Capítulo 1

Fundamentos de LoRaWAN – Teoria e Prática

A. F. Pastório, J. Rossato, J. P. C. A. Sá, F. A. Spanhol, L. A. Rodrigues, E. T. Camargo

Abstract

Low Power Wide Area Network (LPWAN) is a category of wireless network widely used to provide connectivity to the Internet of Things paradigm and have been widely used to support the construction of Smart Cities. Generally, an LPWAN network allows to periodically send only a few kilobits per second (Kbps), but provides longer battery life for end devices (sensors, actuators) while being able to cover areas of up to 50 kilometers in open field. Among the LPWAN technologies and protocols, the open standard LoRaWAN stands out. LoRaWAN operates on the proprietary wireless technology LoRa (Long Range) and occupies an unlicensed frequency band. Among the important features of the LoRaWAN standard are not charging any fee for network usage and encouraging the customer to install their own network infrastructure. This short course covers the fundamentals of LoRaWAN networks, presenting theoretical aspects illustrated with practical labs. The practical part includes how to perform communication between LoRa devices and communication with a network server in order to obtain the monitored data.

Resumo

As redes de longo alcance e baixa potência, conhecidas como LPWAN, são uma categoria de rede sem fio largamente utilizada para fornecer conectividade para o paradigma de Internet das Coisas e vem sendo amplamente utilizada para apoiar a construção de Cidades Inteligentes. Geralmente, uma rede LPWAN permite enviar periodicamente apenas poucos quilobits por segundo (Kbps), mas proporciona maior vida útil para as baterias dos dispositivos finais (sensores, atuadores) ao mesmo tempo em que pode cobrir áreas de até 50 quilômetros em campo aberto. Entre as tecnologias e protocolos de LPWAN, destaca-se o padrão aberto LoRaWAN, que opera sobre a tecnologia sem fio proprietária LoRa (Long Range) e ocupa uma faixa de frequência não licenciada. Entre as características importantes do padrão LoRaWAN estão não cobrar qualquer taxa pelo uso da rede e estimular o próprio cliente a instalar sua infraestrutura de rede. Este

minicurso aborda os fundamentos das redes LoRaWAN, apresentando aspectos teóricos ilustrados com laboratórios práticos. A parte prática inclui como realizar a comunicação entre dispositivos LoRa e a comunicação com um servidor de rede a fim de obter os dados monitorados.

1.1. Introdução

Desde a sua origem as redes de computadores sofrem constantes evoluções. A inserção de capacidades computacionais e de comunicação com objetos comuns do cotidiano, possibilita a dispositivos interagirem entre si, fornecerem informações e serem controlados remotamente através de serviços hospedados em servidores na Internet. Assim, a "Internet das Coisas", ou *Internet of Things* (IoT), é a integração de pessoas, processos e tecnologia com dispositivos e sensores conectados para permitir o monitoramento remoto, a manipulação e verificação do seu estado e avaliar as tendências de tais dispositivos [10]. Tal demanda exige das redes de computadores requisitos diferentes daqueles empregados para usuários que navegam na Internet, por exemplo, e faz com que o sucesso da sua evolução seja crucial para o avanço do mercado de IoT.

O mercado global de soluções para usuários finais da Internet das Coisas ultrapassou 100 bilhões de dólares em receita de mercado pela primeira vez em 2017 [4]. Há previsão que esse valor crescerá para cerca de 1,6 trilhão de dólares em 2025 [37], chegando a com 41 bilhões de dispositivos em 2027 [42]. Este número deve crescer a medida que a conectividade com a Internet se torna uma característica padrão para uma grande variedade de dispositivos eletrônicos e se deve ao fato de que IoT pode ser aplicado em diversos contextos:

- Cuidados em saúde (*Health Care*): IoT no campo da medicina é destinado a manter as pessoas seguras e saudáveis através de monitoramento em tempo real das funções vitais [20, 34];
- Casa Inteligente (*Smart Home*): em um ambiente de casa inteligente, iluminação, eletrodomésticos, computadores, câmeras de segurança e outros estão conectadas à Internet para permitir que o usuário as monitore e as controle independentemente da restrição de tempo e local [21];
- Rastreamento (*Tracking*): a localização tem muita importância no ambiente de vida conectado, como segurança, vigilância e movimentação de produtos e frotas. A localização e o rastreamento baseados em IoT são considerados mais abrangentes e precisos do que as técnicas utilizadas anteriormente [32, 5];
- Indústria Inteligente ou Indústria 4.0: com a introdução da IoT no ambiente de produção e manufatura, itens físicos como sensores, dispositivos e ativos corporativos são conectados à Internet. As informações geradas servem para a tomada de decisões relevantes no processo industrial e na qualidade do produto final [41];
- Agronegócio: IoT colabora no combate ao desperdício de insumos e aumentando a produção. A utilização das informações geradas por sensores colabora na utilização de recursos importantes para o aumento da produção. Utilizar recusos fundamentais

coma a água, por exemplo, é importante independente da cultura cutivada como agricultura, suinocultura, pecuária e outras [9] e;

- Cidades inteligentes (*Smart Cities*): A melhoria da qualidade de vida nas cidades modernas é um problema diário com que os cidadãos e a administração pública precisam lidar. A utilização de dispositivos inteligentes baseados em IoT gera informações para monitorar a qualidade da água e do ar com alerta para aumeno de poluentes, coleta de informações do consumo de água e energia elétrica das residências e órgão públicos, controle de tráfego urbano, mobilidade urbana e outros [33, 5].

Dependendo da aplicação o uso de energia da rede elétrica e a conexão com a Internet através de cabo *ethernet* são inviáveis. Assim a utilização de comunicação sem fio de baixo consumo energético nos objetos inteligentes é uma necessidade e um grande desafio, pois o fator energia é um dos principais limitantes em IoT e na maioria dos casos, o rádio utilizado para comunicação é um dos principais consumidores. Estimativas indicam que o custo energético da comunicação chega a 60% do gasto energético do dispositivo [26].

A Figura 1.1 apresenta algumas tecnologias de rede de comunicação basedas em ondas de rádio (sem fio) e as suas coberturas alcançadas. No contexto IoT destacam-se as redes sem fio de baixo consumo de energia, baixa largura de banda e alcance variando de alguns centímetros até quilômetros. Considerando o alcance, as redes *Wireless Personal Area Network* (WPAN) foram projetadas para interconectar dispositivos eletrônicos dentro do espaço de uma pessoa, sendo exemplos BLE e ZigBee. *Bluetooth Low Energy* (BLE) transmite na frequência de 2,4 GHz, com 1 Mbps de taxa máxima de transmissão e alcance máximo de transmissão de 50 metros. ZigBee por sua vez opera nas frequências de 2,4 GHz, 868 MHz e 915 MHz com uma taxa máxima de 250 kbps e com um alcande máximo de 100 metros [8]. Redes WPAN podem ter cobertura próxima a proporcionada por redes de área local sem fio, ou *Wireless Local Area Network* (WLAN). Contudo uma WLAN oferece larguras de banda significativamente maiores e maior consumo energético. Já redes projetadas para conectar dispositivos a longas distâncias são chamadas *Wireless Wide Area Network* (WWAN) e quando enviam uma pequena quantidade de dados por longas distâncias, mantendo baixo consumo energético, são conhecidas como LPWAN (*Low Power Wide Area Network*) [6].

Dentre as tecnologias LPWAN mais utilizadas atualmente destacam-se: NB-IoT [11], Sigfox [35] e LoraWAN [15]. NB-IoT (*Narrowband Internet of Things*) é uma tecnologia de rádio celular especificada pela 3GPP para IoT, atua na faixa de 700 MHz com uma taxa de dados de 170 kbps (*downlink*) e 250 kbps (*uplink*), alcançando aproximadamente 35 km. Os serviços de infraestrutura de rede são fornecidos por operadoras com a cobrança de taxa pela utilização. Sigfox é uma tecnologia proprietária da empresa francesa de mesmo nome. Utiliza a faixa de 900 MHz com a largura de canal de 1 kHz, com um alcance de 3 km a 50 km e uma taxa de dados de 100 bps transportando até 12 bytes de *uplink*. A Tabela 1.1 compara as tecnologias de comunicação sem fio usadas em IoT. É possível perceber que todas as redes LPWAN possuem uma topologia estrela e consequentemente requerem um ponto de acesso (ou *gateway*), responsável por receber os

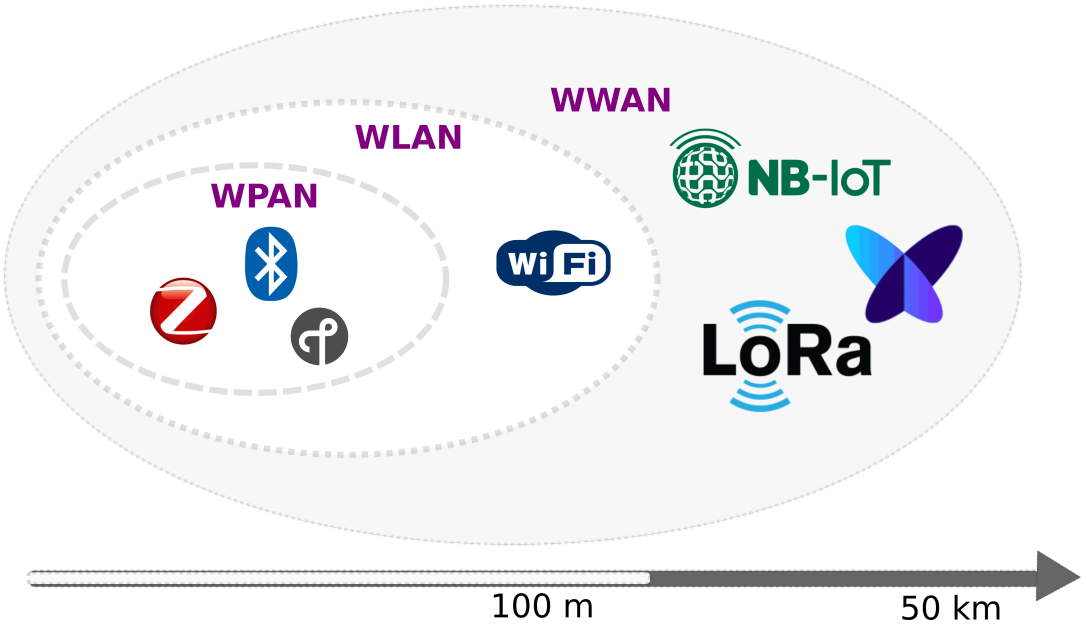

Figura 1.1: Cobertura das tecnologias de rede sem fio.

dados enviados dos dispositivos finais dentro da sua área de cobertura. Finalmente, LoRaWAN é um protocolo aberto de comunicação para a rede que funciona sobre a camada física *Long Range* (LoRa). Alcança uma distância entre 2 km até 40 km na faixa de 915 MHz (faixa destinada para o Brasil e homologada pela Anatel) e uma taxa de dados de 0,3 até 50 kbps. Uma das maiores vantagens dessa tecnologia é não ser necessário contratar uma operadora, possibilitanto desenvolver diversas aplicações em cenários IoT de forma independente.

