

## Estudos e experimentos com redes LPWAN utilizando LoRaWAN

### Research and experiments with LPWANs using LoRaWAN

#### RESUMO

Afonso Henrique Kingeski  
[afonsokingeski@alunos.utfpr.edu.br](mailto:afonsokingeski@alunos.utfpr.edu.br)  
Universidade Tecnológica Federal  
do Paraná, Curitiba, Paraná, Brasil

João Luiz Rebelatto  
[jrebelatto@gmail.com](mailto:jrebelatto@gmail.com)  
Universidade Tecnológica Federal  
do Paraná, Curitiba, Paraná, Brasil

Marcos Eduardo Pivaro Monteiro  
[marcose@utfpr.edu.br](mailto:marcose@utfpr.edu.br)  
Universidade Tecnológica Federal  
do Paraná, Curitiba, Paraná, Brasil

A difusão da Internet das Coisas fez surgir inovações na área de telecomunicação, entre elas as LPWAN (do inglês *Low Power Wide Area Networks*), tecnologias que visam transmitir dados a dezenas de quilômetros com ínfimo custo econômico e energético. Dentre as tecnologias LPWAN, se destaca o LoRa (do inglês *Long Range*), uma técnica de modulação CSS (do inglês *Chirp Spread Spectrum*) de propriedade da Semtech Corporation, cujo protocolo de comunicação mais divulgado é o LoRaWAN. Esta pesquisa teve como objetivo a análise e documentação dos métodos necessários para a criação de uma rede LoRaWAN, além da configuração e incorporação de um *End Node* na rede. Todo o processo auxiliou na consolidação dos conceitos abordados e gerou documentos que facilitam a reprodução dos procedimentos para novos desenvolvimentos.

**PALAVRAS-CHAVE:** LoRa, LoRaWAN, LPWAN

#### ABSTRACT

**Recebido:** 19 ago. 2020.

**Aprovado:** 01 out. 2020.

**Direito autorial:** Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.



The spread of the Internet of Things led to innovations in the area of telecommunications, among them the LPWAN (Low Power Wide Area Networks), technologies that aim to transmit data at tens of kilometers with minimal economic and energy cost. Among the LPWAN technologies stand out the LoRa (Long Range), a CSS (Chirp Spread Spectrum) modulation technique owned by Semtech Corporation, whose most publicized communication protocol is LoRaWAN. This research aims the analysis and documentation of the methods necessary for the creation of a LoRaWAN network, besides the configuration and incorporation of an End Node in the network. The whole process helped consolidate the concepts addressed and generated documents that make it easier for the reproduction of the procedures for new developments.

**KEYWORDS:** LoRa, LoRaWAN, LPWAN.



## INTRODUÇÃO

A difusão da Internet das Coisas (IoT, do inglês Internet of Things) possibilitou a disseminação de novas tecnologias de telecomunicação e transmissão de dados. A busca por dispositivos que cobrissem uma grande área geográfica a um baixo custo de energia fez com que surgisse um novo ramo da IoT, as LPWAN (do inglês Low Power Wide Area Networks) - uma vez que as tecnologias convencionais, como o Bluetooth ou ZigBee, cobrem uma área reduzida (SUNDARAM, 2020).

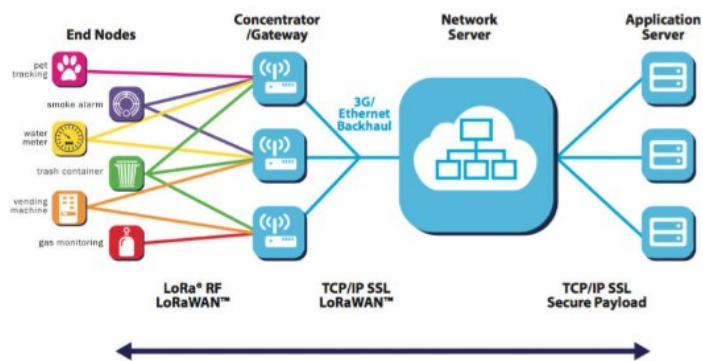
As tecnologias LPWAN objetivam cobertura de áreas na escala de dezenas de quilômetros a um baixíssimo custo econômico e de consumo de bateria (PETÄJÄRVI, 2015). Entretanto, tais tecnologias enfrentam vários desafios de implementação e disseminação, que vão desde dificuldades em padronização a trade-offs para manter o baixo custo (SUNDARAM, 2020).

