

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA**

**GABRIEL MENDES SIMONI
GEOVANA SCARAMELLA**

**MONITORAMENTO DE VEÍCULOS ALTERNATIVOS:
DESENVOLVIMENTO DE UM PROTÓTIPO COM TECNOLOGIA
LORAWAN**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2021

GABRIEL MENDES SIMONI
GEOVANA SCARAMELLA

**MONITORAMENTO DE VEÍCULOS ALTERNATIVOS:
DESENVOLVIMENTO DE UM PROTÓTIPO COM TECNOLOGIA
LORAWAN**

Trabalho de Conclusão de Curso apresentado como requisito para obtenção do grau de Bacharel em Engenharia Eletrônica, do Departamento Acadêmico de Eletrônica, da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Prof. Dr. Hugo Vieira Neto
Coorientadora: Profa. Dra. Carmen Caroline Rasesa

CURITIBA

2021

GABRIEL MENDES SIMONI
GEOVANA SCARAMELLA

**MONITORAMENTO DE VEÍCULOS ALTERNATIVOS: DESENVOLVIMENTO DE
UM PROTÓTIPO COM TECNOLOGIA LORAWAN**

Trabalho de Conclusão de Curso apresentado como requisito para obtenção do grau de Bacharel em Engenharia Eletrônica, do Departamento Acadêmico de Eletrônica, da Universidade Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 30 de agosto de 2021.

Prof. Dr. Hugo Vieira Neto
Universidade Tecnológica Federal do Paraná

Profa. Dra. Carmen Caroline Rasera
Universidade Tecnológica Federal do Paraná

Prof. Me. Luiz Fernando Copetti
Universidade Tecnológica Federal do Paraná

Prof. Dr. Luciano Scandelari
Universidade Tecnológica Federal do Paraná

CURITIBA
2021

Dedicamos este trabalho ao nosso orientador,
professor Hugo Vieira Neto (in memoriam), que
será para sempre um exemplo de gentileza,
dedicação e excelência.

AGRADECIMENTOS

Expressamos nossos sinceros agradecimentos a todos que contribuíram para o desenvolvimento deste projeto.

Ao nosso estimado orientador, professor Hugo Vieira Neto, por quem guardaremos sempre uma profunda admiração, pelo comprometimento e pelos sábios conselhos.

Às nossas famílias, especialmente aos nossos pais, pelo carinho e apoio incondicional ao longo de nossas vidas.

Aos nossos amigos, especialmente a Mariana Molenda Jambersi e a Alexandre Rosa Lemos, pelas conversas, risadas e suporte emocional.

Ao Lactec, por viabilizar o desenvolvimento deste projeto, e especialmente ao pesquisador Lourival Lippmann Junior pelos ensinamentos, pela confiança e por tornar esse projeto possível.

À empresa Emove, pela abertura e por permitir o uso do sistema de veículos elétricos em nosso projeto.

À empresa Radioenge, pelo fornecimento, de muito bom grado, de materiais para o desenvolvimento do trabalho.

A todos os professores do curso de Bacharelado em Engenharia Eletrônica da UTFPR, pela excelência do ensino que nos foi proporcionado.

A todos aqueles que participaram de nossa jornada até aqui.

RESUMO

SIMONI, Gabriel M.; SCARAMELLA, Geovana. **Monitoramento de veículos alternativos: desenvolvimento de um protótipo com tecnologia LoRaWAN**. 2021. 149 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Eletrônica) – Universidade Tecnológica Federal do Paraná. Curitiba, 2021.

Este trabalho propõe um sistema de monitoramento de veículos alternativos da frota corporativa particular do Lactec por meio de uma rede *LoRaWAN*. Essa tecnologia explora a faixa de frequências não licenciada e apresenta baixo consumo de energia, além de longo alcance, reduzindo os custos de operação de aplicações de rastreamento de veículos em uma área limitada. O dispositivo móvel acoplado aos veículos é baseado no *hardware open-source* Dragino LGT-92 enquanto os *gateways LoRaWAN* são compostos por um *Raspberry Pi* e um *shield* Radioenge RD43HATGPS. *Software open-source* disponibilizado pela *Semtech* e pela *Chirpstack Community* é utilizado na infraestrutura *LoRaWAN*. O sistema é uma alternativa viável para o monitoramento de patinetes e bicicletas elétricos, atendendo aos requisitos com baixo custo de operação em relação a outras tecnologias de rede e resolvendo problemas de falta de informação de veículos em percurso entre as sedes do Lactec. Os dados coletados permitem aos administradores do sistema de mobilidade averiguar o histórico dos veículos para extrair informações, como o uso feito por diferentes clientes, as rotas tomadas por eles, se houve quedas, o tempo de percurso entre as estações, entre outras. Com o registro das curvas de carga e descarga da bateria, será possível determinar se uma bateria ou veículo apresenta defeitos e, após a geração de uma base de dados pelo uso do sistema, até prevê-los para agendar manutenções preditivas. Por fim, para transformar o protótipo em um produto, melhorias no dispositivo de monitoramento e seu acoplamento com o veículo são necessárias.

Palavras-chave: LoRaWAN. Internet das coisas. Rastreamento. Meios de transporte alternativos. Mobilidade urbana.

ABSTRACT

SIMONI, Gabriel M.; SCARAMELLA, Geovana. **Alternative vehicles monitoring: development of a prototype with LoRaWAN technology**. 2021. 149 p. Bachelor Thesis (Bachelor's Degree in Electronic Engineering) – Universidade Tecnológica Federal do Paraná. Curitiba, 2021.

This paper proposes a system for monitoring alternative vehicles of Lactec's private corporate fleet through a LoRaWAN network. This technology exploits the unlicensed frequency band and presents low power consumption, besides long range, reducing operation costs of vehicle tracking applications in a limited area. The mobile device attached to the vehicles is based on Dragino LGT-92 open-source hardware while the LoRaWAN gateways are composed of a Raspberry Pi and a Radioenge RD43HATGPS shield. Open-source software provided by Semtech and the Chirpstack Community is used in the LoRaWAN infrastructure. The system is a viable alternative for monitoring electric scooters and bicycles, meeting the requirements with low operating costs compared to other network technologies and solving problems of lack of information of vehicles on the way between Lactec's headquarters. The data collected allows the mobility system administrators to investigate the history of the vehicles to extract information, such as the use made by different clients, the routes taken by them, if there were crashes, the travel time between stations, among others. With the registration of the battery charge and discharge curves, it will be possible to determine whether a battery or vehicle has defects and, after generating a database through the use of the system, even predict them in order to schedule predictive maintenance. Finally, to transform the prototype into a product, improvements in the monitoring device and its coupling with the vehicle are necessary.

Keywords: LoRaWAN. Internet of things. Tracking. Alternative means of transport. Urban mobility.

LISTA DE ILUSTRAÇÕES

Figura 1 – Disponibilidade de serviços de compartilhamento de bicicletas pelo mundo.	18
Figura 2 – Previsão de mercado global para sistemas de compartilhamento de bicicletas de 2019 a 2026.	19
Figura 3 – Classificação de aplicações IoT em relação à taxa de dados e ao uso (H2M e M2M).	20
Figura 4 – Comparação das tecnologias de transmissão IoT <i>LoRa</i> , <i>Sigfox</i> e NB-IoT . . .	21
Figura 5 – Camadas do padrão <i>LoRaWAN</i>	31
Figura 6 – Arquitetura da rede <i>LoRaWAN</i>	32
Figura 7 – Modulação <i>Direct Sequence Spread Spectrum</i> (DSSS).	34
Figura 8 – Sinais <i>upchirp</i> (esquerda) e <i>downchirp</i> (direita).	35
Figura 9 – Exemplo de símbolos <i>LoRa</i> com fator de espalhamento SF7.	36
Figura 10 – Diagrama de blocos da modulação e demodulação <i>LoRa</i>	36
Figura 11 – Frequências de canais do padrão AU915-928MHz	39
Figura 12 – Procedimento de requisição de <i>join</i>	44
Figura 13 – Procedimento de aceitação de <i>join</i>	44
Figura 14 – Chaves de criptografia na rede <i>LoRaWAN</i>	45
Figura 15 – Rotas obrigatórias para uso dos veículos alternativos do Lactec.	56
Figura 16 – Diagrama de bloco da arquitetura do sistema de monitoramento de veículos.	59
Figura 17 – Arquitetura da pilha Chirpstack.	59
Figura 18 – Módulo Dragino LGT92.	62
Figura 19 – Módulo DCDC <i>step down</i>	62
Figura 20 – Circuito DCDC <i>step down</i> fixo para 5V.	63
Figura 21 – Circuito de proteção de subtensão.	63
Figura 22 – Circuito de monitoramento de tensão da bateria do veículo e estação de carregamento.	64
Figura 23 – Circuito de acionamento do <i>buzzer</i> CEM1203-42.	65
Figura 24 – Circuito de alimentação, acionamento do <i>buzzer</i> e acoplamento com o ADC integrados na mesma PCB.	65
Figura 25 – Antena original do módulo LGT92-Li.	66
Figura 26 – Antena mais adequada para a associação mecânica com o veículo.	66
Figura 27 – Diagrama de estados do <i>firmware</i> do dispositivo de monitoramento.	67
Figura 28 – Eixos do acelerômetro no patinete.	69
Figura 29 – Pontos disponíveis para instalação dos <i>gateways LoRaWAN</i>	74
Figura 30 – Dados de RSSI coletados para avaliação da cobertura do <i>gateway LoRaWAN</i> instalado na cobertura do prédio sede do Lactec.	76
Figura 31 – Simulação da cobertura do <i>gateway LoRaWAN</i> instalado na cobertura do prédio sede do Lactec.	76
Figura 32 – Simulação da cobertura do <i>gateway LoRaWAN</i> instalado no telhado do prédio Leme do Lactec.	77
Figura 33 – Simulação da cobertura do <i>gateway LoRaWAN</i> instalado no telhado do prédio Cehpar do Lactec.	78
Figura 34 – Mapa de relevo da região de operação do sistema com destaque da localização para possível instalação dos <i>gateways LoRaWAN</i>	78