Tabela 1.1: Comparação entre as redes sem fio.

Protocolo	**Alcance**	**Frequência**	**Taxa**	**Topologia**
Wi-Fi	50 m	2,4/5 Ghz	1.300 Mbps	Estrela
BLE	80 m	2,4 Ghz	1 Mbps	Estrela/Malha
ZigBee	100 m	868/915 MHz/2,4 GHz	250 kbps	Estrela/Malha
NB-IoT	35 km	700 MHz/1,8/2,1/2,5 GHz	170-250 kbps	Estrela
SigFox	3-50 km	868/902 MHz	10-100 bps	Estrela
LoRaWAN	2-40 km	915 MHz (Brasil)	0,3-50 kbps	Estrela

Conforme apresentado em [25], a conectividade LoRa/LoRaWAN vem servindo de base para aplicações no contexto de cidades intelientes, *smart grids*, agronegócio, saúde, localização, indústrias, entre outros. O mesmo trabalho apresenta que é crescente a quantidade de trabalhos científicos e práticos publicados envolvendo a tecnologia ao redor do globo. Como observado pelos autores os resultados de alcance da tecnologia LoRaWAN são influenciados pela topologia, densidade urbana, tipo de hardware, antena, mode de gerenciamento de pacotes, entre outros.

1.2. LoRa

LoRa é uma tecnologia de transmissão sem fio desenvolvida e patenteada pela empresa Semtech. LoRa realiza a modulação de radiofrequência (RF) baseada no espalhamento de espectral de chirp (*Chirp Spread Spectrum - CSS*). Essa técnica é utilizada na comunicação militar e espacial permitindo cobrir longas distâncias de comunicação e alcançar robustez à interferência [14]. Uma mensagem LoRa pode ser de *uplink* (eviada pelo dispositivo final) ou *downlink* (recebida pelo dispositivo final). A estrutura da mensagem é semelhante em ambos os casos, no entanto apenas a mensagem de *uplink* adiciona um código de verificação (CRC) que garante a integridade da carga útil (*PHYPayload*) [15].

No Brasil, a ANATEL publicou em agosto de 2018 o Ato nº 6.506 que aprova os procedimentos para avaliação da conformidade de equipamentos de radiocomunicação de radiação restrita, permitindo a operação de dispositivos LoRa no território nacional. O padrão adotado foi o australiando, ele utiliza a faixa de 915 MHz e compreende a faixa de 902 MHz a 907,5 MHz e 915 MHz a 928 MHz [33]. O padrão australiano possui 72 canais para *uplink* e 8 para *downlink*. Os canais de *uplink* vão de 0 a 63 utilizam uma largura de banda de 125 kHz com uma taxa de codificação 4/5, com início em 915,2 MHz sendo incrementado linearmente em 200 kHz até 927,8 MHz, já os canais de 64 a 71 possuem uma largura de banda de 500 kHz a partir de 915,9 MHz e incrementando linearmente de 1,6 MHz a 927,1 MHz . Para *downlink* são utilizdos os canais de 0 a 7 com uma largura de banda de 500 kHz, começando em 923,3 MHz e incrementando linearmente em 600 kHz até 927,5 MHz.

Para utilizar a modulação LoRa em um dispositivo final é preciso configurar três parâmetros: (i) Largura de Banda (*Bandwidth*, BW): com um dos tem três valores determinados, 125 kHz, 250 kHz ou 500 kHz, sofrendo um deslocamento de até 20% que não influenciará na decodificação [15]. (ii) Fator de Espalhamento (*Spreading Factor*, SF): determina o número de chirps necessários para representar um símbolo (um ou mais bits de dados), apresentando 2^{SF} valores possíveis de 7–12, quanto maior o SF mais energia será usada por bit e maior será o alcance entre transmissor e receptor [15]. (iii) Taxa de Codificação (*Coding Rate*, CR): define o número de bits destinados para dados de redundância na mensagem, a fim de realizar a recuperação de erros, definidos com os valores valores do CR: 4/5 , 4/6 , 4/7 e 4/8, incrementando o CR aumenta a proteção, mas também o tempo do bit no ar [15]. A Tabela 1.2 apresenta um exemplo de configuração para maior alcence e outro com o tamanho da mensagem maior [15]. É possível utilizar uma calculadora de *airtime* LoRaWAN para realizar as configurações para os dispostivos finais[1].

1.3. LoRaWAN

LoRaWAN é um protocolo aberto de comunicação que funciona sobre a camada física LoRa. Ele é mantido pela associação de empresas LoRa Alliance. A Figura 1.2 mostra a estrutura típica de uma rede LoRaWAN [15]. Os dispositivos finais (*End Nodes*) são objetos que possuem sensores ou atuadores. O *gateway* conecta os dispositivos na rede LoRaWAN. O servidor de rede (*Network Server*) gerencia a comunicação dos dispositivos com o servidor de aplicação através da Internet. O servidor de aplicação (*Application Ser-*

[1] `https://avbentem.github.io/airtime-calculator/ttn/au915`

Tabela 1.2: Exemplo para configuração na frequência de 915 MHz.

Parâmetros	Maior Alcance	Maior Mensagem
Data Rate (DR)	0	4
Spreading Factor (SF)	SF10	SF8
Bandwidth (BW)	125 kHz	500 kHz
Coding Rate (CR)	4/5	4/5
Bit Rate (BR)	976 bps	12.500 bps
Payload máximo	11 bytes	242 bytes
Air Time	371 ms	175 ms

ver) exibe as informações vindas dos dispositivos para o usuário final [14]. Os elementos da arquitetura LoRaWAN são descritos a seguir.

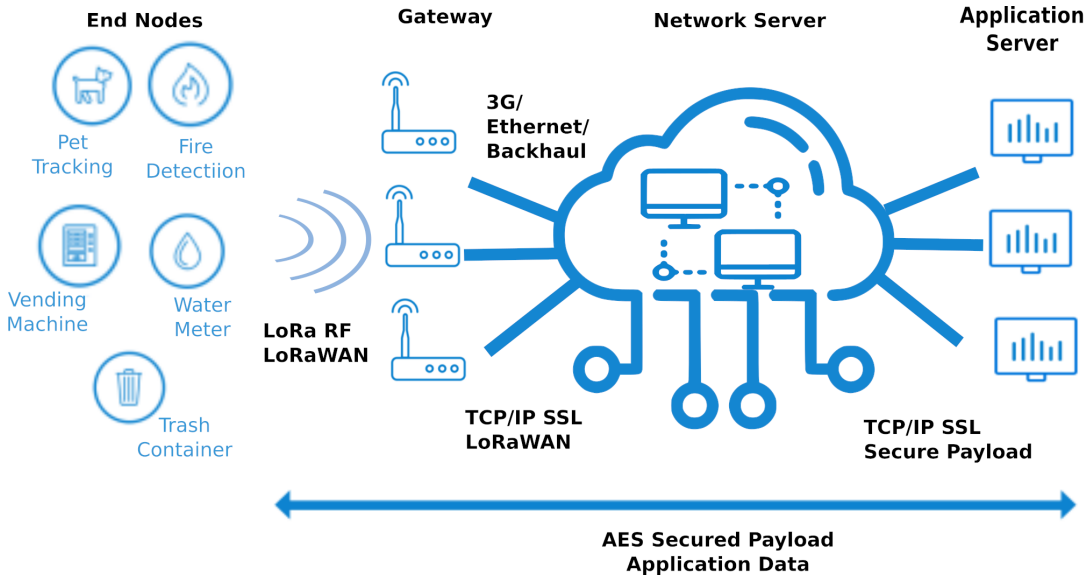

Figura 1.2: Visão geral da arquitetura LoRaWAN. Baseada em [14].

1.3.1. Dispositivos Finais

Os dispositivos finais coletam informações do ambiente, como temperatura, umidade, posição geográfica, batimentos cardíaco e outros, tanto em ambientes abertos como fechados. As informações são enviadas diretamente para um ou mais *gateways* da rede. Os dispositivos podem transmitir em um canal a qualquer momento, já para a próxima transmissão o canal deverá ser alterado respeitando a duração máxima de transmissão conforme as regulamentações locais.

Os *end nodes* podem assumir três configurações [15]. (i) Classe A: configuração obrigatória nas as demais classes, nela o dispositivo final inicia comunicação com o *gateway* a qualquer momento, realizando uma transmissão e abrindo duas janelas de recepção de dados do *gateway*. (ii) Classe B: a comunicação é iniciada pelo *gateway*, que determina o momento de transmissão do dispositivo final e na sequência abre duas janelas de recepção. (iii) Classe C: o dispositivo final realiza uma transmissão de dados, abre duas janelas de recepção e mantém uma aberta até a próxima transmissão, fazendo com

que o dispositivo final necessite de uma fonte constante de energia. LoRaWAN oferece criptografia através do padrão AES em dois níveis: o primeiro está localizado na camada de rede garantindo a autenticidade do dispositivo final e o segundo está na camada de aplicação e garante que a informação gerada no dispositivo somente será conhecida pelo servidor de aplicação na comunicação entre dispositivos finais e a aplicação. Há dois tipos de autenticação: *Over The Air Activation* (OTAA) e *Activation By Personalization* (ABP). A primeira é mais segura porque é o próprio dispositivo que gerencia sua ativação realizando um procedimento de conexão para ser ativado em uma rede selecionada. Já na ativação ABP o processo é mais simplificado pois os dispositivos finais estão diretamente vinculados a uma rede específica, ignorando o procedimento de entrada existente na OTAA [16]. A Figura 3 mostra os dois tipos de autenticação dos dispositivos finais.