Uma das tecnologias LPWAN que se destacam, pelo seu vasto uso em IoT e em trabalhos acadêmicos, é LoRa (do inglês, Long Range), uma técnica de modulação CSS (do inglês Chirp Spread Spectrum) de propriedade da Semtech Corporation (SUNDARAM, 2020). Ela possui propriedades únicas que justificam seu amplo uso: atinge uma grande área de cobertura; baixo custo dos componentes; grande durabilidade de bateria; capacidade de recepção simultânea nos gateways e resistência ao efeito Doppler (IKPEHAI, 2019).

A LoRa é a primeira implementação CSS de baixo custo para uso comercial (A TECHNICAL..., 2015). A troca de potência ou cobertura por taxa de transmissão em uma mesma largura de banda se deve a três parâmetros base: a largura de banda (BW, do inglês Bandwidth), o fator de espalhamento e um indicador CR (do inglês, Code Rate). As larguras de banda geralmente utilizadas são as de 125 kHz, 250 kHz ou 500 kHz, valores diretamente proporcionais a taxa efetiva de dados transmitidos. O fator de espalhamento pode variar entre 7 e 12 e, aumentando seu valor, tem-se uma melhor confiabilidade no sinal transmitido - em caso de precariedade do canal, por exemplo - mas a taxa de transmissão cai significativamente. Variando de 0 a 4, o CR é um parâmetro que determina a taxa do código corretor de erro (VANGELISTA, 2015).

A LoRa convive e conversa com arquiteturas de rede já existentes. Isto permite que diversas pilhas de protocolos funcionem com a tecnologia (LORA MODULATION..., 2015). Estes protocolos, junto à arquitetura de rede, definem características como segurança, vida útil de baterias, capacidade e qualidade do serviço. Atualmente, o protocolo de comunicação mais divulgado é a LoRaWAN. Utilizando a topologia estrela-de-estrelas, sua arquitetura de rede está esquematizada na Figura 1. Os gateways fazem as comunicações entre End Nodes através da comunicação sem fio single-hop. Os End Nodes não são atrelados a um único gateway, ou seja, os dados transmitidos por um End Node são aceitos por diversos gateways. Estes dados são então enviados para o servidor da rede na nuvem, através de outro tipo de transmissão, como Wi-Fi (do inglês, wireless fidelity) ou Ethernet. Toda a parte de processamento é deixada então para o servidor de rede, que realizará testes de segurança, filtrará dados redundantes e fará outros procedimentos mais complexos (A TECHNICAL..., 2015), como gerenciar as taxas de transmissão. Por fim, o servidor de aplicação tem a principal função de descriptografar as mensagens do End Node e repassá-las para a utilização da implementação.

Figura 1 – Esquemático da rede LoRaWAN



Fonte: Lora Alliance (A TECHNICAL..., 2015).

Embora existam progressos na comercialização e padronização da tecnologia, um obstáculo a ser superado é a escalabilidade da rede, uma vez que probabilidade de cobertura cai exponencialmente com o aumento do número de dispositivos na rede (GEORGIU, 2017). A escalabilidade do LoRa não só é afetada pela interferência de transmissões que utilizam o mesmo fator de espalhamento, mas também por interferência em canais com fator de espalhamento diferente, visto que eles não são completamente ortogonais (MAHMOOD, 2019). Tentativas de aprimoramento incluem o uso de múltiplas antenas e replicação de transmissões (HOELLER, 2018), e novos métodos de replicação codificada (SANT'ANA, 2020).