Figura 35 – Simulação da cobertura dos <i>gateways LoRaWAN</i> instalado no prédios sede e Cehpar.	79
Figura 36 – Dados de RSSI coletados para avaliação da cobertura do <i>gateway LoRaWAN</i> instalado no telhado do prédio Leme do Lactec.	79
Figura 37 – Dados de RSSI coletados para avaliação da cobertura dos <i>gateways LoRaWAN</i> instalados nos prédios sede e Leme do Lactec – patinete.	80
Figura 38 – Dados de RSSI coletados para avaliação da cobertura dos <i>gateways LoRaWAN</i> instalados nos prédios sede e Leme do Lactec – carro.	80
Figura 39 – Diagrama de blocos do servidor de aplicação.	81
Figura 40 – Diagrama de banco de dados do servidor de aplicação.	85
Figura 41 – Sequência de telas do processo de cadastro de dispositivos com auxílio do aplicativo de cadastro.	86
Figura 42 – Área legal de circulação dos veículos para o teste no estacionamento do Lactec.	88
Figura 43 – Resultados do teste de rastreamento no estacionamento – mapa.	89
Figura 44 – Resultados do teste de rastreamento no estacionamento – tabela.	89
Figura 45 – Resultados do teste de rastreamento no estacionamento – velocidade e detecção de movimento por acelerômetro.	90
Figura 46 – Resultados de verificação de veículo fora de área – teste no estacionamento.	90
Figura 47 – Área legal de circulação dos veículos do Lactec.	91
Figura 48 – Resultados de verificação de veículo fora de área – teste no estacionamento com área oficial.	91
Figura 49 – Resultados de rastreamento do veículo em percurso da Sede ao Leme, à esquerda, e do Leme à Sede, à direita.	93
Figura 50 – Resultados de identificação de veículo fora de área no percurso da Sede ao Leme, à esquerda, e do Leme à Sede, à direita.	94
Figura 51 – Notificações recebidas ao sair e retornar à área durante percurso com veículo nas rotas do Lactec.	94
Figura 52 – Resultados de detecção de movimento do veículo no percurso da Sede ao Leme, acima, e do Leme à sede, abaixo.	95
Figura 53 – Resultados de RSSI e DOP do veículo no percurso da Sede ao Leme, acima, e do Leme à sede, abaixo.	96
Figura 54 – Resultados dos testes de rastreamento do veículo em percurso entre as sedes do Lactec – tabela.	96
Figura 55 – Tela de acompanhamento do veículo em tempo real com possível acidente no percurso Leme-Sede.	97
Figura 56 – Notificação de possível acidente enviada durante teste no percurso Leme-Sede.	98
Figura 57 – Resultados de monitoramento da bateria do veículo no percurso da Sede ao Leme, acima, e do Leme à Sede, abaixo.	99
Figura 58 – Resultados de monitoramento da bateria do veículo em carregamento na estação Leme.	100
Figura 59 – Carregamento da bateria do veículo na estação.	100
Figura 60 – Configuração dos parâmetros do dispositivo de monitoramento por meio da API.	101
Figura 61 – Log do dispositivo durante a configuração remota utilizando a API.	102
Figura 62 – Reinicialização do dispositivo por meio de comando da API.	103
Figura 63 – Log do dispositivo ao reiniciar após o recebimento de um comando de <i>reset</i>	103
Figura 64 – Erro ao tentar configurar um parâmetro com um valor fora da faixa aceita.	104
Fotografia 1 – Estação de carregamento na sede do Lactec.	56

Fotografia 2 – Local escolhido para a instalação do dispositivo de monitoramento no patinete.	72
Fotografia 3 – Acoplamento entre o <i>hardware</i> e estrutura frontal do patinete.	72
Fotografia 4 – Conectores <i>micro fit</i> para conexão do dispositivo com o patinete.	73
Fotografia 5 – Conexão do dispositivo com o patinete.	73
Fotografia 6 – Gabinete para o <i>gateway</i> LoRaWAN.	75
Gráfico 1 – Aceleração em cada eixo com o veículo em movimento e parado.	70
Gráfico 2 – Distribuição normal das acelerações com o veículo parado.	70
Gráfico 3 – Distribuição normal das acelerações com o veículo em movimento.	70
Quadro 1 – SNR necessário para a demodulação em função do SF escolhido.	38
Quadro 2 – Planos de frequência - AU915-928MHz	39
Quadro 3 – Configurações de data rate- AU915-928MHz	40
Quadro 4 – Planos de frequência - AU915-928MHz	40
Quadro 5 – Formatação de <i>payload Cayenne LPP</i>	46
Quadro 6 – Tipo de dado <i>Cayenne LPP</i>	47
Quadro 7 – Formatação NMEA.	48
Quadro 8 – Mensagens NMEA para GPS (QUECTEL, 2016)	49
Quadro 9 – Mensagens de <i>uplink</i>	60
Quadro 10 – Mensagens de <i>downlink</i>	61
Quadro 11 – Comparação de modelos de <i>buzzer</i>	64
Quadro 12 – Parâmetros de configuração e estado da aplicação do dispositivo de monitoramento salvos na EEPROM.	67
Quadro 13 – Descrição dos alertas sonoros e condições de acionamento/desacionamento.	68

LISTA DE ABREVIATURAS, SIGLAS E ACRÔNIMOS

SIGLAS

ABP	<i>Activation by Personalization</i>
ADR	<i>Adaptive Data Rate</i>
API	<i>Application Programming Interface</i>
AWGN	<i>Additive White Gaussian Noise</i>
CRC	<i>Cyclic Redundancy Check</i>
CSS	<i>Chirp Spread Spectrum</i>
DOP	<i>Dilution of Precision</i>
DSSS	<i>Direct Sequence Spread Spectrum</i>
EIRP	<i>Effective Isotropic Radiated Power</i>
EUI	<i>Extended Unique Identifier</i>
GPS	<i>Global Positioning System</i>
H2M	<i>Human-to-Machine</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IEEE	<i>Institute of Electrical and Electronic Engineers</i>
IoT	<i>Internet of Things</i>
IP	<i>Internet Protocol</i>
ISM	<i>Industrial Scientific and Medical</i>
JSON	<i>JavaScript Object Notation</i>
LPWAN	<i>Low-Power Wide-Area Network</i>
LTE	<i>Long Term Evolution</i>
M2M	<i>Machine-to-Machine</i>
MAC	<i>Media Access Control</i>
ORM	<i>Object Relational Mapping</i>
OTAA	<i>Over-The-Air Activation</i>
QoS	<i>Quality of Service</i>
RSSI	<i>Received Signal Strength Indicator</i>
SaaS	<i>Software as a Service</i>
SF	<i>Spreading Factor</i>
SNR	<i>Signal-to-Noise Ratio</i>
SQL	<i>Structured Query Language</i>
SS	<i>Spread Spectrum</i>
UNB	<i>Ultra Narrow Band</i>
URI	<i>Uniform Resource Identifier</i>

UWB

Ultra Wide Band

SUMÁRIO

1	INTRODUÇÃO	14
1.1	OBJETIVOS	15
1.1.1	Objetivos gerais	15
1.1.2	Objetivos específicos	15
1.2	ESTRUTURA DO TRABALHO	16
2	REVISÃO DA LITERATURA	17
2.1	ESTADO DA ARTE	17
2.1.1	Sistemas de comunicação para monitoramento de veículos	19
2.1.2	Trabalhos relacionados	23
2.1.3	Geolocalização de dispositivos móveis	26
2.1.4	Soluções comerciais	27
2.2	LORAWAN	30
2.2.1	Arquitetura de rede	31
2.2.2	LoRa	31
2.2.3	Parâmetros regionais	38
2.2.4	End devices	40
2.2.5	Gateway	41
2.2.6	Network Server	42
2.2.7	Join Server	43
2.2.8	Application Server	43
2.2.9	Device Commissioning	43
2.2.10	Join Procedure	44
2.2.11	Classes de dispositivos	45
2.3	CAYENNE LPP	46
2.4	GPS	47
2.4.1	Protocolo NMEA	48
2.5	ACELERÔMETRO	49
2.6	NODEJS	51
2.7	BANCO DE DADOS	51
2.8	API RESTFUL	53
3	DESENVOLVIMENTO DO SISTEMA LORAWAN DE MONITORAMENTO DA FROTA DE VEÍCULOS DO LACTEC	55
3.1	SISTEMA ATUAL DE VEÍCULOS DO LACTEC E PROPOSTAS DE MELHORIAS	55
3.2	ARQUITETURA DO SISTEMA	58
3.3	PROTOCOLO	60
3.4	DISPOSITIVO DE MONITORAMENTO	61
3.4.1	Adaptação do <i>hardware</i>	61
3.4.2	<i>Firmware</i>	67
3.4.3	Acoplamento	71
3.5	GATEWAYS	74
3.6	SERVIDOR DE APLICAÇÃO PIOTNET	81