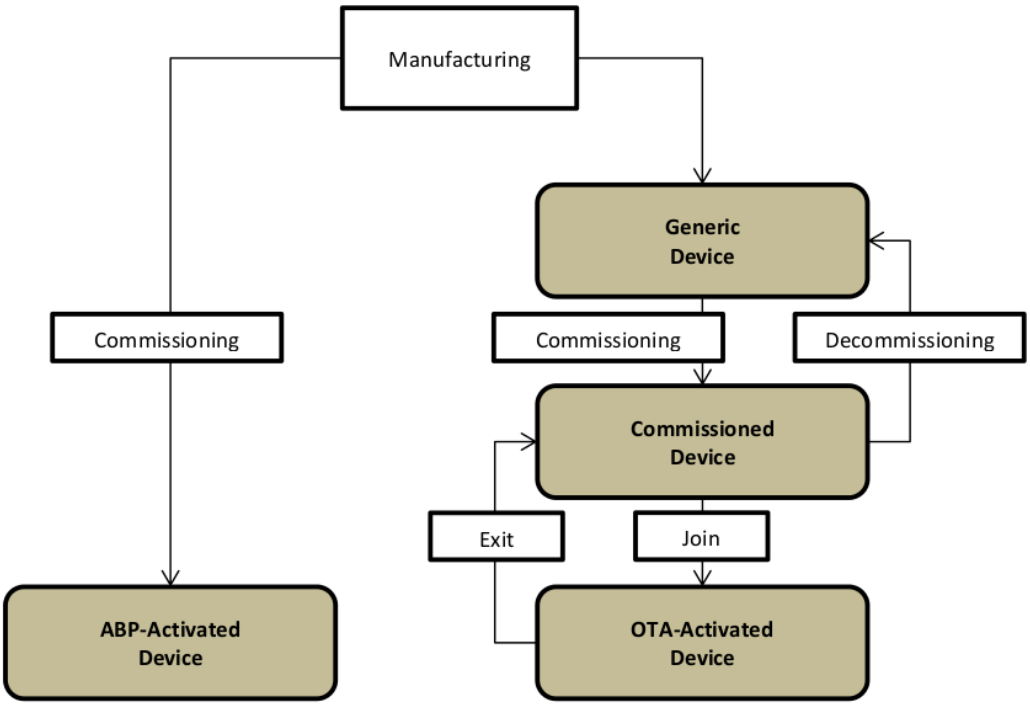

Figura 1.3: Autenticação por ABP e OTAA [16].

O desenvolvimento de dispositivos finais pode ser realizado através de placas de desenvolvimento que já incluem LoRaWAN [18] ou placas de prototipagem rápida como ESP32, Arduino, Raspberry Pi e outras, juntamente com módulos que possuem um ou vários sensores (temperatura, umidade, distância, GPS, etc.) e módulos transmissores [1]. Também é possível adquirir dispositivos finais prontos com sensores e atuadores de acordo com a finalidade do projeto [17]. No Brasil encontramos módulos LoRaWAN homologados pela Anatel fabricados por empresas nacionais ou importados. [31].

1.3.2. *Gateways*

O *gateway*, concentrador ou ponto de acesso, é responsável por estabelecer a conexão com vários dispositivos finais via RF usando a tecnologia de transmissão LoRa e o protocolo LoRaWAN, assim pode receber as informações dos dispositivos finais, organizá-la e

repassá-la via TCP/IP (rede *ethernet*, Wi-Fi, 3G/4G) para o servidor de rede LoRaWAN. Toda a comunicação na rede LoRaWAN é bidirecional, mas o *uplink* é o tráfego predominante e o *downlink* é utilizado quando é necessário enviar alguma informação para os dispositivos finais como parâmetros de configuração por exemplo [14].

Podemos encontrar *gateways* comerciais completos com um ou vários canais [19], ou apenas módulos concentradores para adicionar em placas de prototipagem como a Raspberry Pi e criar um *gateway* com maior liberdade para configurar e monitorar o tráfego [30].

1.3.3. Servidores de Rede

O servidor de rede é responsável pelo gerenciamento da rede LoRaWAN: recebe as informações dos *gateways*, elimina os pacotes duplicados e então encaminha-os para os servidores de aplicação. Também é responsável por descriptografar os pacotes de *uplink* validando ou descartando em caso de inconsistência. Além disso, envia para os dispositivos finais as configurações do *data rate* e a potência de transmissão. Para soluções de servidores de rede há várias opções, dentre elas duas destacam-se:

- ChirpStak: fornece componentes de código aberto para redes LoRaWAN. O servidor apresenta uma solução pronta para uso, incluindo uma interface Web para gerenciamento de dispositivos e APIs para integração. É necessário instalar a solução em um servidor local ou em nuvem [7];
- The Things Network (TTN) ou The Thinks Stack (TTS): fornece um conjunto de ferramentas abertas e uma infraestrutura de rede LoRaWAN pronta para uso. Realiza o cadastro de *gateways*, dispositivos finais e faz a integração com servidores de aplicação [40]

1.3.4. Servidores de Aplicação

O servidor de aplicação é responsável por receber os pacotes vindos do servidor de rede e apresentar os dados monitorados dos dispositivos finais. Nesse segmento os *payloads* são armazenados, descriptografados e os dados brutos são transformados em informação útil para o usuário que pode visualizar através de *softwares* específicos. Essa configuração permite que os servidores de rede e de aplicação sejam instalados na mesma máquina ou em equipamentos separados no mesmo local ou em pontos geograficamente distantes. Existem diversos servidores de aplicação com as mais diversas funcionalidades, destacando-se:

- TagoIO: plataforma Web totalmente *cloud* com ferramentas para gerenciar dispositivos finais, armazenar dados, executar análises e integrar serviços [13];
- AllThingsTalk: além de exibir as informações dos dispositivos finais, possui vários *Software Development Kit* (SDK) para plataformas de prototipagem como Arduino e para as linguagens de programação Java, Python e Go [2];
- Cayenne: servidor de aplicação desenvolvido pela empresa myDevice. Tem como destaque a integração com as placas de prototipagem Arduino e Raspberry Pi [23].

Todos esses servidores de aplicação possuem uma *Application Programming Interface* (API) para auxiliar no desenvolvimento de soluções para diversos cenários. É importante salientar que os servidores de aplicação oferecem também planos de uso gratuitos.

1.4. Atividades Práticas

Esta seção apresenta os experimentos práticos para ilustrar o processo de desenvolvimento de uma aplicação LoRaWAN. Inicialmente são apresentados os dispositivos necessários, assim como os requisitos de *software*. Na sequência são descritas as práticas utilizando a tecnologia LoRa. Os primeiros experimentos têm foco na camada física realizando a comunicação somente com LoRa de forma ponto a ponto (P2P). Por fim é realizado um experimento utilizando um *gateway* mais robusto conectando-se ao servidor de rede TTS e a um servidor de aplicação.

1.4.1. Dispositivos

O dispositivo final responsável por realizar a comunicação LoRa durante as práticas é um Heltec WiFi LoRa 32 V2, visto na Figura 1.4. Integra um microprocessador ESP32, contando assim com todas as funcionalidades típicas como WiFi, Bluetooth, ADC (*Analog to Converter*), DAC (*Digital to Analog Converter*) e etc., além do módulos integrado LoRa e um *display* OLED.

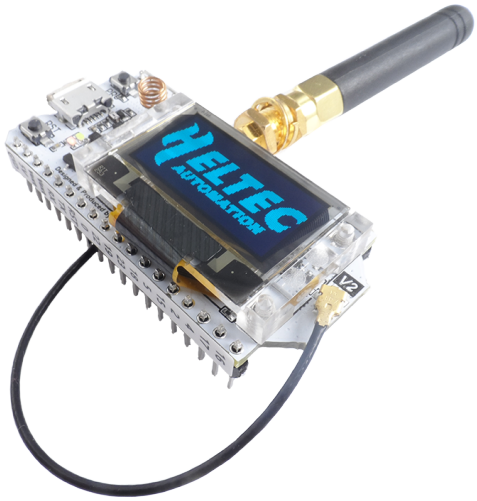

Figura 1.4: Heltec WiFi LoRa 32 V2 [12].

As especificações do Heltec são comparadas com outros dispositivos semelhantes e são apresentadas na Tabela 1.3. Devido ao ESP32, o Heltec é superior ao Arduino em processamento, mas ainda limitado em relação ao Raspberry Pi.

Outro dispositivo a ser utilizado é um *gateway* composto por um Raspberry Pi 3 B+ e um módulo HAT que é um *gateway* LoRaWAN da Radioenge, visto na Figura 1.5. O módulo possui GPS integrado, oferecendo geolocalização e sincronia de tempo com uma precisão de microsegundos.

Tabela 1.3: Comparação entre dispositivos.

	Arduino Uno R3	ESP32	Raspberry Pi 3 B+
CPU	ATmega328	Xtensa LX6 DualCore	BCM2837B0 QuadCore
Clock	20 MHz	240 Mhz	1.4 Ghz
Arquitetura	8 bit	32 bit	64 bit
RAM	2 KB SRAM	520 KB SRAM	1 GB SDRAM

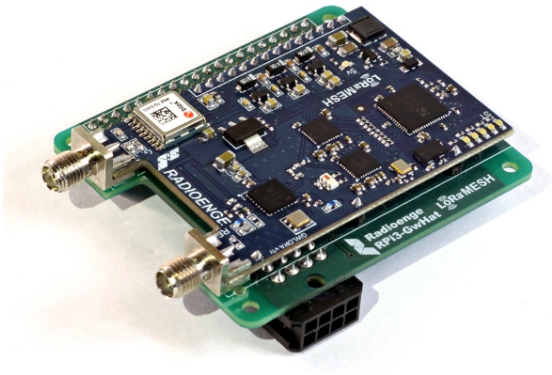

Figura 1.5: *Gateway* LoRaWAN Radioenge RD43HAT [30].