Embora não seja o foco do LoRa, uma questão importante é a baixa taxa de dados por segundo ofertada, o que limita aplicações que necessitam de um grande tráfego de dados ou baixa latência (HSU, 2018). Para amenizar o problema, um dispositivo na rede poderia utilizar seus vizinhos para impulsionar uma transmissão mais volumosa, já que geralmente os dispositivos IoT estão ociosos (HSU, 2018).

Outra abordagem é a utilização de comunicação direta entre os dispositivos da rede. Além de permitir uma maior taxa de dados e oferecer uma menor latência, o uso de comunicação direta entre os dispositivos diminui a carga da rede (MIKHAYLOV, 2017). Os reveses da comunicação direta é que informações da transmissão não são armazenadas pela rede, possíveis interferências internas e problemas de segurança e privacidade (MIKHAYLOV, 2017), embora já existam iniciativas para o último (KIM, 2018).

Neste contexto, esta pesquisa possibilita uma melhor compreensão das etapas de comunicação na rede LoRaWAN e, salientando a comunicação dos dispositivos com a rede, é possível utilizar a mesma biblioteca empregada na criação do firmware customizado para implementar a comunicação direta entre dois End Nodes.

## MATERIAIS E MÉTODOS

Para este projeto foram utilizados os seguintes materiais:

- Módulo LoRaWAN RD49C, Radioenge
- Gateway LoRaWAN RD43HAT, Radioenge
- Raspberry Pi 3 Model B+
- Placa FTDI FT232RL
- ST-LINK V2, STM32Microeletronics

O primeiro passo foi a implementação de uma rede LoRaWAN utilizando o Gateway LoRaWAN RD43HAT da Radioenge, conectado ao Raspberry Pi. O sistema operacional Raspbian foi instalado utilizando um gravador de imagem, o Win32DiskManager (WIN32DISKIMAGER, 2020), em um SD Card. Com o Raspbian (RASPBERY PI OS, 2020) instalado no Raspberry Pi, foi possível habilitar SSH (*Secure Shell*) e controlá-lo remotamente através do *software* Putty (PUTTY, 2020).

Foi habilitado o SPI (*Serial Peripheral Interface*) no Raspberry Pi e instalado o Git (GIT, 2020) para obter as bibliotecas e aplicações necessárias à rede. Então, foram instalados o Lora Gateway project (LORA..., 2020), uma biblioteca com o código fonte para a construção de um receptor de radiofrequência, e o *packet forwarder* (PACKET, 2020), um programa que encaminha os pacotes de radiofrequência recebidos pelo *gateway* para o servidor e vice-versa, ambos fornecidos pela Semtech. A configuração das larguras de banda utilizadas puderam ser obtidas no repositório gateway-conf (TTN, 2020), da TheThingsNetwork, uma vez que as configurações regionais do Brasil coincidem com as configurações da Austrália.

Com o *packet forwarder* funcionando, foi necessária a instalação de outros elementos da rede LoRaWAN. Para isto, foram utilizados os serviços do Chirpstack LoRaWAN Network Server (CHIRPSTACK, 2020), que fornece os componentes da rede em código aberto assinados por uma chave PGP (*Pretty Good Privacy*). O Chirpstack utiliza um servidor MQTT (*Message Queuing Telemetry Transport*) para envio e recebimento de *payloads* nas aplicações. Portanto, foi instalado no Raspberry Pi a versão mais recente do Mosquitto (MOSQUITTO, 2020), um *broker* de mensagens que implementa o protocolo de comunicação MQTT. Além disto, o Chirpstack utiliza o banco de dados PostgreSQL para o armazenamento de dados persistentes e o banco de dados Redis para armazenamento de dados não-persistentes. Desta forma, ambos os serviços foram instalados no Raspberry Pi e dois bancos de dados PostgreSQL foram criados, um para o *ChirpStack Network Server* e outro para o *ChirpStack Application Server*. O Chirpstack oferece em um repositório todos os pacotes dos serviços assinados por uma chave PGP. Assim, o repositório e a chave de assinatura foram adicionados e os seguintes serviços foram instalados:

- *ChirpStack Gateway Bridge*: processa a comunicação com os *gateways* LoRaWAN.
- *ChirpStack Network Server*: implementação de um LoRaWAN Network Server
- *ChirpStack Application Server*: implementação de um LoRaWAN Application Server.