4	TESTES E RESULTADOS DO SISTEMA DE MONITORAMENTO DOS VEÍCULOS	87
4.1	MONITORAMENTO DE VEÍCULOS EM PERCURSO	87
4.2	MONITORAMENTO DE CARGA E DESCARGA DA BATERIA DOS VEÍCULOS	99
4.3	ENVIO DE <i>DOWNLINKS</i> DE CONFIGURAÇÃO	99
5	CONCLUSÕES E PERSPECTIVAS	105
	REFERÊNCIAS	108
	APÊNDICES	114
	APÊNDICE A – ESPECIFICAÇÃO DE PROBLEMAS, NECESSIDADES E REQUISITOS	115
	APÊNDICE B – DIAGRAMAS DE FUNCIONAMENTO DO <i>FIRMWARE</i> DO DISPOSITIVO DE MONITORAMENTO	131
	APÊNDICE C – DIAGRAMA DE ATIVIDADES DO BLOCO "LÓGICA DO SISTEMA" DO SERVIDOR DE APLICAÇÃO	137
	APÊNDICE D – DIAGRAMA DE ATIVIDADES PARA TRATAMENTO DE DADOS DE PERCURSO	138
	APÊNDICE E – DOCUMENTAÇÃO - API	139
	ANEXO	146
	ANEXO A – SIMULAÇÃO DE COBERTURA DOS <i>GATEWAYS</i> <i>LO-RWAN</i> POR MEIO DA FERRAMENTA <i>VE2DBE</i>	147

1 INTRODUÇÃO

O sistema de transporte baseado majoritariamente em automóveis possui muitos impactos negativos como, por exemplo, a poluição atmosférica e sonora, elevada taxa de acidentes e estresse psicológico. Motivadas a amenizar estes problemas, diversas iniciativas têm surgido com o propósito de oferecer uma alternativa de transporte individual. Entre estas, a implantação de sistemas de compartilhamento de patinetes elétricos e bicicletas, em especial, nas regiões centrais de grandes cidades. No Lactec, um destes sistemas foi inaugurado em 2019 para o uso corporativo, permitindo aos colaboradores se locomoverem entre as três instalações da empresa em Curitiba (Sede, Leme e Cehpar).

Para gerenciamento do sistema de mobilidade pelos responsáveis conforme definido na norma do Lactec, é necessária uma base de informações referentes ao funcionamento dos veículos e o uso deles pelos funcionários. No entanto, não existia nenhuma informação gerada durante percursos com os veículos. Além disso, nenhum dado era salvo em relação às baterias dos mesmos. Desta forma a manutenção era exclusivamente corretiva. Por fim, as estatísticas de uso dos patinetes eram limitadas àquelas geradas pela empresa fornecedora do sistema, a Emove. Assim, era impossível consultar informações mais específicas, como quem usou os veículos em determinados dias, como foram os percursos tomados pelos usuários, as velocidades com que eles conduziam, o uso da bateria durante o percurso, se ocorreram acidentes com queda do veículo, entre outras. Para solucionar esses problemas, o sistema de rastreamento apresentado neste trabalho foi proposto. Como o Lactec é um instituto de pesquisa e desenvolvimento, com diversos projetos na área de IoT, a concepção de uma rede de monitoramento *LoRaWAN* se torna ainda mais interessante para o estudo da tecnologia e a sua utilização para outros propósitos.

Para desenvolver o sistema de monitoramento, as diferentes possibilidades de tecnologias de rede foram analisadas e decidiu-se pelo uso de uma rede *LoRaWAN*. Uma especificação de requisitos técnicos foi elaborada para resolver o problema em questão. Para a implantação da rede *LoRaWAN*, foram montados *gateways* com a placa *Raspberry Pi* e o *shield* Radioenge RD43HATGPS e uma máquina virtual Ubuntu foi criada para executar os serviços Chirpstack, uma pilha *LoRaWAN open-source*. Após simulações de enlace de rádio, um *gateway* foi instalado no Lactec Sede e outro no Lactec Leme. Os dispositivos de monitoramento acoplados aos veículos foram baseados no *hardware open-source* Dragino LGT92, uma solução *LoRaWAN* para rastreamento. Modificações de *hardware* foram necessárias para a integração, bem como

o desenvolvimento de um novo *firmware*. Uma estrutura de banco de dados relacional e um servidor de aplicação foram implementados para atender aos requisitos. O acesso aos recursos do sistema e aos dados dos veículos é possibilitado por uma API do tipo *RESTful* e um servidor *WebSocket* para transmissão dos dados em tempo real.

O projeto desenvolvido atestou a viabilidade do uso do *LoRaWAN* no rastreamento de veículos alternativos, com infraestrutura de rede própria, em uma região espacial limitada. A grande maioria dos requisitos propostos foi totalmente satisfeita. Com o *hardware* atual e seu posicionamento no veículo, o GPS sofre falha ocasionais, causando perda de dados em alguns trechos do percurso. A cobertura *LoRaWAN*, apesar de apresentar resultados satisfatórios no envio de dados do dispositivo de monitoramento para os *gateways* nas rotas estabelecidas, exibiu algumas instabilidades ao redor delas e, principalmente, na recepção de *downlinks* pelos dispositivos. Um novo desenho de placa de circuito impresso deverá ser desenvolvido, com os principais componentes utilizados no projeto, integrando os circuitos de alimentação e o dispositivo IoT em um design mais compacto para ser instalado no interior do veículo de forma ótima, com possíveis alterações nas antenas de GPS e *LoRa*.

1.1 OBJETIVOS

1.1.1 Objetivos gerais

Este trabalho tem por objetivo desenvolver uma solução de monitoramento de veículos alternativos, operando em região limitada, por meio da tecnologia *LoRaWAN*. Foram desenvolvidos o protótipo de dispositivo de monitoramento a ser acoplado ao veículo, o servidor de aplicação e a API para consulta dos dados coletados.

1.1.2 Objetivos específicos

- Desenvolver o *hardware* e o *firmware* do dispositivo de monitoramento embarcado para a coleta e envio de informações a respeito do veículo.
- Implantar a infraestrutura de rede *LoRaWAN* no Lactec.
- Desenvolver uma estrutura de banco de dados para armazenamento das informações dos veículos e da aplicação de monitoramento

- Desenvolver um servidor de aplicação capaz de se comunicar com a rede *LoRaWAN*, armazenar as informações dos veículos e executar a lógica do sistema.
- Desenvolver uma API que permita a outros desenvolvedores utilizarem os recursos do sistema para implementar aplicações.
- Desenvolver um *software* para automatizar o cadastro de novos veículos na rede de monitoramento.

1.2 ESTRUTURA DO TRABALHO

No Capítulo 2 é apresentada uma revisão da literatura, incluindo o estado da arte e explicações sobre as tecnologias utilizadas neste projeto. O processo de desenvolvimento do sistema é descrito no Capítulo 3. Os testes realizados com o sistema e os resultados obtidos são contemplados no Capítulo 4. Por fim, no Capítulo 5 são apresentadas as melhorias projetadas para o sistema e as considerações finais sobre o trabalho. No texto, alguns termos foram mantidos no idioma original, em inglês, principalmente relativos à tecnologia de rede *LoRaWAN*, já que a tradução dos mesmos poderia prejudicar o significado específico. Os autores ressaltam que, apesar disso, consideram importante a utilização de termos técnicos em português sempre que o entendimento não seja prejudicado.

2 REVISÃO DA LITERATURA

2.1 ESTADO DA ARTE

O crescimento acelerado das cidades devido, principalmente, ao rápido aumento populacional, não tem sido capaz de promover de maneira satisfatória, ao mesmo passo, soluções para o deslocamento de pessoas e recursos. Os impactos negativos do transporte baseado majoritariamente em automóveis há tempo são conhecidos. Entre estes, pode-se destacar a poluição atmosférica e sonora, alta taxa de acidentes automotivos e o estresse psicológico como consequência do sistema de tráfego ineficiente (OLIVEIRA *et al.*, 2021).