1.4.2. The Things Stack

O *gateway* conecta-se a rede através da *The Things Stack* (TTS). TTS é uma rede colaborativa que conecta *gateways* LoRa [39]. É uma plataforma gratuita e aberta que conta com várias funcionalidades e integrações a serem utilizadas por aplicações desenvolvidas.

1.4.3. Programação

A programação do dispositivo final pode ser feita em C, C++ e Python. Pode-se utilizar os *frameworks* Arduino ou Espressif ESP-IDF. A comunicação LoRa disponível para o dispositivo é fornecida por *firmware* e distribuído em forma de bibliotecas, LMIC[2] e arduino-LoRa. LMIC foi inicialmente criada pela IBM, que descontinuou o projeto. Sendo assumido pela comunidade disponibilizando o código de forma aberta no GitHub. A arquitetura de um *firmware* que utiliza LMIC pode ser vista na Figura 1.6. Sendo a camada do microcontrolador a camada de *hardware*, contendo o módulo de radiofrequência LoRa, sensores e atuadores. A camada de *software* utiliza o *framework* do Arduino, sendo composta pela lógica da aplicação, os *drivers* dos componentes e da biblioteca LMIC. A biblioteca LMIC é responsável por seguir os protocolos LoRaWAN e *Medium Access Control* (MAC), manter o ambiente de execução e fazer a abstração de *hardware* com relação aos transmissores. Já a biblioteca arduino-LoRa implementa a camada física da LoRaWAN, podendo ser utilizada de forma independente sendo a camada de controle MAC de responsabilidade do desenvolvedor.

[2]Disponível em `https://github.com/mcci-catena/arduino-lmic`

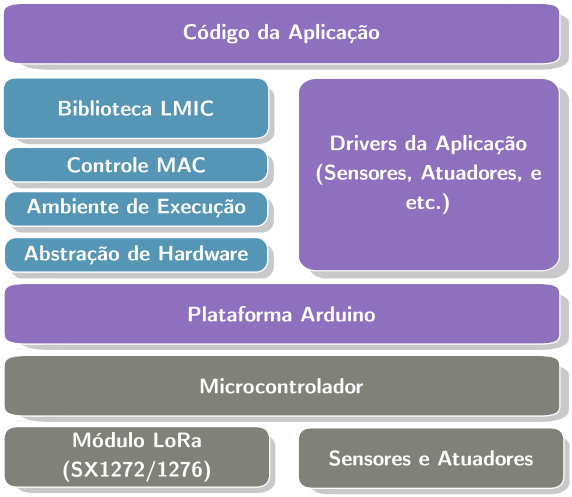

Figura 1.6: Arquitetura em camadas de uma aplicação LMIC. Baseada em [22].

1.4.3.1. IDE

A IDE (*Integrated Development Environment*), ambiente de desenvolvimento, a ser utilizado durante as práticas é o VSCode[3] junto da extensão PlatformIO. É um ambiente simples e que permite uma rápida adaptação e fornece uma estrutura de projeto organizada, possuindo funcionalidades de depuração e testes unitários.

A instalação também é bem simples. Uma vez tendo o VSCode instalado, basta acessar a aba de extensões, procurar por *PlatformIO* e clicar em *Install*. A Figura 1.7 apresenta o processo de instalação da extensão indicando com números os passos a serem realizados.

Figura 1.7: Intalação da extensão PlatformIO IDE.

Após intalação do PlatformIO será necessário reiniciar o VSCode. Na próxima inicialização a extensão estará habilitada como visto na Figura 1.8.

Para criar um novo projeto basta clicar no botão *Home* e então *New Project*, surgindo uma tela semelhante a Figura 1.9. Onde pode-se definir o nome do projeto, a placa de desenvolvimento e o *framework* a serem utilizados.

[3]Disponível em `https://code.visualstudio.com/`

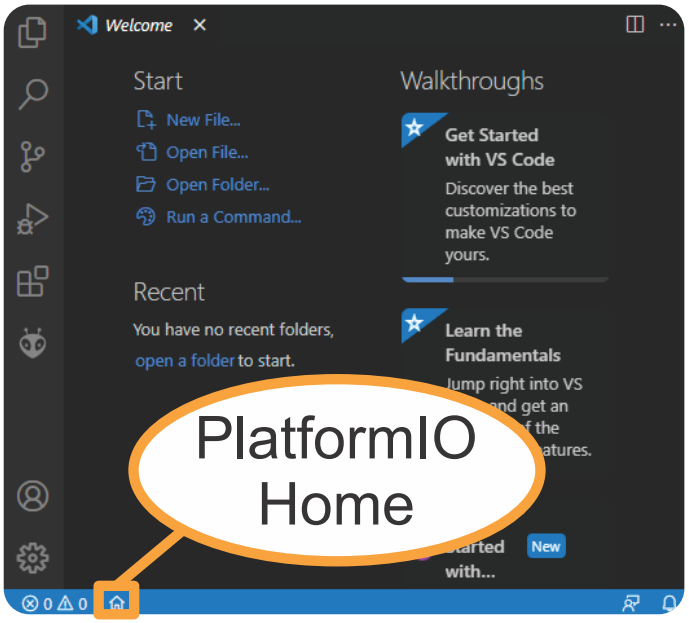

Figura 1.8: Inicialização do PlatformIO IDE.

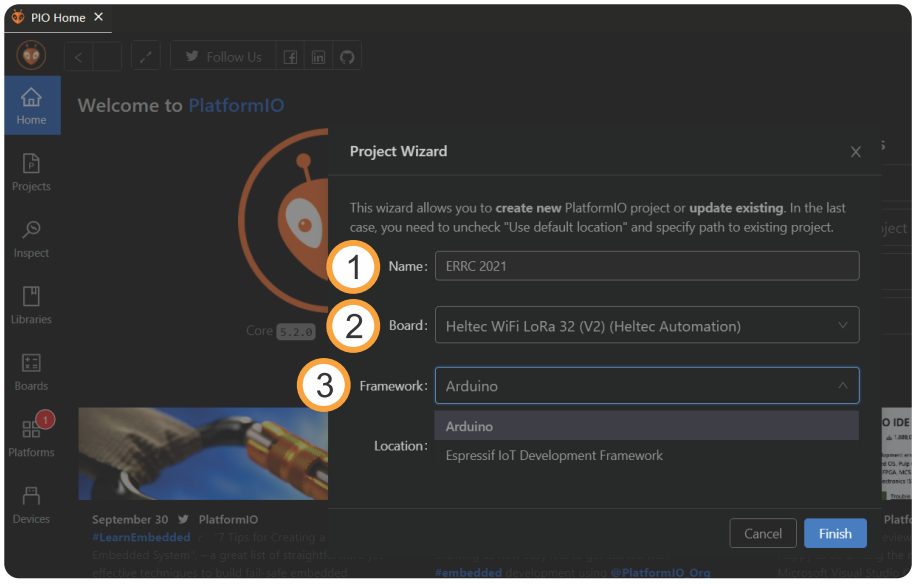

Figura 1.9: Painel de novo projeto do PlatformIO IDE.

Com o projeto criado podemos ver a estrutura do projeto e os botões importantes como na Figura 1.10. Onde em *lib* ficam localizadas as bibliotecas a serem utilizadas. Em *include* os arquivos de cabeçalho. Os arquivos principais se encontram no diretório *src*. Com relação aos botões, *Build* compila o *firmware* e monta o arquivo a ser gravado na placa. *Upload* faz a gravação de fato. E *Serial Monitor* disponibiliza uma interface serial com o microcontrolador.

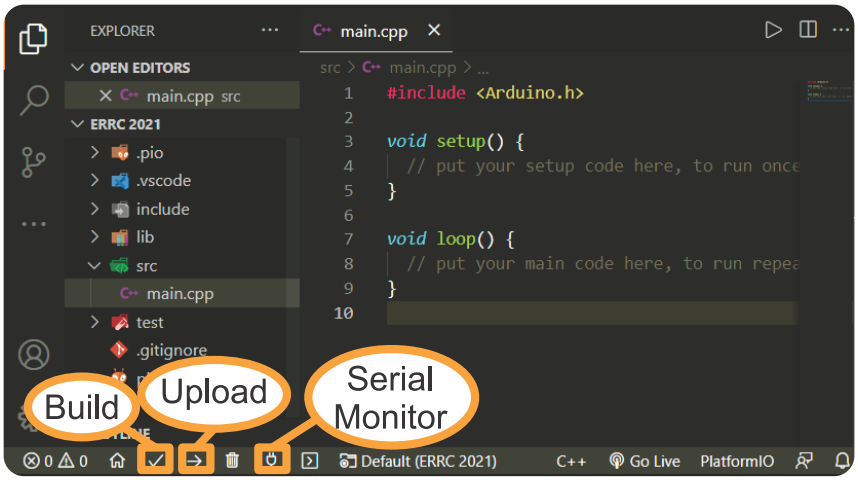

Figura 1.10: Botões importantes do PlatformIO IDE.

1.4.4. Prática 1: Comunicação LoRa Ponto a Ponto

Esta prática tem o objetivo de demonstrar o funcionamento da comunicação LoRa utilizando somente a camada fisica. Será utilizada a biblioteca `arduino-lora` que está contida na biblioteca do dispositivo, `Heltec_ESP32`[4] disponibilizada pela Heltec Automation. Para instalar a biblioteca basta acessar a aba de *Libraries* como na Figura 1.11 e pesquisar o termo "Heltec", ou instalar manualmente baixando a biblioteca e colocando-a na pasta `lib`.