No arquivo de configuração do *ChirpStack Network Server*, foram adicionados o *login* e senha de seu banco de dados criado anteriormente e as informações sobre a faixa de frequência e sub-faixas utilizadas – de acordo com a configuração do *packet forwarder*. No arquivo de configuração do *ChirpStack Application Server* também foram adicionados o *login* e senha de seu respectivo banco de dados, além de uma senha na base 64 para o JWT (*JSON Web Token*).

Embora a Radioenge disponibilize um *software* para configuração do módulo, este era pouco intuitivo e não salientava partes importantes do processo. Então, foi adotada a comunicação serial para a configuração através de comandos AT, por meio de um conversor USB Serial FTDI. Foi utilizado o programa Termite (TERMITE, 2020), um terminal RS232, para envio e recebimento de mensagens do módulo.

Primeiro, foi alterada a máscara de canais do módulo para que ela se adequasse às sub-faixas configuradas no *packet forwarder*. O parâmetro *Adaptative Datarate* foi desativado e a taxa de dados foi ajustada para um valor maior que zero. Então, o modo de ativação do módulo na rede foi escolhido. Ao definir o modo de ativação, o módulo retorna os parâmetros necessários para incluí-lo na rede LoRaWAN, como o EUI (*Extended Unique Identifier*) do dispositivo e a *Application Key*. Ambos os modos de ativação, OTAA (*Over the Air Authentication*) e ABP (*Authentication by Personalisation*), foram testados, notando-se uma maior facilidade no modo OTAA, uma vez que este exige menos campos de configuração.

O *firmware* do Módulo LoRaWAN RD49C não é código aberto, portanto foi necessária a busca de repositórios que permitissem um *firmware* customizado. Primeiramente, foi desativada a proteção de bytes do módulo utilizando o programa STM32Cube Programmer (STM32CUBEPROGRAMMER..., 2020). Então, foi utilizado um ST-Link V2 para gravar um novo *firmware* através do *software* STM32 System Workbench (STM32, 2020), um programa baseado no Eclipse (ECLIPSE, 2020). Para testes, um programa simples de comutação GPIO (*General Purpose Input Output*) foi obtido através dos pacotes STM32Cube MCU (STM32CUBE..., 2020a), da Semtech. O pacote STM32CubeLO (STM32CUBE..., 2020b) não contém o microcontrolador presente no Módulo LoraWAN RD49C (STM32L071CZ), mas contém o STM32L073RZ, de pinagem semelhante. Corrigindo as divergências entres os microcontroladores, a comutação funcionou corretamente.

Para o *firmware* customizado, a primeira opção foi o repositório Arduino-Lmic, uma biblioteca IBM LMIC (IBM, 2016) modificada para funcionar com a Arduino IDE (ARDUINO, 2020). Então, foi adicionada a biblioteca Arduino-LMIC na IDE do Arduino e selecionada a placa Nucleo-64 junto a Nucleo L073RZ. No arquivo de configuração, foi definido o transmissor do Módulo LoraWAN RD49C (SX1272) e os parâmetros para ativação do módulo na rede LoRaWAN. Por fim, foi aberto um exemplo da biblioteca para a entrada do módulo na rede por OTAA. Infelizmente, a transmissão de dados era inconsistente e, portanto, outro *firmware* era exigido.

O repositório encontrado foi LmiC (LMIC, 2020), um variante da biblioteca IBM LMIC (IBM, 2016) com foco na portabilidade. O código foi alterado para que as configurações e conexões das portas do microcontrolador combinassem com o Módulo LoraWAN RD49C. Mesmo com os parâmetros da biblioteca devidamente ajustados, havia um problema de sincronização na recepção de mensagens, identificado como uma contagem de tempo acelerada pelo LSI (*Low Speed*

*Internal*) clock do microcontrolador. O programa STM32CubeMX (STM32CUBEMX, 2020) foi utilizado para obter as configurações necessárias para que o HSE (*High Speed External*) clock gerasse a contagem de tempo. Desta forma, o módulo foi aceito pela rede.