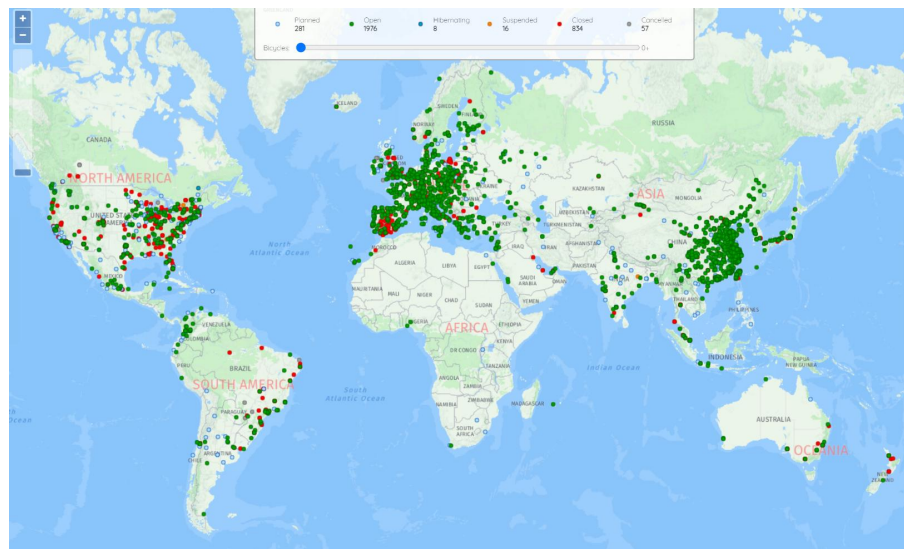
Estudos mais antigos defendiam o planejamento de cidades descentralizadas e com uma densidade populacional reduzida como a principal forma de proporcionar um ambiente urbano saudável. Neste cenário, a longa distância entre casa, trabalho, escola e outros locais importantes tornava o carro quase que uma obrigatoriedade. O planejamento, por si só, não foi capaz de impedir a degradação do sistema de transporte à medida que a população ampliou. Por isso, pesquisas mais recentes estão se voltando para a ideia das cidades compactadas. Estas têm como ideal a diminuição das distâncias que seus habitantes necessitam percorrer diariamente, possibilitando o uso de veículos alternativos ou até mesmo a possibilidade de caminhar até o local. Este deslocamento dito “ativo”, além de diminuir os efeitos causados pelo uso dos carros, traz benefícios diretos à saúde da população, dado que, na maioria dos países, existe uma relação inversa entre o uso de transporte ativo e os índices de obesidade (NEWMAN; MATAN, 2012).

Este cenário tem se mostrado propício para o surgimento de iniciativas, em especial, tecnológicas, que facilitem a adoção de meios de transporte alternativos, como bicicletas e patinetes, ou que também incentivem a utilização do transporte público. Em grandes cidades, sistemas de compartilhamento de veículos têm sido implantados em regiões centrais que possuem essa característica de compactação (NEWMAN; MATAN, 2012).

O projeto *Witte Fietsenplan* (plano das bicicletas brancas), desenvolvido em 1960 em Amsterdam, é considerado o primeiro sistema de compartilhamento de bicicletas da história. A medida tinha um viés ambiental, visando diminuir a utilização de carros por meio da disponibilização de bicicletas sem custo de utilização. O projeto não teve sucesso por conta de alguns fatores, tais como o vandalismo. Apesar disso, serviu como inspiração para o posterior surgimento de outros sistemas (ZEE, 2016).

Nos últimos anos, o número destas iniciativas tem aumentado rapidamente. Em 2013, havia 535 serviços de compartilhamento de veículos, distribuídos em 49 países com uma frota de 500 mil bicicletas. Em 2018, esses números aumentaram para mais de 1600 serviços que juntos disponibilizavam mais de 18 milhões de veículos (CROCE *et al.*, 2019). Pelo mapa de distribuição destes sistemas, apresentado na Figura 1, verifica-se que este modelo de micromobilidade é especialmente popular na Europa e na China.

Figura 1 – Disponibilidade de serviços de compartilhamento de bicicletas pelo mundo.



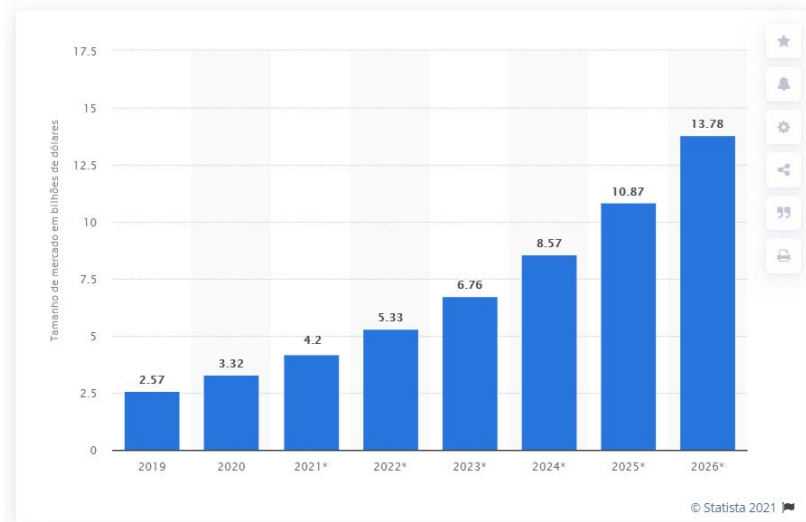
Fonte: PBSC (2021).

No Brasil, em 2019, as bicicletas e patinetes da empresa brasileira *Yellow* se tornaram populares nas principais capitais. O grande sucesso levou à fusão com a mexicana *Grin*, formando a *Grow*. Infelizmente, no atual momento, a empresa vem enfrentando dificuldades e tem anunciado fim da operação em vários locais, apontando a crise causada pela pandemia da COVID-19 como a principal causa da instabilidade financeira. Apesar disso, a empresa pretende continuar a investir na ideia futuramente, focando agora no aluguel de patinetes (RIVEIRA *et al.*, 2020).

No ano de 2020 houve uma baixa na utilização de sistemas de transporte em geral pela população, que tem evitado sair de casa devido à pandemia. Apesar disso, a utilização de bicicletas se popularizou no mundo todo como meio de transporte seguro em meio à crise sanitária, visto que por ser um transporte individual, auxilia nas questões de distanciamento e higiene (MENGUE, 2021). Isto posto, ainda é esperado que este mercado continue crescendo, atingindo um valor de 13,78 bilhões de dólares em 2026 (Figura 2), segundo o Statista (2021).

Figura 2 – Previsão de mercado global para sistemas de compartilhamento de bicicletas de 2019 a 2026.

Mercado global de serviços de compartilhamento de bicicleta entre 2020 e 2026
(em bilhões de dólares)



Fonte: Adaptado de Statista (2021).

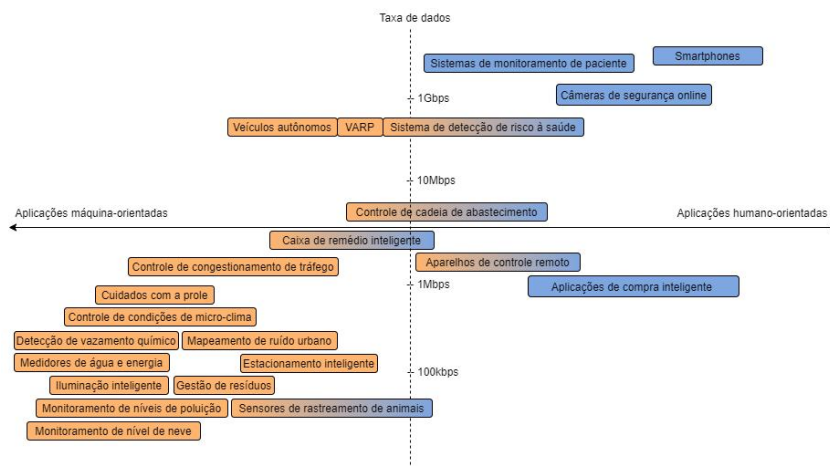
De modo geral, os sistemas de compartilhamento de veículos, podem ser divididos em duas categorias: *docked* e *dockless*. No primeiro, os veículos são retirados e devolvidos em estações de localização fixa instaladas na cidade, enquanto no segundo, não há um local designado para este fim. Independentemente do modelo de negócio, estas iniciativas ainda sofrem com questões de logística (redistribuição dos veículos em pontos de interesse), bem como em assegurar o uso correto e integridade física dos veículos. Desta forma, necessitam de uma forma de monitoramento eficiente e economicamente viável para melhorar o serviço oferecido. Esse monitoramento consiste, em geral, na coleta de dados de geolocalização e, em alguns casos, como no de veículos elétricos, informações a respeito do estado da bateria (CROCE *et al.*, 2019).

2.1.1 Sistemas de comunicação para monitoramento de veículos

É esperado que até o fim de 2021, pelo menos 25 bilhões de dispositivos estejam conectados à internet devido ao crescimento significativo de aplicações IoT (GARTNER, 2018). A variedade destas impulsionou o surgimento de novas tecnologias de comunicação, cada qual com suas vantagens e desvantagens que precisam ser analisadas para que se escolha adequadamente entre elas. Neste contexto, várias delas se mostram aplicáveis para o monitoramento de veículos alternativos. Portanto, nesta seção serão apresentadas as características das diversas tecnologias que podem ser empregadas neste problema para justificar a escolha pelo padrão *LoRaWAN*.

Em redes IoT, que requerem ampla cobertura e dispositivos com baixo consumo de energia, algumas das tecnologias largamente utilizadas em aplicações *Human-to-Machine* (H2M) podem ser descartadas. Isto porque, em geral, apresentam alto consumo ou baixo alcance, como é o caso do *Bluetooth*, *Wi-Fi* e 3G/4G (DING *et al.*, 2020). Alguns exemplos de classificação de aplicações entre H2M e *Machine-to-Machine* (M2M) em relação à taxa de dados necessária pode ser observada na Figura 3.