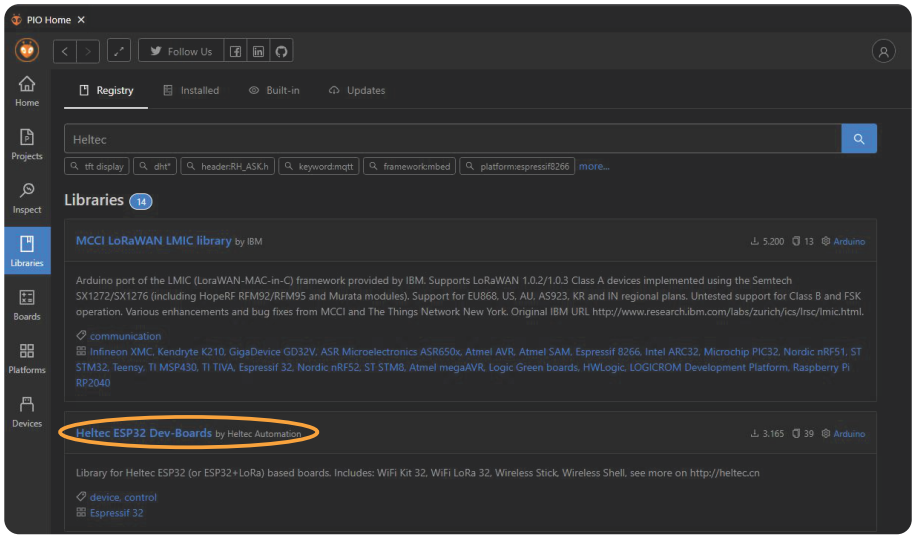

Figura 1.11: Biblioteca Heltec utilizada.

Agora basta executar o mesmo procedimento para instalar as bibliotecas *DHT sensor library* e *Unified Sensor* da Adafruit, bem como a biblioteca *SDS011 Sensor Library* de R. Zschiegner.

Esta prática, durante a apresentação, será dividida em três etapas. A primeira con-

[4]Disponível em: `https://github.com/HelTecAutomation/Heltec_ESP32`

siste de uma comunicação simples entre os dois dispositivos enviando um pacote LoRa com um contador e alguns dados relativos a intensidade do sinal. Na segunda etapa incluiremos o sensor de umidade e temperatura DHT11 e enviaremos os dados para o receptor. Por fim incluiremos o sensor SDS011, fabricado pela Nova Fitness. Os códigos completos estão disponibilizados em um repositório no GitHub[5].

1.4.4.1. Listagens de Código

Listagem 1 *Includes & Defines*

```
#include <Arduino.h>
#include "heltec.h"

#define BAND    915E6
String rssi = "RSSI --";
String packSize = "--";
String packet;
```

Após a importação das bibliotecas necessárias e a definição das variáveis globais, a função `setup` inicializa o dispositivo Heltec e faz as configurações iniciais do *display* e do rádio LoRa.

Listagem 2 *Setup* do receptor

```
void setup() {
   Heltec.begin(true, true, true, true, BAND);
   Heltec.display->init(); //Inicia o display
   Heltec.display->flipScreenVertically();
   Heltec.display->setFont(ArialMT_Plain_10);
   delay(1500);
   Heltec.display->clear();

   Heltec.display->drawString(0, 0, "LoRa iniciado com sucesso!");
   Heltec.display->drawString(0, 10, "Esperando dados...");
   Heltec.display->display();
   delay(1000);
   //LoRa.onReceive(cbk);
   LoRa.receive();
 }
```

Já a função `loop` aguarda a chegada do pacote LoRa. Uma vez recebido o pacote chama as funções responsáveis por formatar e apresentr os dados no *display*.

Com todas estas funções definidas, o *upload* para o dispositivo receptor pode ser feito, Caso o dispositivo seja ligado sem que exista um *sender*configurado e enviando pacotes o receptor ficará retido no *loop*, sem entrar na função `cbk`. Apenas o *display* do receptor ficará ligado, uma vez que o SDS011 utiliza a mesma entrada responsável pelo *reset* do OLED no *sender*

As bibliotecas inclusas no dispositivo responsável por enviar os dados diferem um pouco do *receiver*, será necessario utilizar as bibliotecas do DHT11 e do SDS011, assim como utilizar algumas variaveis globais e instanciar as devidas classes.

[5]Disponível em `https://github.com/afpastorio/ERRC-2021`

Listagem 3 Outras funções do receptor

```
void loop() {
   int packetSize = LoRa.parsePacket();
   if (packetSize)
      cbk(packetSize);
   delay(10);
}

void LoRaData(){
   Heltec.display->clear();
   Heltec.display->setTextAlignment(TEXT_ALIGN_LEFT);
   Heltec.display->setFont(ArialMT_Plain_10);
   Heltec.display->drawString(0 , 15 , "Received " + packSize + " bytes"); //apenas em 1.4.4.1
   Heltec.display->drawString(0, 0, rssi);
   Heltec.display->drawString(0, 15, packet);
   Heltec.display->display();
}

void cbk(int packetSize){
   packet = "";
   packSize = String(packetSize,DEC);
   for (int i = 0; i < packetSize; i++)
      packet += (char) LoRa.read();
   rssi = "RSSI " + String(LoRa.packetRssi(), DEC);
   LoRaData();
}
```

Listagem 4 *Includes & Defines* do *sender*

```
#include <Arduino.h>
#include "heltec.h"
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>
#include <SDS011.h>

#define BAND 915E6
#define DHTPIN 3
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);
SDS011 my_sds;
String rssi = "RSSI --";
String packSize = "--";
String packet ;
float p10, p25;
int err,ip10,ip25;

#ifdef ESP32
   HardwareSerial port(2);
#endif
```

A função `setup` é responsável por iniciar todos os objetos. Desta vez na função `Heltec.begin()` iremos utilizar o valor de *false*, já que o *display* não será iniciado. A função `loop` é responsável por montar e enviar o pacote LoRa com os dados da temperatura, umidade e de qualidade do ar (PM10 e PM2.5).

Após a definição de todas estas funções podemos fazer o *upload* do código para o dispositivo responsável por enviar os pacotes. O SDS011 demora cerca de um minuto para fazer a leitura dos dados de PM10 e PM2.5. Portanto, caso o receptor esteja ligado, dentro de aproximadamente um minuto o *display* será atualizado com as informações de

Listagem 5 *Setup & Loop* do *sender*

```
void setup(){
   Serial.begin(9600);
   Heltec.begin(false, true, true, true, BAND);
   dht.begin();
   my_sds.begin(&port);
   pinMode(25, OUTPUT);
}

void loop(){
   float h = dht.readHumidity();
   float t = dht.readTemperature();
   err = my_sds.read(&p25, &p10);
   if (!err){
      Serial.println("Ok");
      LoRa.beginPacket();
      LoRa.print("Umi: ");
      LoRa.print(h);
      LoRa.println("\%");
      LoRa.print("Temp: ");
      LoRa.print(t);
      LoRa.println("Graus celsius");
      LoRa.print("PM25: ");
      LoRa.println(p25);
      LoRa.print("PM10: ");
      LoRa.println(p10);
      LoRa.endPacket();
   } else
      Serial.println("Erro!");
 }
```

temperatura, umidade, PM10, PM2.5 e RSSI.

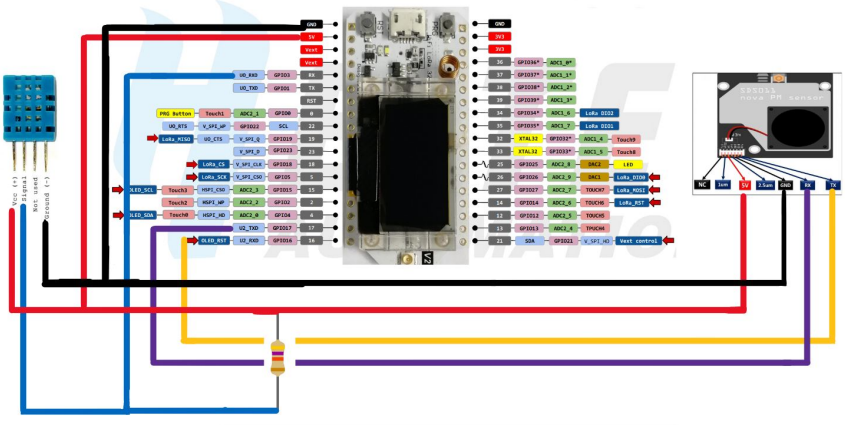

Figura 1.12: Diagrama esquemático da prática 1.

1.4.5. Prática 2: Comunicação LoRaWAN com Servidor de Rede

A comunicação com um servidor de rede TTS é possível utilizando um *gateway* de ao menos 4 canais. Uma alternativa de baixo custo era a programação de um *gateway* "caseiro" de um canal com algum dispositivo final atuando como *gateway*. Mas, recentemente a plataforma TTN recebeu uma atualização, passando para TTS. Esta nova plataforma é escala melhor e soluciona alguns problemas encontrados na sua versão anterior. Porém,

os *gateways* simples passaram a não funcionar mais.

Portanto, nesta prática será utilizado o *gateway* comercial homologado pela Anatel para demonstrar a comunicação com o servidor de rede TTS. O próprio fabricante disponibiliza um tutorial demonstrando como realizar a instalação do *firmware* do *gateway* e como registrá-lo na rede. Como a plataforma TTS sofreu mudanças recentemente somente o seu registro será abordado.

Acessando a TTS[6] é possível realizar o *login*, ou cadastro, na plataforma. Uma vez feita a autenticação o console é exibido apresentando as seções de aplicações e *gateways*. Para registrar um *gateway* entramos na área dos *gateways* e clicamos em *Add Gateway*. Neste momento três parâmetros devem ser configurados, *Gateway-ID*, *Gateway EUI* e *Frequency Plan*. *Gateway-ID* se refere a um identificador único que vai dar nome ao *gateway*. *Gateway EUI* também é um identificador único do dispositivo, mas este é fornecido pelo fabricante. Já *Frequency Plan* está relacionado ao plano de frequência a ser usado, no caso Australia 915-928 MHz, FSB2. O processo também pode ser visualizado no vídeo no YouTube[7].