## RESULTADOS

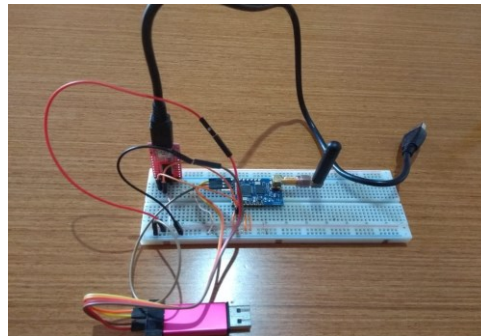
Com a rede LoRaWAN pronta, com todos os serviços funcionando no Raspberry Pi, foi possível testá-la enviando mensagens com o Módulo LoraWAN RD49C para o servidor de aplicação e vice-versa. Os testes de envio e recebimento de mensagens foram realizados tanto para com o módulo com o *firmware* original de fábrica quanto para o *firmware* customizado, sendo bem-sucedidos em ambos os casos.

Figura 2 - Gateway LoRaWAN RD43HAT sobre um raspberry Pi.



Fonte: Autoria Própria (2020).

Figura 3- Módulo LoraWAN RD49C conectado à placa FTDI FT232RL e ao ST-LINK V2



Fonte: Autoria Própria (2020).

Cada serviço do Chirpstack fornece um *log-output*, que mostra os pacotes de radiofrequência recebidos e enviados. Desta forma, foi possível observar um pacote enviado pelo Módulo LoraWAN RD49C chegando ao *packet forwarder*, sendo enviado para o ChirpStack Gateway Bridge, repassado para o ChirpStack Network Server e alcançando o ChirpStack Application Server. Também foi possível verificar o caminho oposto, quando um *payload* era enviado para módulo, passando por todas as camadas da rede.

Figura 4 - log-output do programa *packet forwarder*

```

192.168.3.12 - PuTTY
# TX rejected (collision beacon): 0.00% (req:42, rej:0)
# TX rejected (too late): 0.00% (req:42, rej:0)
# TX rejected (too early): 0.00% (req:42, rej:0)
# BEACON queued: 0
# BEACON sent so far: 0
# BEACON rejected: 0
### [JIT] ###
# SNI301 time (PPS): 182433297
src/jitqueue.c:442:jit_print_queue(): INFO: [jit] queue contains 1 packets:
src/jitqueue.c:453:jit_print_queue(): INFO: [jit] queue contains 0 beacons:
src/jitqueue.c:459:jit_print_queue(): - node[0]: count_us=213048851 - type=0
### [GPS] ###
# GPS +PTMC coordinates: latitude 0.00000, longitude 0.00000, altitude 0 m
##### END #####

JSON up: {"stat":{"time":"2020-07-18 18:55:29 GMT","lati":0.00000,"long":0.00000,"alti":0,"zmb":5,"zmk":5,"zxfw":5,"packr":100.0,"dmb":5,"xmb":4}}
INFO: [up] PUSH_ACK received in 1 ms
INFO: tx_start_delay=1497 (1497.000000) - (1497, bw_delay=0.000000, notch_delay=0.000000)
INFO: Received pkt from mote: 01944E42 (fcnt=4)

JSON up: {"expk":{"time":213134588,"chan":5,"rfch":1,"freq":904.900000,"erat":1,"modu":"LORA","datr":"SF7BR125","codr":4/6,"lancr":9.0,"rpsr":-33,"size":16,"data":"QEJ0LAGCBASQgRk5td/q=="}}
INFO: [up] PUSH_ACK received in 0 ms
INFO: [down] FULL_RESP received - token[93:175] :)

JSON down: {"txpk":{"time":false,"rfch":0,"powe":20,"ant":0,"brd":0,"tast":214134588,"freq":926.3,"modu":"LORA","datr":"SF7BR500","codcr":4/6,"spol":true,"size":22,"data":"YEJ0LAGKRCAGB7EynXmcZt/Wi/qjVA=="}}
    
```

Fonte: Autoria Própria (2020).