Figura 3 – Classificação de aplicações IoT em relação à taxa de dados e ao uso (H2M e M2M).



Fonte: Adaptado de Ding *et al.* (2020)

Boa parte das aplicações M2M requerem taxa de transmissão de dados relativamente baixa e a partir desta margem, foram desenvolvidas tecnologias que priorizam menor consumo e maior alcance. Nesse cenário, surgiu o termo *Low-Power Wide-Area Networks* (LPWANs) para classificar esse tipo de comunicação.

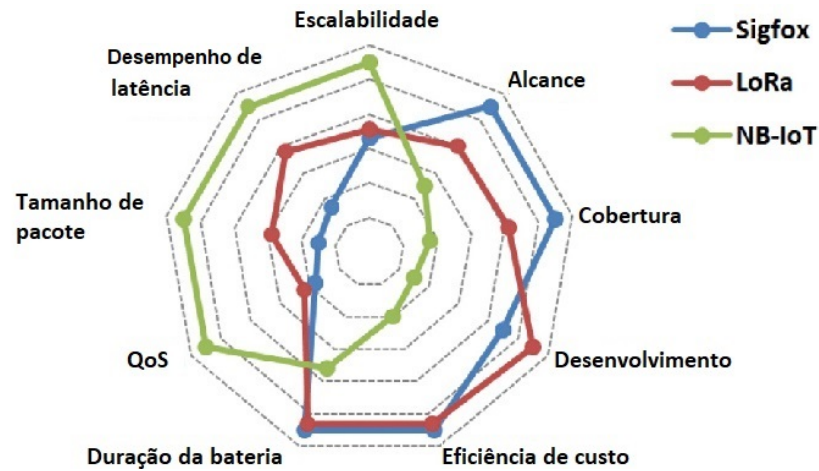
As principais LPWANs existentes no mercado são *LoRaWAN*, *Sigfox* e *NB-IoT*. *Sigfox*, da empresa homônima, segue o modelo *Software as a Service* (SaS), com um custo mensal para uso da rede. *NB-IoT* é integrada à rede *Long Term Evolution* (LTE). Portanto, sua operação utiliza banda licenciada e depende de um contrato com operadoras de telefonia.

Comparação entre *Sigfox* e *LoRaWAN*

Dentro das LPWANs, *LoRaWAN* e *Sigfox* se encontram na categoria de padrões que fazem uso da faixa de frequências não licenciadas (ISM). Esta escolha tem como vantagem a diminuição dos custos de desenvolvimento e de operação de ambos sistemas (DING *et al.*, 2020).

Quanto à modulação, as tecnologias empregam abordagens opostas no uso de banda. O

Figura 4 – Comparação das tecnologias de transmissão IoT *LoRa*, *Sigfox* e NB-IoT



Sigfox trabalha com modulação GFSK (*Gaussian Frequency Shift Keying*)/DBPSK (*Differential Binary Phase Shift Keying*) em *Ultra Narrow Band* (UNB) com canais de 100Hz (MEKKI *et al.*, 2019), o *LoRa*, CSS em *Ultra Wide Band* (UWB) com canais de 125kHz e 500kHz. Naturalmente, essa diferença confere ganhos e perdas a cada uma das técnicas.

O *Sigfox* apresenta vantagens a respeito do alcance e da cobertura, como mostra a Figura 4, variando de 10km (urbano) a 40km (rural), enquanto o *LoRa* atinge de 5km (urbano) a 20 km (rural) (MEKKI *et al.*, 2019). Em aplicações com dispositivos móveis, a comunicação pode sofrer instabilidades devido ao efeito *Doppler*. Neste cenário, o *LoRa* se sai melhor em comparação com as tecnologias UNB, podendo suportar velocidades de até 40km/h, enquanto o *Sigfox* perde estabilidade com velocidades por volta de 5 a 10 km/h (DMITRIEVICH; DMITRIEVICH, 2019). Ademais, a taxa de transmissão de dados do *LoRa*, de 240bps a 37,5kbps (SEMTECH, 2019b), supera o valor possível, de 100bps a 600bps, para o *Sigfox* (SIGFOX, 2017).

Além de diferenças na tecnologia de transmissão, os padrões apresentam filosofias de negócio bem distintas. O *Sigfox* possui uma estratégia voltada para *software* e serviço. Toda a infraestrutura de rede é gerenciada pela própria empresa, não sendo possível o desenvolvimento de redes privadas ou a instalação de concentradores em áreas com cobertura inexistente, ou insatisfatória, por iniciativa do usuário. Desta forma, apesar de trabalhar na banda ISM, livre de custos, o *Sigfox* requer a contratação de um serviço anual, por dispositivo, para a utilização da rede. O *LoRaWAN*, por sua vez, oferece uma estrutura de rede mais flexível. As especificações de protocolo e o *software* para *gateways* e nós são disponibilizados gratuitamente pelas próprias

Semtech e LoRa Alliance. Isso dá ao usuário a opção de montar uma rede totalmente privada, sem custos de serviço, desenvolvida a partir da documentação ou das diversas opções comerciais disponíveis (Subseção 2.1.4).

Em relação aos nós, diferentemente do *Sigfox* que utiliza uma modulação bem difundida, o que permite que seus *end devices* sejam confeccionados por diferentes companhias, o *LoRaWAN* exige o uso da modulação proprietária *LoRa*. Até pouco tempo, somente a *Semtech* podia produzir circuitos integrados capazes de utilizar a modulação *LoRa*. No entanto, recentemente, a *STMicroelectronics* passou a produzir *SoCs* com a tecnologia, como o *STM32WL*.

Assim, cada dispositivo *LoRaWAN* requer um *chip* da *Semtech*, atualmente a única fabricante de rádios *LoRa*.

Por fim, existem limitações de *uplink* que podem inviabilizar o uso do *Sigfox* em algumas aplicações. No plano *platinum*, cada nó conta com até 140 *uplinks* e 7 *downlinks* por dia e cada mensagem carrega no máximo 12 *bytes* (SIGFOX, 2017). A disponibilidade de *uplinks* para o *LoRaWAN* varia conforme a região de operação por conta da legislação vigente para rádios operando na banda ISM. Na região AU915-928, utilizada no Brasil, um pacote pode conter até 242 *bytes* de aplicação (ALLIANCE, 2018).

Comparação entre *NB-IoT* e *LoRaWAN*

Diferentemente do *LoRaWAN* e do *Sigfox*, o padrão *NB-IoT* é utilizado em faixa licenciada, por fazer parte do padrão LTE, desde o *Release 13* feito pela 3GPP. Apesar disso, pode ser considerado uma nova forma de comunicação pelo ar. Para reduzir o consumo de energia elétrica e os custos dos dispositivos, vários recursos do LTE foram removidos (SINHA *et al.*, 2017). Por apresentar banda estreita, permitindo uma maior densidade de potência, a *NB-IoT* apresenta uma maior penetração em construções do que outras tecnologias e, aliando isso à baixa taxa de dados, apresenta uma maior cobertura em relação ao LTE (DING *et al.*, 2020). A rede *LoRaWAN* apresenta vantagem em relação ao consumo energético, capacidade da rede e custo de implantação, enquanto a *NB-IoT* tem uma melhor métrica de qualidade de serviço (QoS), menor latência e maior taxa de dados (MEKKI *et al.*, 2019).

O maior tempo de vida da bateria dos dispositivos *LoRaWAN* se deve à possibilidade de entrarem em modo de baixo consumo de energia em qualquer momento, por utilizarem um método de comunicação assíncrono baseado no protocolo ALOHA. Do outro lado, o padrão *NB-IoT* necessita de sincronização regular. Além disso, as técnicas OFDM (*Orthogonal Frequency*

Division Multiplexing) e FDMA (*Frequency Division Multiple Access*) requerem uma maior corrente de pico para o transmissor. Dessa forma, a tecnologia NB-IoT atinge menor latência e maior taxa de transmissão mas, em troca, apresenta um consumo energético mais elevado (SINHA *et al.*, 2017).

Uma outra questão a ser levada em conta no desenvolvimento de novos projetos é a facilidade de implantação da infraestrutura de rede em diferentes localidades. Neste quesito, como o NB-IoT é limitado a se comunicar com estações LTE, não é aplicável para áreas rurais ou suburbanas que não possuam essa cobertura. Como se trata de uma banda licenciada, a implantação de uma infraestrutura própria acaba sendo inviável e, sendo assim, a possibilidade de se implantar projetos com NB-IoT depende da disponibilidade do serviço pelas operadoras de telefonia. Por outro lado, a rede *LoRaWAN*, além de utilizar a banda ISM, possui *gateways* de baixo custo, facilitando a implantação em qualquer local (SINHA *et al.*, 2017).