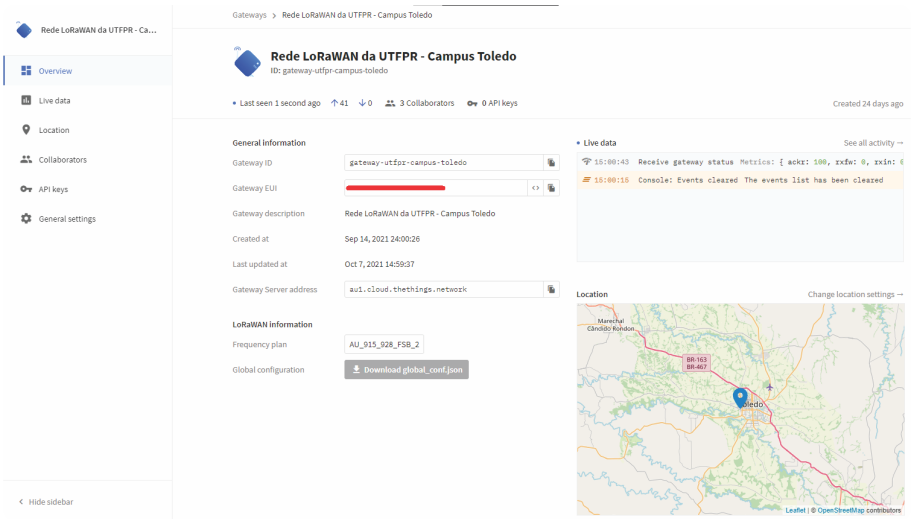

Figura 1.13: Console *gateway*.

Com o *gateway* configurado é momento de adicionar uma aplicação. Na seção *Applications* clique em *Add Application*. De forma bem simples preencha os campos como informado, aqui somente o *Application ID* é obrigatório.

No painel da aplicação precisamos adicionar um dispositivo final. Basta clicar em *Add End Device*. Mudamos para configuração manual e selecionamos a versão LoRaWAN e o plano de frequência. As chaves podem ser geradas automaticamente, destaque para AppEUI que deve ter pelo menos um número diferente de zero. Esse processo pode ser visto na Figura 1.14.

Para a programação do dispositivo será utilizada a última versão da LMIC. Podendo ser instalada tanto pelo gerenciador de bibliotecas quanto manualmente. O código

[6]Disponível em `https://au1.cloud.thethings.network`
[7]Disponível em `https://youtu.be/vD2TaFjzCWI?t=117` a partir de 1:57

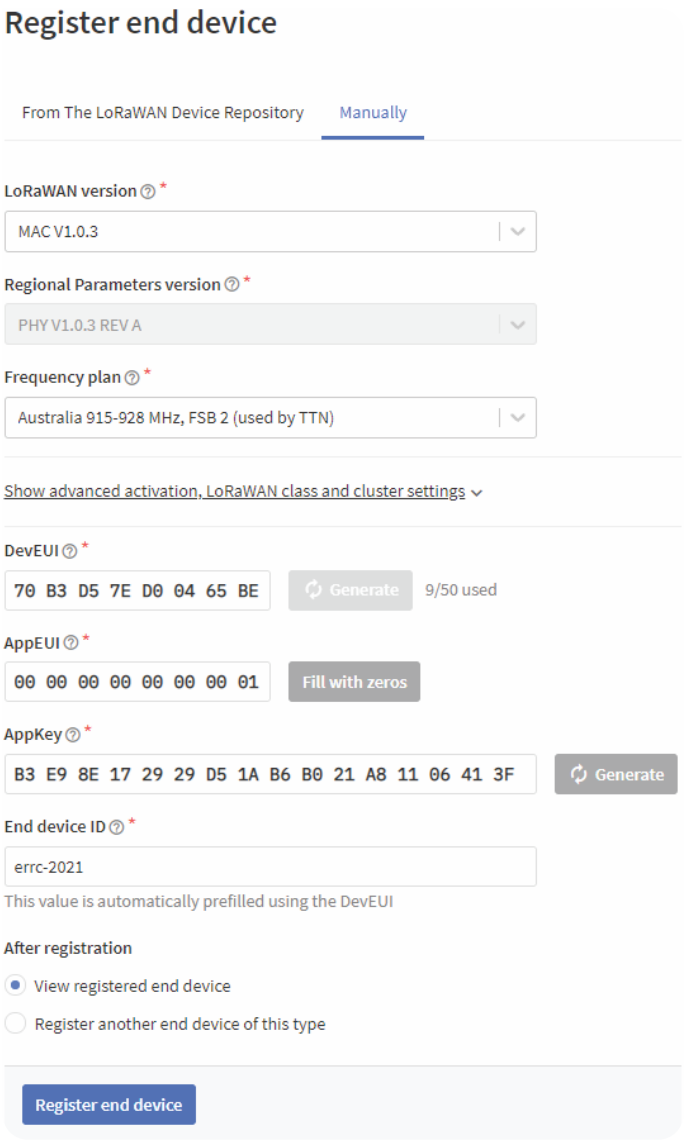

Figura 1.14: Registrando dispositivo final.

completo pode ser encontrado no repositório do minicurso no GitHub, sendo necessário somente modificar as chaves para corresponder com o dispositivo criado como na Listagem 6 e configurar a frequência. Para configurar a frequência, dentro da pasta da biblioteca LMIC existe um arquivo chamado `lmic_project_config`, este arquivo deve ser alterado de forma a selecionar a frequência a ser usada pelo dispositivo, no caso, `CFG_au915`.

Outro ponto importante a ser considerado é a variável que armazena a mensagem e a função que faz o envio da mensgem. Estamos enviando uma mensagem contendo a frase *"Hello World!"* atribuída à variável `payload`. A rotina responsável pelo envio da mensagem é `do_send`. Ela é configurada para um envio periódico, que pode ser alterado através da variável `TX_INTERVAL`.

Uma vez que o dispositivo foi programado basta ligá-lo e o tráfego de mensa-

Listagem 6 Chaves de autenticação dos dispositivos

```
// AppEUI em formato LSB
static const u1_t PROGMEM APPEUI[8] =  {0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};
// DevEUI em formato LSB
static const u1_t PROGMEM DEVEUI[8] =  {0xBE, 0x65, 0x04, 0xD0, 0x7E, 0xD5, 0xB3, 0x70};
// AppKey em formato MSB
static const u1_t PROGMEM APPKEY[16] = {0xB3, 0xE9, 0x8E, 0x17, 0x29, 0x29, 0xD5, 0x1A,
                                        0xB6, 0xB0, 0x21, 0xA8, 0x11, 0x06, 0x41, 0x3F};
```

gens começará a aparecer na console TTS. Inicialmente os dados serão mostrados em forma de *bytes*, portanto faz-se necessário a decodificação da menagem. Acessando a aba *Payload Formatters - Uplink* podemos incluir um código para formatar a mensagem. O código pode ser visualizado na Listagem 8, onde aplicamos a transformação para `String` somente aos *bytes* do *payload* completo, `input`.

Listagem 7 *Payload Formatter bytes para string*

```
function decodeUplink(input) {
    return {
        data: {
            msg: String.fromCharCode.apply(null, input.bytes)
        },
        warnings: [],
        errors: []
    };
}
```

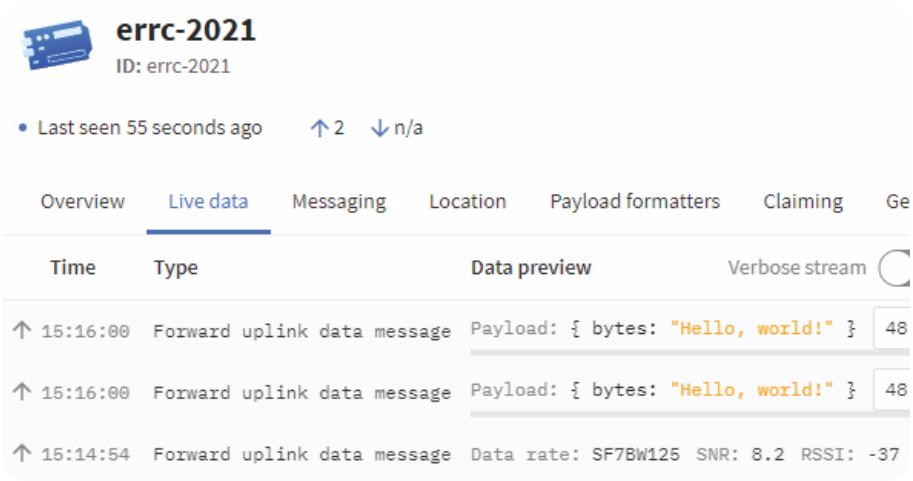

Figura 1.15: Tráfego de mensagens no servidor de rede TTS.

Exemplificando como enviar as mensagens para um servidor de aplicação utilizaremos a plataforma TagoIO[8]. Para isso, acessamos o *site* e geramos uma nova chave de autorização, em *Devices* clicando em *Authorization*. Nesta seção pode-se dar um nome a chave e clicar em *Generate*. Está chave é necessária para realizar a integração entre TTS e TagoIO. Ainda em *Devices*, adicionamos um novo dispositivo clicando em *Add Device*.

[8] `https://tago.io/`

Selecionamos a opção de dispositivo customizado presente em *LoRaWAN TTI/TTN v3*, atribuímos um nome ao dispositivo e colocamos o seu respectivo *DevEUI*. O painel *Devices* pode ser visto na Figura 1.16, onde estão destacados os botões principais do processo.

