrial-evolution-and-drives-industry-5-0s-sustainabili

Khomp. Endpoint lora: Medidor de energia com tc e bluetooth. https://www.khomp.com/pt/produto/endpoint-lora-medidor-de-energia-com-tc-e-bluetooth/. Acesso em 21 ago. 2023.

khomp. Medidor de energia lora® com tcs. https://www.khomp.com/pt/produto/endpoint-lora-medidor-de-energia-com-tc-e-bluetooth/. Acessado em 6 de março de 2024.

Fiona Kuan. Qual é a tecnologia por trás da frequência lora. https://www.mokosmart.com/pt/lora-frequency/, 2019. Acesso em: 3 mar. 2024.

Alexandru Lavric and Adrian Ioan Petrariu. Lorawan communication protocol: The new era of iot. In *2018 International Conference on Development and Application Systems (DAS)*, pages 74–77, 2018. doi: 10.1109/DAAS.2018.8396074.

E. Manavalan and K. Jayakrishna. A review of internet of things (iot) embedded sustainable supply chain for industry 4.0 requirements. *Computers & Industrial Engineering*, 127: 925–953, 2019. ISSN 0360-8352. doi: https://doi.org/10.1016/j.cie.2018.11.030. URL https://www.sciencedirect.com/science/article/pii/S0360835218305709.

Oleksiy Mazhelis and Pasi Tyrväinen. A framework for evaluating internet-of-things platforms: Application provider viewpoint. In *2014 IEEE World Forum on Internet of Things (WF-IoT)*, pages 147–152, 2014. doi: 10.1109/WF-IoT.2014.6803137.

Joao Neto, Lucas Feijó, Jorge Fonseca, and Fernando Carvalho. O uso do knot como uma plataforma tradicional de iot. *Semana Universitária (SUPER 2017) - UPE Caruaru 49*, 2017.

Marcelo T Okano. Iot and industry 4.0: The industrial new revolution. In *International Conference on Management and Information Systems*, volume 25, page 26, 2017.

Lara Carolina Oliveira and Flávio Silva. Análise comparativa de plataformas baseadas em cloud para o desenvolvimento de aplicações iot. In *Anais Estendidos do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 257–264, Porto Alegre, RS, Brasil, 2020. SBC. doi: 10.5753/sbrc_estendido.2020.12427. URL `https://sol.sbc.org.br/index.php/sbrc_estendido/article/view/12427`.

Keyur K Patel, Sunil M Patel, and P Scholar. Internet of things-iot: Definition, characteristics, architecture, enabling technologies, application & future challenges. *International Journal of Engineering Science and Computing*, 6(5), 2016.

Paulo F Pires, Flavia Delicato, Thais Batista, Thomaz Barros, Everton Cavalcante, and Marcelo Pitanga. Plataformas para a internet das coisas. *Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, 2015.

RAK. Rak7240 quick start guide. `https://docs.rakwireless.com/Product-Categories/WisGate/RAK7240/Quickstart/#product-configuration`. Acessado em 6 de março de 2024.

Ankita Raychowdhury and Ankita Pramanik. Survey on lora technology: Solution for internet of things. In Sabu M. Thampi, Ljiljana Trajkovic, Sushmita Mitra, P. Nagabhushan, El-Sayed M. El-Alfy, Zoran Bojkovic, and Deepak Mishra, editors, *Intelligent Systems, Technologies and Applications*, pages 259–271, Singapore, 2020. Springer Singapore.

Julio Ross. *Redes de computadores*. Julio Ross, 2008.

semtech. What are lora® and lorawan®? `https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/`, 2019. Acesso em: 3 mar. 2024.

Hafiz Husnain Raza Sherazi, Luigi Alfredo Grieco, Muhammad Ali Imran, and Gennaro Boggia. Energy-efficient lorawan for industry 4.0 applications. *IEEE Transactions on Industrial Informatics*, 17(2):891–902, 2021. doi: 10.1109/TII.2020.2984549.

thethingsindustries. The things industries launches a global lorawan® network peering. `https://thethingsindustries.pr.co/185854-the-things-industries-launches-a-global-lorawan-peering`, 2020. Acessado em 6 de março de 2024.

Yan Wang and Zhengwei Du. A printed dual-antenna system operating in the gsm1800/gsm1900/umts/lte2300/lte2500/2.4-ghz wlan bands for mobile terminals. *IEEE Antennas and Wireless Propagation Letters*, 13:233–236, 2014. doi: 10.1109/LAWP.2014.2303574.