Pelo que foi destacado nesta seção, a tecnologia NB-IoT se destina a aplicações que precisem de uma baixa latência e alta taxa de dados, com maior QoS, sacrificando um pouco o consumo energético, que ainda é baixo o suficiente para classificá-la como uma LPWAN. Como a taxa de dados e a latência são críticos para o monitoramento dos veículos do Lactec, o uso de uma rede *LoRaWAN* parece ser mais adequado ao projeto. Além disso, por se tratar de uma empresa com foco na pesquisa e desenvolvimento de projetos, a implementação de uma rede *LoRaWAN* apresenta uma oportunidade maior para o futuro, por oferecer mais flexibilidade, possibilidade de expansão do número de dispositivos sem custos e implantação em áreas rurais e suburbanas. Por não depender da contratação de uma operadora, a rede *LoRaWAN* apresenta um custo de manutenção menor com o aumento do número de dispositivos conectados, apesar de um investimento inicial maior se considerada a implantação da rede contra a simples contratação de um plano. Sendo assim, as vantagens oferecidas pelo *LoRaWAN* são um chamativo para a adoção da tecnologia pelas empresas de aluguel de veículos alternativos que, em geral, apresentam uma grande frota.

2.1.2 Trabalhos relacionados

Visando suprir as necessidades dos sistemas de compartilhamento de veículos e, ao mesmo tempo, usufruir das vantagens do *LoRaWAN*, apresentadas na Subseção 2.1.1, diversos estudos foram realizados de modo a atestar a viabilidade de sua utilização em aplicações de rastreamento.

Adelantado *et al.* (2017) apresentaram uma abordagem teórica para determinar as limitações das redes *LoRaWAN* quanto à capacidade da rede, em quantidade de dispositivos. Para isso, consideram a rede *LoRaWAN* uma superposição de redes baseadas no protocolo ALOHA – uma para cada par formado por canal e fator de espalhamento (SF). Foi mostrado que a probabilidade de uso de taxas de transmissão mais baixas é maior, aumentando o tempo médio no ar dos pacotes *LoRa*. Como a banda não licenciada impõe, na maioria das regiões, limitação de ciclo de trabalho, isto é, da razão de tempo de ocupação do canal pelo dispositivo, a quantidade de pacotes enviados pelos dispositivos nessas configurações é bastante limitada. Para calcular a taxa de sucesso na entrega de pacotes em redes mais ocupadas, os autores consideraram uma quantidade variável de dispositivos transmitindo pacotes em três canais. A vazão dos dados é limitada pelas colisões e, portanto, diminui com o aumento do número de dispositivos na rede. Se a quantidade de dispositivos é pequena, a vazão dos dados é limitada pela regulação do ciclo de trabalho. A probabilidade de recepção dos pacotes pelo *gateway* diminui com o aumento da quantidade de pacotes por hora por dispositivo e vale entre 10% e 14% no ponto de máxima vazão de dados. Essa taxa pode ser melhorada pela adição de mais *gateways* na mesma rede, controlada pelo mesmo *Network Server*, aumentando as chances de que um *gateway* receba as mensagens com sucesso (VOIGT *et al.*, 2016).

Aftab *et al.* (2019) trataram de um cenário com sobreposição de redes, ou seja, de proximidade entre múltiplos *gateways*, controlados por diferentes servidores de rede. Para diferentes quantidades de dispositivos distribuídos em um raio de 12km de distância de seus respectivos *gateways*, foi mostrado que a probabilidade de cobertura de um dispositivo decai exponencialmente com a quantidade de redes sobrepostas. Para solucionar este problema, Voigt *et al.* (2016) propuseram dois métodos. O primeiro, já citado, consiste em aumentar a quantidade de *gateways* de uma mesma rede. O segundo aumenta a intensidade do sinal recebido pelos *gateways* sem aumentar a potência de transmissão, por meio do uso de antenas direcionais. Realizando simulações, foi mostrado que o aumento no número de *gateways* é mais efetivo que o uso de antenas direcionais.

Carvalho *et al.* (2018) avaliaram o atraso da comunicação *LoRaWAN*, dos nós até a aplicação, considerando nós móveis em uma grande área. O atraso entre o envio pelo dispositivo e a recepção pelo *Network Server* também foi avaliado. Os horários de todos os dispositivos da rede foram sincronizados em UTC (*Coordinated Universal Time*) por meio do NTP (*Network Time Protocol*). De acordo com o autor, as incertezas neste sincronismo, devido à qualidade da

conexão com a internet, está entre 0,2 ms e 7 ms. A incerteza na ação de tomar uma estampa de tempo é de 2,6ms. Em um carro andando a uma média de 17 km/h em um trajeto de 4km, foi acoplado um dispositivo configurado com o maior taxa de dados (e menor alcance, com SF = 7) da comunicação *LoRa*, diminuindo o tempo no ar dos pacotes *LoRaWAN*, 57 ms em média para os pacotes considerados no artigo. Como tinha uma grande quantidade de *gateways* na região de teste, nenhum pacote foi perdido. Chegou-se à conclusão que o atraso entre o envio da mensagem pelo *end device* e o recebimento pelo *Network Server*, em maioria devido à comunicação *LoRa*, teve média de 250 ms. O atraso total, até a aplicação, foi de 300ms, ou seja, satisfatório para aplicações não críticas de rastreamento de dispositivos móveis, como é o caso tratado neste trabalho.

Kim *et al.* (2017) analisaram a taxa de sucesso de pacotes *LoRa* com nós móveis distantes de 500m a 3km do *gateway*, em ambiente urbano sem linha de visada entre os rádios, em três diferentes modos de transmissão. Do primeiro até o terceiro, aumenta-se o alcance e diminui-se a taxa de dados. A taxa de sucesso caiu muito rapidamente com a mais alta taxa de transmissão dados. Como o comportamento da taxa de sucesso dos dois outros modos foi muito similar, concluiu-se que o ambiente afeta significativamente a comunicação. Em especial, à distância de 1,5km, acontece uma queda na taxa de sucesso para os dois casos, devido à diminuição da SNR. Com a mais baixa taxa de transmissão, a taxa de sucesso foi próxima de 100%, excetuando-se nos casos de 1,5km (com a queda do SNR) e 3km de distância. Com linha de visada, a taxa é de 100% mesmo com o nó a 3km de distância do *gateway*.

Petäjäljärvi *et al.* (2015) também avaliaram, em ambiente urbano, a taxa de perda de pacotes utilizando *LoRa* com a menor taxa de dados, para cinco zonas de distância – entre elas, a região compreendida a um raio de 0 a 2 km do *gateway* e a um raio de 2 a 5 km. A perda de pacotes, nestas duas zonas, foi de 12% e 15%, respectivamente, as quais os autores atribuíram a bloqueios da linha de visada por obstáculos ou a interferências por outros sistemas de rádio, já que a potência do sinal dos pacotes recebidos em maioria excedia -100dBm (bem acima da sensibilidade dos rádios *LoRa*).

Baharudin e Yan (2016) concluíram, experimentalmente, que o comportamento do *Received Signal Strength Indicator* (RSSI) em relação à distância, do sinal recebido pelo *gateway*, piora com a existência de múltiplos nós na rede. Os autores atribuem isso à interferência entre os sinais de múltiplos nós vizinhos. No entanto, concluem que a diferença não é significativa devido à alta sensibilidade dos rádios *LoRa*.

2.1.3 Geolocalização de dispositivos móveis

Atualmente, a grande maioria das aplicações que utilizam geolocalização de dispositivos o faz por meio do Sistema de Posicionamento Global GPS. Com o surgimento da IoT e a diversidade de aplicações que possuem este requisito, surgem novas formas de rastrear a posição de objetos. Em especial, uma abordagem que aparece para os sistemas de IoT é a utilização da trilateração ou da multilateração do sinal de rádio das LPWANs para determinar a localização de um nó da rede. A primeira requer o conhecimento da distância entre o transmissor e o receptor (*end device* e *gateway* no caso das LPWANs), que pode ser obtida pelo TOA (*Time Of Arrival*), pelo TOF (*Time Of Flight*) ou pelo RSSI. A segunda utiliza o TDOA (*Time Difference of Arrival*). Com o valor desses parâmetros em pelo menos três *gateways* em relação a um mesmo nó, é possível calcular a posição deste. Essas técnicas são as mesmas utilizadas no sistema de GPS, com a diferença que os cálculos são feitos no dispositivo enquanto, no caso da rede *LoRaWAN*, estes são feitos do lado do servidor (FARGAS; PETERSEN, 2017).

Além da óbvia vantagem de redução dos custos dos dispositivos por não necessitar de uma outra tecnologia para geolocalização (GHADIRZADEH, 2017), o uso da técnica de multilateração do sinal *LoRa* tenta resolver outras limitações do GPS. Uma delas é a impossibilidade de ser utilizado em ambientes extremos, como embaixo d'água, embaixo do solo ou com um grande obstáculo à frente do dispositivo, como uma montanha. A outra é o alto consumo de dispositivos GPS, incompatível com sistemas de IoT cujo tempo de vida das baterias deve ser de alguns anos (MACKEY; SPACHOS, 2019). Como neste projeto não há uma restrição tão grande quanto ao consumo, já que os veículos estão sempre recarregando as baterias quando acoplados às estações, e também não será implantado em um ambiente que exija um sistema mais robusto do que o GPS, essas duas vantagens da utilização da multilateração de sinais *LoRa* não são tão significativas. Outras características dessa técnica devem ser, portanto, analisadas.