Em síntese, os resultados obtidos atenderam às expectativas. A biblioteca utilizada para o *firmware* se provou bem adaptável, uma vez que vários dos parâmetros necessários foram facilmente alterados no código. Além disso, a solução apresentada ao final do projeto é bastante estável, em razão do envio e recebimento de mensagens entre as camadas, bem como a entrada do módulo na rede, serem realizadas com êxito.

## CONCLUSÃO

As atividades realizadas nesta iniciação científica se provaram bastante proveitosas. O estudo de artigos, *datasheets* e manuais permitiram uma grande compreensão não somente do protocolo LoRaWAN em si, mas também das tecnologias LPWAN. Muito embora estas ainda enfrentem problemas de implementação e padronização, existe um interesse acadêmico e econômico muito grande por trás da tecnologia, e novas soluções surgem diariamente.

A modulação LoRa, junto ao protocolo LoRaWAN, provou-se uma solução bastante notável e confiável, destacando-se entre as demais tecnologias do setor. A criação de uma rede LoRaWAN do zero favoreceu um melhor entendimento das características e propriedades do LoRa, além de propiciar conhecimentos não diretamente relacionados, como o uso do terminal no Raspberry Pi e atuar com programação de microcontroladores.

Quanto à documentação de todos os processos realizados (CRIAÇÃO..., 2020), esta facilitará a reprodução destes para o trabalho de pesquisa da comunidade acadêmica. Além disto, ela permite um ponto de partida mais acessível para novos desenvolvimentos tecnológicos.

## AGRADECIMENTOS

O presente trabalho foi realizado com o apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico CNPq – Brasil, da Fundação Araucária FA – Paraná/Brasil e da Universidade Tecnológica Federal do Paraná/Brasil.

## REFERÊNCIAS

SUNDARAM, J. P. S.; DU, W.; ZHAO, Z.. A Survey on LoRa Networking: Research Problems, Current Solutions, and Open Issues. **IEEE Communications Surveys and Tutorials**, v. 22, n. 1, p. 371–388, 2020.

PETÄJÄJÄRVI, J. *et al.* On the Coverage of LPWANs: Long Evaluation and Channel Attenuation Model for LoRa Technology. *In: Proc. 14th Int. Conf. ITS Telecommun. (ITST). Proceedings...*, p. 55–59, dez. 2015.

IKPEHAI, A. *et al.* Low-Power Wide Area Network Technologies for Internet-of-Things: A Comparative Review. **IEEE Internet of Things Journal**, v. 6, n. 2, p. 2225–2240, abril 2019.

A TECHNICAL overview of lora and lorawan. **Lora Alliance**.Tech. Rep., nov. 2015.

VANGELISTA, L.; ZANELLA, A.; ZORZI, M.. Long-Range IoT technologies: The dawn of lora. *In: 1st EAI International Conference on Future access enablers of ubiquitous and intelligent infrastructure. Proceedings...*, p. 51–58, set.2015.

LORA MODULATION basics. **Semtech**.Tech. Rep., AN1200.22,maio.2015.

GEORGIU, O.; RAZA, U.. Low Power Wide Area Network analysis: Can LoRa scale?. **IEEE Wireless Communications Letters**, v. 6. n. 2.p. 162–165. abril 2017.

MAHMOOD, A. *et al.* Scalability analysis of a LoRa network under imperfect orthogonality. **IEEE Transactions on Industrial Informatics**, v. 15, n. 3, p. 1425–1436, mar. 2019.

HOELLER, A. *et al.* Analysis and performance optimization of LoRa Networks with time and antenna diversity. **IEEE Access**, v. 6, p. 32 820–32 829, 2018.

SANT'ANA, J. M. S.. *et al.* Hybrid Coded Replication in LoRa Networks. **IEEE Transactions on Industrial Informatics**, v. PP, p. 1–1, jan. 2020.