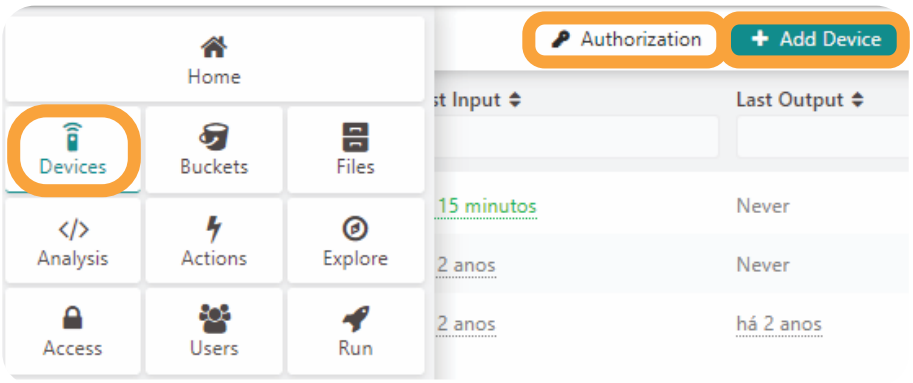

Figura 1.16: Seção de dispositivos da plataforma TagoIO.

De volta a TTS, adiconamos um *WebHook* para à TagoIO em *Integrations*, selecionando a opção TagoIO, como demonstrado na Figura 1.17. Neste momento, inserimos um nome para a integração e a chave gerada anteriormente pela TagoIO. Finalizando o processo de criação da integração ao clicar em *Create tagoio webhook*.

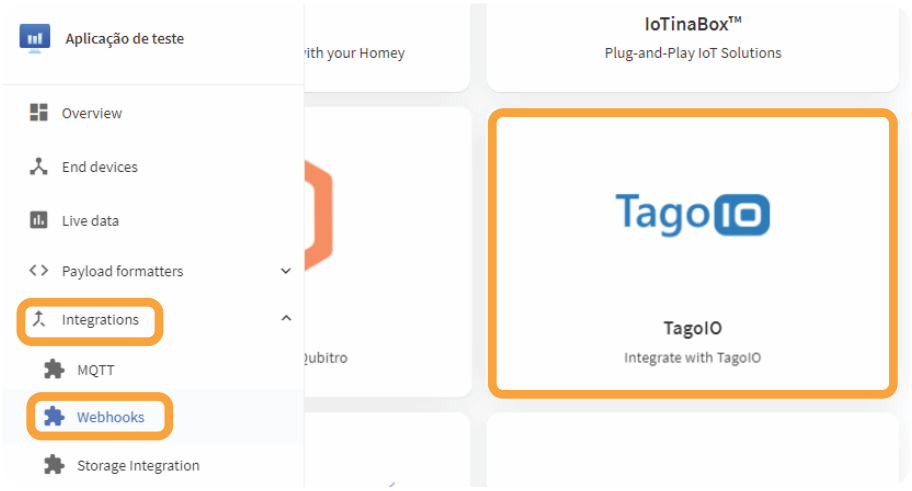

Figura 1.17: Seção de integrações *Webhooks* da TTS.

Alterando o código para enviar um valor de um contador conforme pressionamos um botão e também *Payload Formatter* para decodificar os *bytes* em um número inteiro, como na Listagem 8.

Nesta etapa a plataforma TagoIO fornece a criação de painéis. Estes painéis são formados por diferentes *widgets*, que acabam por exibir as informações recebidas do dispositivo. Para criar um painel clicamos em *Add Dashboard*, destacado na Figura 1.18, assim como o botão *Add Widget* que deve ser selecionado em seguida. Após clicar em *Add Widget* as opções de *widget* serão mostradas, como na Figura 1.19.

Listagem 8 *Payload Formatter bytes para inteiro*

```
function decodeUplink(input) {
    return {
        data: {
            counter: (input.bytes[0] << 8) +  input.bytes[1]
        },
        warnings: [],
        errors: []
    };
}
```

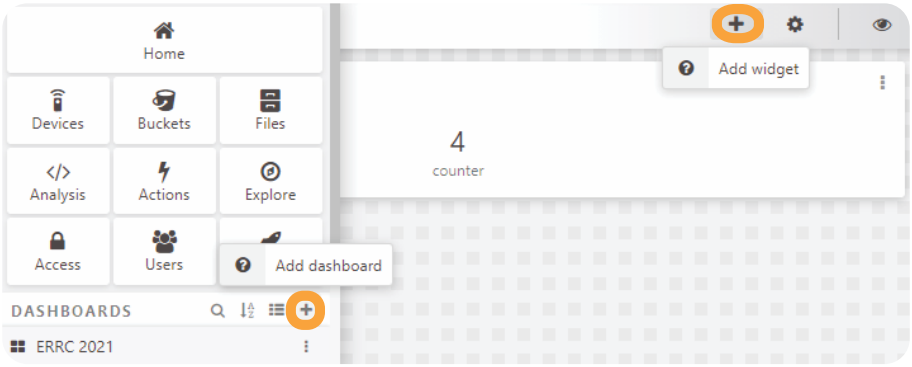

Figura 1.18: Adicionar painel e adicionar *widget* da TagoIO.

Como a variável a ser exibida é do tipo numérico, adicionamos um *Display*. Ao selecionar o *widget* a ser adicionado um outro painel é aberto para configurá-lo. Este painel pode ser visualizado na Figura 1.20. Aqui selecionamos o dispositivo criado e a variável a ser exibida, no caso, *counter* que foi definida no `Payload Formatter`.

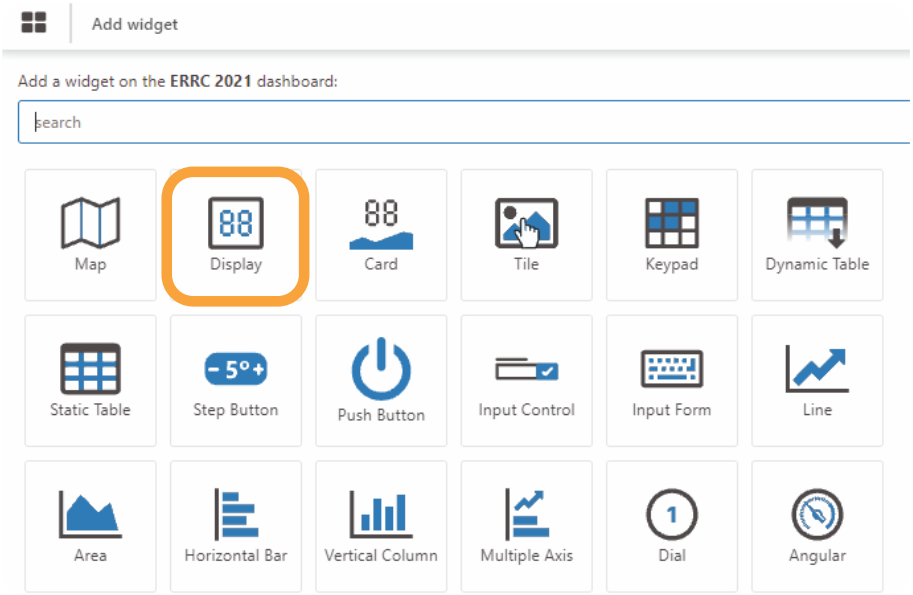

Figura 1.19: *Widgets* disponíveis na TagoIO.

Desta forma, toda vez que o botão do ESP32 Heltec for pressionado, o valor de

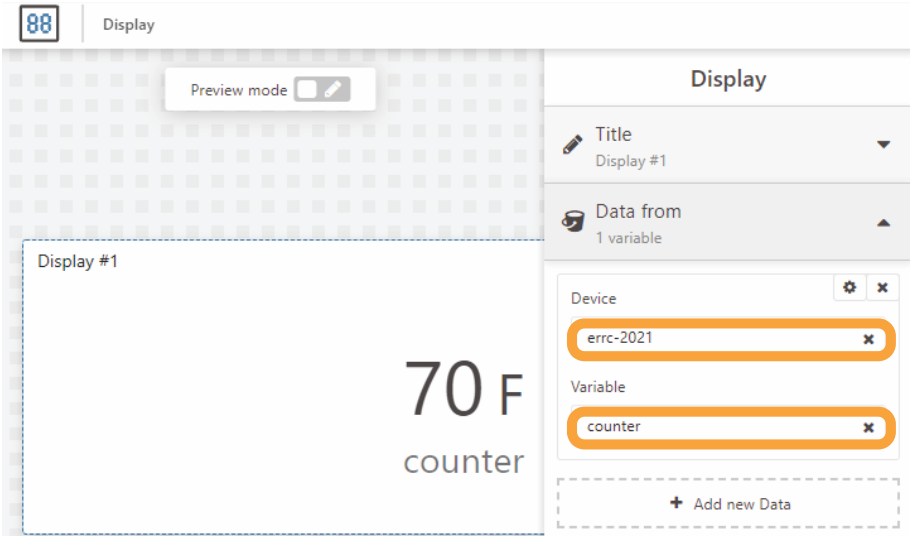

Figura 1.20: Configuração do *widget display*.

counter será incrementado e enviado para o servidor de rede TTS. O servidor decodifica e encaminha a mensagem para o servidor de aplicação, TagoIO, que por sua vez faz a exibição dos dados.

1.5. Considerações Finais

Este minicurso apresentou aspectos teóricos e práticos sobre o padrão aberto LoRaWAN e sua importância para fornecer conectividade às aplicações IoT em diversos contextos.

São muitas as possibilidades de desenvolvimento e pesquisa que a rede LoRaWAN apresenta. Destaca-se, por exemplo, iniciativas de instalação da rede LoRaWAN em diversas cidades brasileiras por operadoras como a American Tower[3] e TCTEC Telecom [38]. Há também iniciativas das próprias cidades aliadas a universidades, como na cidade de Toledo/PR, onde a prefeitura está fornecendo a conectvidade LoRaWAN para as aplicações IoT [5]. Em termos de oportunidades de pesquisa, destacam-se, as relacionadas a eficiência energética [26], tolerância a falhas e confiabilidade [28, 1, 27], mobilidade [29, 5, 36] e segurança [24].

Referências

[1] Thiago Almeida et al. "Em Busca de um Algoritmo de Retransmissão em Redes LoRaWAN". Em: *Anais do V Workshop de Computação Urbana*. Evento Online: SBC, 2021, pp. 70–83. URL: https://sol.sbc.org.br/index.php/courb/article/view/17105.