Vários trabalhos anteriores demonstram que o uso da multilateração de sinais *LoRa* para geolocalização encontra-se em estágio de desenvolvimento, sendo possível o uso para sistemas que não necessitem tanta exatidão ou acurácia. Como destacam Fargas e Petersen (2017) e Ghadirzadeh (2017), para um sistema de rastreamento em tempo real, essa técnica ainda não tem resultados satisfatórios. Em um ambiente muito simplificado, com *gateways* e nós instalados em um campo de futebol, Mackey e Spachos (2019) obtiveram erros de posicionamento de até 21,9 m – muito grande considerando que a distância entre os *gateways* era de 100m. Fargas e Petersen

(2017), experimentando em um ambiente urbano, relataram um desempenho significativamente melhor, porém ainda inadequado (entre 60 e 200 metros de erro para nós estáticos distantes cerca de 2km). Podevijn *et al.* (2017), também em ambiente urbano, fizeram uma análise de desempenho na localização de nós móveis por meio de multilateração para diversos fatores de espalhamento da modulação *LoRa*. Os nós foram considerados em velocidades de caminhada, de bicicleta e de carro. Os resultados apontaram erros na ordem de centenas de metros, ou seja, muito insatisfatórios, ainda mais quando considerado o 90º percentil, quando o erro fica na faixa dos 300 a 600m. Ainda, os melhores resultados são encontrados com a utilização da menor taxa de transmissão possível da modulação *LoRa*, o que limita o tamanho dos pacotes de dados. Em suma, como mencionado por Danebjer e Halldorsson (2018), os casos de uso dessa alternativa de geoposicionamento se limitam àqueles em que somente uma estimativa grosseira da localização é requerida ou quando o uso de GPS não é uma opção devido a limitações de custo ou de consumo da bateria.

Cabe destacar aqui que, embora a abordagem escolhida para geoposicionamento no contexto deste trabalho seja por meio de GPS, a implantação de um sistema com ambas as tecnologias permitiria ao Lactec, no futuro, realizar experimentos e pesquisa a respeito da multilateração de sinais *LoRa* para geolocalização, para considerar possíveis implantações em outros projetos ou neste próprio, caso novos trabalhos demonstrem um desempenho mais satisfatório.

2.1.4 Soluções comerciais

Dispositivos *LoRaWAN* para aplicações de rastreamento

Conforme apresentado na Subseção 2.1.1, o protocolo de rede *LoRaWAN*, que tem *LoRa* como camada física, possui longo alcance e baixo consumo como suas principais vantagens. Além disso, a baixa latência e um alto valor de taxa de dados não são críticos para grande parte das aplicações de rastreamento. Esses fatores fizeram com que o *LoRaWAN* se tornasse uma escolha popular nestes casos e atualmente é possível encontrar módulos “*LoRaWAN tracker*” de diversos fabricantes. Em geral, esses dispositivos possuem tamanho reduzido e são constituídos de um rádio *LoRa*, um GPS e um acelerômetro alimentados por bateria.

A aplicação padrão de fábrica destes módulos costuma ser o envio de mensagens *LoRaWAN* contendo informações de localização ao se detectar variações de aceleração, indicando

uma provável movimentação. Alguns dispositivos contam também com botão de pânico que permite envio de mensagens em situação de emergência por meio do acionamento de um *push button*.

Entre os modelos disponíveis no mercado, alguns são vendidos como produtos prontos, como é o caso das soluções oferecidas pela holandesa IoTracker (2021) e pela húngara ZTrack (2021). No caso deste projeto, buscou-se uma solução aberta, com ampla documentação de *hardware* e *software*, que pudesse servir como base para o desenvolvimento de uma nova aplicação. O módulo escolhido foi o LGT-92 da Dragino (2019a). Outras soluções também com essas características são as da chinesa RAK (2021).

Gateways *LoRaWAN*

Na arquitetura *LoRaWAN*, descrita em maiores detalhes na Seção 2.2, a comunicação dos dispositivos com o *Network Server* é feita por intermédio de um *gateway*. Este estabelece uma ponte entre a comunicação *LoRa*, dos nós e a *Internet Protocol* (IP), do servidor.

Em geral, um *gateway* consiste em um sistema *Linux* embarcado com interface de rádio *LoRa*. As soluções existentes, com exceção das iniciativas de desenvolvimento de mini *gateways* de canal único a partir de um hardware semelhante a um *end device*, utilizam os chips SX1301/SX1308 e SX1255/SX1257 da *Semtech*, empresa detentora da propriedade intelectual da modulação *LoRa*, específicos para a implementação de concentradores. Quanto ao *software*, a mesma empresa disponibiliza gratuitamente o código que pode ser utilizado para este fim.

Atualmente, há uma gama de soluções comerciais disponíveis no mercado. Uma primeira classificação entre elas seria entre dispositivos *standalone* e *shields*, como o Sentries RG1xx, da Laird (2018) e o RAK831 da RAK (2021), respectivamente. Os últimos necessitam ser conectados a um sistema *Linux*, como uma *Raspberry PI*, enquanto os primeiros já são a solução completa. Além desta diferenciação, as opções variam em relação à robustez (aplicações *indoor* e *outdoor*) e número de canais de recepção simultâneos.

O número mínimo de canais de recepção, com largura de banda 125kHz, de um *gateway* convencional são oito. Opções com 16 ou 64 canais também são comumente encontradas à venda – modelos Kona Macro e Kona Mobile da Tektelic (2021), por exemplo. O aumento do número de canais proporciona uma melhora na taxa de sucesso da comunicação, especialmente em redes com uma alta densidade de dispositivos. Isto ocorre porque a chance de colisão entre mensagens diminui, dado que existem mais canais disponíveis para o envio de *uplinks*.

Para ser capaz de escutar simultaneamente mais de dez canais – nove *LoRa*, sendo oito com banda de 125kHz e um com banda 500kHz, e um GFSK – é necessário replicar o circuito de recepção. A inclusão destes *chips* de rádio da *Semtech* acaba tendo grande impacto no custo do *gateway*.

A necessidade de gabinete para uso em ambientes externos é outro requisito que causa um aumento significativo no preço, principalmente nas soluções completas. Como exemplo, o *gateway* Sentries RG191 da *Laird*, que custa 250 dólares, triplica de preço na sua versão *outdoor* (450-0191) com encapsulamento IP67, chegando ao valor de 756 dólares por unidade (MOUSER, 2021).

Para a aplicação aqui apresentada, um *gateway* de oito canais é o suficiente, visto que não há muitos nós conectados à rede. Optou-se então por utilizar o dispositivo RD43HATGPS tipo *shield*, desenvolvido pela empresa brasileira Radioenge (2021), por conta do custo e da flexibilidade da utilização de um sistema *Linux* totalmente aberto (*Raspberry Pi*) em um projeto de pesquisa e desenvolvimento. A montagem de um gabinete para uso em ambiente externo ficou por conta dos autores.

Network Server LoRaWAN

Uma parte importante da rede *LoRaWAN*, responsável por grande parte da inteligência da rede, conforme será explicado melhor na Seção 2.2, é o *Network Server*. Ele é incumbido de receber os dados de aplicação vindos dos *gateways* e direcionar para a aplicação correta. Sendo assim, uma instância de rede *LoRaWAN* é definida pelo seu *Network Server* – os *gateways* que façam parte dessa rede devem apontar para esse servidor.

Para promover a internet das coisas de forma global, a comunidade *The Things Network* desenvolveu uma infraestrutura *LoRaWAN* colaborativa de código aberto, em que os usuários podem integrar novos *gateways* à rede. Nós pertencentes a essa rede podem utilizar qualquer um desses *gateways* para se comunicar com os servidores de rede e com seu servidor de aplicação. Atualmente, a rede conta com mais de 15 mil *gateways* espalhados pelo mundo – um pouco mais de 100 deles em território brasileiro. Apesar da vantagem de poder usufruir de uma infraestrutura de rede *LoRaWAN* colaborativa proporcionada pela *The Things Network*, algumas regras de uso devem ser seguidas, como limitar o tempo no ar somado de todos os pacotes *LoRa* enviados por um nó durante um dia em 30 segundos e enviar no máximo 10 mensagens de *downlink* para um nó por dia (NETWORK, 2021). Por isso, em algumas aplicações é necessário recorrer a redes

privadas, seja pelo lançamento da própria estrutura *LoRaWAN* completa ou pela contratação de uma rede *LoRaWAN* comercial.

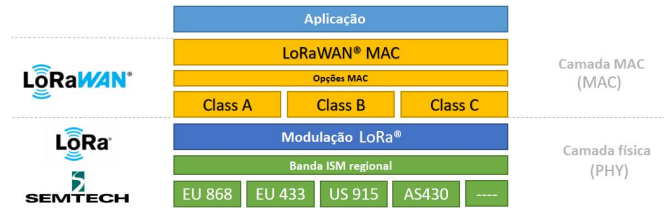
A própria *The Things Industry*, mantenedora do projeto *open-source* da comunidade *The Things Network*, oferece alguns planos para contratação de redes *LoRaWAN* privadas, que vão desde uma simples privatização do sistema em nuvem previamente descrito, passando por versões hospedadas pelos clientes e suportadas pela empresa, até uma versão em nuvem administrada pela *The Things Industry* com suporte prioritário, garantia de disponibilidade do serviço, entre outros benefícios. Existem outras redes *LoRaWAN* comerciais disponíveis no Brasil (INDUSTRIES, 2021). A *American Towers*, por exemplo, instalou *gateways* em suas torres espalhadas pelo país para criação de uma rede *LoRaWAN* neutra em parceria com a *Everynet* (AMERICAN TOWERS, 2021).