HSU, S. *et al.* Breaking Bandwidth Limitation for Mission-Critical IoT Using Semisequential Multiple Relays. **IEEE Internet of Things Journal**, v. 5, n. 5. p.3316–3329, out. 2018.



MIKHAYLOV, K. *et al.* D2D communications in LoRaWAN Low Power Wide Area Network: From idea to empirical validation. *In*: 2017 IEEE International Conference on Communications Workshops (ICC Workshops). **Proceedings...**, p. 737–742, maio 2017.

KIM, J.; SONG, J.. A secure device-to-device link establishment scheme for LoRaWAN. **IEEE Sensors Journal**, v. 18, n. 5, p. 2153–2160, mar. 2018.

WIN32DISKIMAGER. Disponível em:  
<https://sourceforge.net/projects/win32diskimager/>. Acesso em: 29 jul. 2020.

RASPBERRY PI OS. Disponível em:  
<https://www.raspberrypi.org/downloads/raspberry-pi-os/>. Acesso em: 29 jul. 2020.

PUTTY. Disponível em: <https://www.putty.org/>. Acesso em: 30 jul. 2020.

GIT. Disponível em: <https://git-scm.com/>. Acesso em: 29 jul. 2020.

LORA gateway project. Disponível em: <https://github.com/Lora-net/loragateway>. Acesso em: 29 jul. 2020.

PACKET Forwarder. Disponível em: <https://github.com/Lora-net/packetforwarder>. Acesso em: 29 jul. 2020.

TTN Master Gateway configurations. Disponível em:  
<https://github.com/TheThingsNetwork/gateway-conf>. Acesso em: 29 jul. 2020.

CHIRPSTACK Open-source Lorawan Network Server. Disponível em:  
<https://www.chirpstack.io/>. Acesso em: 04 ago. 2020.

MOSQUITTO. Disponível em: <https://mosquitto.org/>. Acesso em: 04 ago. 2020.

TERMITE. Disponível em: [https://www.compuphase.com/software\\_termite.htm](https://www.compuphase.com/software_termite.htm). Acesso em: 30 jul. 2020.

STM32CUBEPROGRAMMER *software* for all STM32. Disponível em:  
<https://www.st.com/en/development-tools/stm32cubeprog.html#get-software>. Acesso em: 02 ago. 2020.

STM32 System Workbench. Disponível em:

<https://www.openstm32.org/Installing%2BSystem%2BWorkbench%2Bfor%2BSTM32%2Bwith%2Binstaller>. Acesso em: 02 ago. 2020.

ECLIPSE. Disponível em: <https://www.eclipse.org/>. Acesso em: 02 ago. 2020.

STM32CUBE MCU MPU packages. Disponível em:

<https://www.st.com/en/embedded-software/stm32cube-mcu-mpu-packages.html#products>. Acesso em: 02 ago. 2020.

STM32CUBE MCU package for stm32l0 series. Disponível em:

<https://www.st.com/content/stcom/en/products/embedded-software/mcu-mpu-embedded-software/stm32-embedded-software/stm32cube-mcu-mpu-packages/stm32cubel0.html>. Acesso em: 02 ago. 2020.

IBM. **IBM LoRaWAN in C (LmiC)**. IBM Zurich Research Laboratory, 8803 Rüschlikon, Switzerland, version 1.6. jul. 2016.

ARDUINO. Disponível em: <https://www.arduino.cc/en/Main/software>. Acesso em: 04 ago. 2020.

LMIC. Disponível em: <https://github.com/dudmuck/LmiC>. Acesso em: 04 ago. 2020.

STM32CUBEMX. Disponível em: <https://www.st.com/en/development-tools/stm32cubemx.html>. Acesso em: 04 ago. 2020.

CRIAÇÃO de uma Rede LoRaWAN. <https://github.com/AfonsoKingeski/Cria-o-de-uma-rede-LoraWAN.git>. Acesso em: 09 out. 2020.