[2] ALSO. *AllThingsTalk*. URL: https://www.allthingstalk.com/. Acesso: 4 set/2021.

[3] ATC. *REDE ATC LoRaWAN*. URL: https://americantower.com.br/pt/solu%C3%A7%C3%B5es/rede-neutra-loRaWAN.html. Acesso: 7 out/2021.

[4] Bain. *Unlocking Opportunities in the Internet of Things*. Online. Acessado em 23/06/2019. 2018. URL: https://www.bain.com/insights/unlocking-opportunities-in-the-internet-of-things/.

[5] Edson Tavares de Camargo, Fabio Alexandre Spanhol e Álvaro Ricieri Castro e Souza. "Deployment of a LoRaWAN network and evaluation of tracking devices in the context of smart cities". Em: *Journal of Internet Services and Applications* 12.8 (out. de 2021), pp. 1–24. ISSN: 1869-0238. DOI: 10.1186/s13174-021-00138-7.

[6] M. Centenaro et al. "Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios". Em: *IEEE Wireless Communications* 23.5 (out. de 2016), pp. 60–67. ISSN: 1558-0687. DOI: 10.1109/MWC.2016.7721743.

[7] ChirpStack. *The ChirpStack project*. 2021. URL: https://www.chirpstack.io/project/. Acesso: 26 set/2021.

[8] CSA. *ZIGBEE - The Full-stack Solution Interlacing all your Smart Devices*. URL: https://zigbeealliance.org/solution/zigbee/. Acesso: 22 out/2021.

[9] Manishkumar Dholu e K.A. Ghodinde. "Internet of Things (IoT) for Precision Agriculture Application". Em: *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*. 2018, pp. 339–342. DOI: 10.1109/ICOEI.2018.8553720.

[10] Ala Al-Fuqaha et al. "Internet of things: A survey on enabling technologies, protocols, and applications". Em: *IEEE communications surveys & tutorials* 17.4 (2015), pp. 2347–2376.

[11] GSMA. *NB-IoT Deployment Guide to Basic Feature Set Requirements*. GSMA. 2019, p. 51. URL: https://www.gsma.com/iot/wp-content/uploads/2019/07/201906-GSMA-NB-IoT-Deployment-Guide-v3.pdf. Acesso: 07 out/2021.

[12] Heltec Automation. *WiFi LoRa 32 (V2)*. 2021. URL: https://heltec.org/project/wifi-lora-32/. Acesso: 22 set/2021.

[13] Tago LLC. *TagoIO*. URL: https://tago.io/. Acesso: 4 set/2021.

[14] LoRa Alliance. *What is LoRaWAN?* 2015. URL: https://lora-alliance.org/wp-content/uploads/2020/11/what-is-lorawan.pdf. Acesso: 28 oct/2021.

[15] LoRa Alliance. *LoRaWAN 1.1 Specification*. LoRa Alliance. 2017, p. 101. URL: https://lora-alliance.org/wp-content/uploads/2020/11/lorawantm_specification_-v1.1.pdf. Acesso: 23 set/2021.

[16] LoRa Alliance. *LoRaWAN® Back-End Interfaces v1.0*. 2017. URL: https://lora-alliance.org/wp-content/uploads/2020/11/lorawantm-backend-interfaces-v1.0.pdf. Acesso: 17 out/2021.

[17] LoRa Alliance. *LoRaWAN Certified Device*. 2021. URL: https://lora-alliance.org/showcase/search/?_sft_product_categories=end-node-sensors-or-other-devices. Acesso: 26 set/2021.

[18] LoRa Alliance. *LoRaWAN Development Board*. 2021. URL: https://lora-alliance.org/showcase/search/?_sft_product_categories=development-board,module. Acesso: 26 set/2021.

[19] LoRa Alliance. *LoRaWAN Gateway*. 2021. URL: https://lora-alliance.org/showcase/search/?_sft_product_categories=gateway. Acesso: 26 set/2021.

[20] Obaidulla Al-Mahmud et al. "Internet of Things (IoT) Based Smart Health Care Medical Box for Elderly People". Em: *2020 International Conference for Emerging Technology (INCET)*. 2020, pp. 1–6. DOI: 10.1109/INCET49848.2020.9153994.

[21] Timothy Malche e Priti Maheshwary. "Internet of Things (IoT) for building smart home system". Em: *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. 2017, pp. 65–70. DOI: 10.1109/I-SMAC.2017.8058258.

[22] MCCI. *Arduino LoRaWAN MAC in C (LMIC)*. 2019. URL: https://github.com/mcci-catena/arduino-lmic/blob/master/doc/LMIC-v3.0.99.pdf. Acesso: 22 set/2021.

[23] myDevices. *Cayenne*. URL: https://developers.mydevices.com/cayenne/features/. Acesso: 4 set/2021.

[24] Hassan Noura et al. "LoRaWAN security survey: Issues, threats and possible mitigation techniques". Em: *Internet of Things* 12 (2020), p. 100303. ISSN: 2542-6605. DOI: https://doi.org/10.1016/j.iot.2020.100303. URL: https://www.sciencedirect.com/science/article/pii/S2542660520301359.

[25] Lucas de Oliveira, Arlindo da Conceição e Lauro S. Neto. "Revisão sistemática da literatura sobre aplicações das tecnologias LoRa e LoRaWAN". Em: *Anais Estendidos do VIII Simpósio Brasileiro de Engenharia de Sistemas Computacionais*. Salvador: SBC, 2018. URL: https://sol.sbc.org.br/index.php/sbesc_estendido/article/view/11002.

[26] Javan de Oliveira Júnior e Marcio Oyamada. "Avaliando o impacto da compressão de dados no desempenho e energia em redes LoRa". Em: *Anais Estendidos do X Simpósio Brasileiro de Engenharia de Sistemas Computacionais*. Evento Online: SBC, 2020, pp. 81–88. DOI: 10.5753/sbesc_estendido.2020.13094. URL: https://sol.sbc.org.br/index.php/sbesc_estendido/article/view/13094.

[27] André Pastório e Edson Camargo. "Técnicas de Geolocalização em Redes LoRaWAN como Abordagem de Tolerância a Falhas para Dispositivos IoT Baseados em GPS". Em: *Anais do XXII Workshop de Testes e Tolerância a Falhas*. Evento Online: SBC, 2021, pp. 29–42. URL: https://sol.sbc.org.br/index.php/wtf/article/view/17202.

[28] André Pastório, Luiz Rodrigues e Edson de Camargo. "Uma Revisão Sistemática da Literatura Sobre Tolerância a Falhas em Internet das Coisas". Em: *Anais Estendidos do X Simpósio Brasileiro de Engenharia de Sistemas Computacionais*. Evento Online: SBC, 2020, pp. 57–64. DOI: 10.5753/sbesc_estendido.2020.13091. URL: https://sol.sbc.org.br/index.php/sbesc_estendido/article/view/13091.

[29] João Pastório, Edson Tavares de Camargo e Álvaro Ricieri Castro e Souza. "Simulação do monitoramento de veículos em uma rede LoRaWAN usando NS-3". Em: *ERRC 2021 - IC ()*. Out. de 2021. URL: http://XXXXX/220180.pdf.

[30] Radioenge. *Gateway LoRaWAN*. 2021. URL: https://www.radioenge.com.br/solucoes/iot/18-gateway-lorawan.html. Acesso: 26 set/2021.

[31] Radioenge. *Módulo LoRaWAN*. 2021. URL: https://www.radioenge.com.br/solucoes/iot/17-modulo-lorawan.html. Acesso: 26 set/2021.

[32] Swathi Ramnath et al. "IoT based localization and tracking". Em: *2017 International Conference on IoT and Application (ICIOT)*. 2017, pp. 1–4. DOI: 10.1109/ICIOTA.2017.8073629.

[33] Jonas Rossato, Fabio Spanhol e Edson Camargo. "Implantação e Avaliação de uma Rede Sem-Fio de Longo Alcance e Baixa Potência para Cidades Inteligentes". Em: *Anais do IV Workshop de Computação Urbana*. Rio de Janeiro: SBC, 2020, pp. 192–205. DOI: 10.5753/courb.2020.12363. URL: https://sol.sbc.org.br/index.php/courb/article/view/12363.

[34] Sureshkumar Selvaraj e Suresh Sundaravaradhan. "Challenges and opportunities in IoT healthcare systems: a systematic review". Em: *SN Applied Sciences* 2 (jan. de 2020). DOI: 10.1007/s42452-019-1925-y.

[35] Sigfox. *Sigfox Technology*. 2021. URL: https://www.sigfox.com/en/what-sigfox/technology#id_technology. Acesso: 7 out/2021.

[36] Marcos da Silva et al. "Avaliação de Dispositivos de Rastreamento em uma Rede LoRaWAN no Contexto de Cidades Inteligentes". Em: *Anais do IV Workshop de Computação Urbana*. Rio de Janeiro: SBC, 2020, pp. 1–14. DOI: 10.5753/courb.2020.12349. URL: https://sol.sbc.org.br/index.php/courb/article/view/12349.

[37] Statista. *Forecast end-user spending on IoT solutions worldwide from 2017 to 2025*. acesso em 30/03/2020. 2020. URL: https://www.statista.com/statistics/976313/global-iot-market-size/.

[38] TCT. *TCT Telecom*. URL: https://www.tctectelecom.com/. Acesso: 7 out/2021.

[39] The Things Industries. *The Things Stack (TTS)*. 2021. URL: shorturl.at/klwTU. Acesso: 22 set/2021.

[40] TTN. *The Things Network*. URL: https://www.thethingsnetwork.org/. Acesso: 7 set/2021.

[41] Antonio Velez-Estevez, Lorena Gutiérrez-Madroñal e Inmaculada Medina-Bulo. "IoT-TEG 4.0: A New Approach 4.0 for Test Event Generation". Em: *IEEE Transactions on Reliability* (2021), pp. 1–13. DOI: 10.1109/TR.2021.3087781.

[42] vXchnge. *Comprehensive Guide to IoT Statistics You Need to Know in 2020*. acesso em 30/03/2020. 2020. URL: https://www.vxchnge.com/blog/iot-statistics.