A comunidade *Chirpstack* é outra que desenvolve uma infraestrutura *LoRaWAN* de código aberto. Neste caso, toda a pilha pode ser instalada em um ambiente *Linux* privado, incluindo o sistema operacional *Raspbian* utilizado nas placas *Raspberry Pi*, que é uma variante do *Linux Debian*, o que possibilita a instalação do *Network Server* no mesmo computador em que o *Gateway LoRaWAN* é executado (CHIRPSTACK, 2021). Essas características fazem da *Chirpstack* a pilha *LoRaWAN* ideal para projetos que necessitem rede privada, que não se enquadrem nas restrições de uso da *The Things Network* e que não precisem de toda a infraestrutura e benefícios proporcionados pelas soluções comerciais. Como o monitoramento de veículos privados do Lactec apresentou todas essas características, a pilha *Chirpstack* foi escolhida para este projeto, tendo sido implantada em uma máquina virtual com sistema *Ubuntu* hospedada na própria infraestrutura da empresa.

2.2 LORAWAN

Long Range Wide Area Network é um protocolo de rede para dispositivos de baixo consumo concebido pela *Semtech* e mantido pela *LoRa Alliance*. Apesar do protocolo ser aberto, sua camada física exige a utilização da modulação *LoRa*, desenvolvida pela *Cycleo* e, atualmente, propriedade da *Semtech*. Entre as características deste padrão, pode-se destacar o longo alcance e robustez da comunicação, o baixo consumo de energia e a possibilidade de viabilizar, inclusive em uma região espacial limitada, a conexão de um grande número de pontos com comunicação bidirecional. Estes atributos vão ao encontro de requisitos de dispositivos para uso em aplicações de IoT, sendo o *LoRaWAN* cada vez mais popular neste meio.

Figura 5 – Camadas do padrão *LoRaWAN*.



Fonte: Adaptado de Semtech (2019a).

2.2.1 Arquitetura de rede

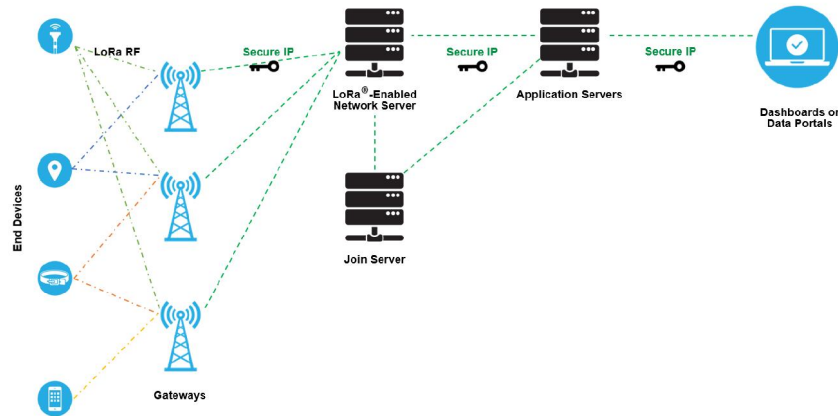
Em uma rede *LoRaWAN*, os nós se comunicam com um ou mais *gateways* e cada um destes se comunica com o *Network Server* (Figura 6). Nesta topologia, conhecida como “estrela de estrelas”, não é possível haver comunicação direta entre os nós, diferentemente do que ocorre em uma rede ponto-a-ponto ou em uma rede *mesh*. Nesta, um nó pode servir como uma ponte, repassando mensagens de dispositivos mais afastados para o *gateway*. Esta propriedade acaba aumentando a cobertura da rede, mas também sua complexidade. Dado que a modulação *LoRa* é capaz de operar a longas distâncias por si só, a escolha da topologia estrela de estrelas permite que os componentes da rede, especialmente os *end devices*, possam ser mais simples, necessitando menos poder de processamento e menos tempo em estado ativo. Assim, sem comprometer a cobertura de rede, o *LoRaWAN* consegue ter uma diminuição significativa no consumo de energia e redução dos custos de infraestrutura de rede (SEMTECH, 2019a).

A topologia da rede *LoRaWAN* se encaixa perfeitamente no cenário deste trabalho, com os *end devices* acoplados aos veículos monitorados e os *gateways* instalados cada um em uma sede do Lactec onde haja uma estação de carregamento dos veículos. Assim, bastaria que em cada ponto geográfico, próximo às vias destinadas a estes veículos, onde se deseja monitorá-los, o enlace entre o nó e algum dos *gateways* fosse suficiente para comunicação, o que é esperado dado o longo alcance e a robustez da modulação *LoRa*.

2.2.2 LoRa

A modulação proprietária *LoRa* foi desenvolvida para trabalhar com níveis baixos de potência e ainda assim apresentar uma elevada imunidade ao ruído, sendo capaz de atingir um longo alcance entre transmissor e receptor. Distâncias de 5km em áreas urbanas e 15km em áreas com visada (rurais) podem ser obtidas com o uso desta tecnologia (SEMTECH, 2019a). O

Figura 6 – Arquitetura da rede LoRaWAN.



Fonte: Semtech (2019a)

aumento da robustez do sinal, por sua vez, tem como revés a diminuição da taxa de dados. Esses aspectos são característicos da técnica conhecida como *Spread Spectrum* (SS) (em português, espalhamento espectral). Mais especificamente, o *LoRa* é uma variação da modulação *Chirp Spread Spectrum* (CSS).

A camada física *LoRa* é completamente independente das camadas superiores do protocolo *LoRaWAN*, podendo ser utilizada em aplicações que não sejam *LoRaWAN* compatíveis, com outras tecnologias de rede, transporte e aplicação.

Spread Spectrum

A base do *LoRa* é a técnica de modulação *spread spectrum*. Esta consiste no espalhamento na frequência do sinal a ser transmitido de modo que este ocupe uma banda de transmissão muito maior do que sua taxa de dados.

Este processo é responsável por conferir ao *LoRa*, bem como às outras modulações SS, uma robustez de sinal superior em comparação com outras técnicas. Os sinais *spread spectrum* apresentam também uma alta tolerância em relação a ruído e a interferência, tanto de sinais provenientes de outros dispositivos utilizando o mesmo canal, quanto de interferências intencionais – *jamming*. Além disto, o espalhamento da energia na frequência faz com que o sinal transmitido esteja muito próximo do nível de ruído, quando não abaixo dele, dificultando a interceptação da mensagem por outros sistemas. Estas características fizeram com que esta tecnologia fosse amplamente empregada na indústria militar (PROAKIS, 2008).

A robustez de um sinal propagado junto ao nível de ruído, embora contra intuitiva à

primeira vista, se torna evidente por meio do teorema de Shannon-Hartley (Equação 1) (PRO-AKIS, 2008), bem como a consequente redução da taxa de dados. O teorema define o valor quantitativo do limite da taxa de transmissão de dados, ou capacidade do canal, com base na relação sinal-ruído e largura de banda, para se alcançar uma transmissão livre de erros em um canal com ruído aditivo branco e gaussiano (AWGN).

$$C = BW * \log_2 \left(1 + \frac{S}{N} \right) \quad (1)$$

Onde:

C = Capacidade [bits/s]

BW = Largura de banda [Hz]

S = Potência média do sinal [W]

N = Potência média do ruído [W]

Rearranjando a Equação 1 em termos do logaritmo natural e aproximando por série de Taylor, dado que o sinal *spread spectrum* se encontra normalmente abaixo do nível de ruído, chega-se na Equação 2 (SEMTECH CORPORATION, 2015).

$$\frac{N}{S} \approx \frac{BW}{C} \quad (2)$$

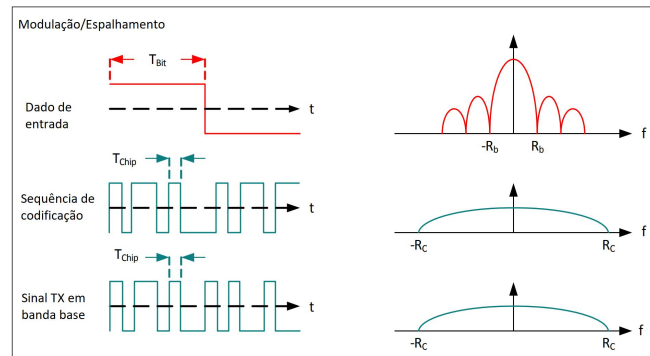
Assim, dado que a relação sinal-ruído é baixa e a banda disponível para transmissão é limitada, pode-se concluir que os ganhos de robustez vêm em desfavor da taxa de transmissão de informação, fazendo com que este tipo de modulação seja mais adequada para aplicações que não requerem um elevado tráfego de dados.

A modulação *spread spectrum* mais simples é a *Direct Sequence Spread Spectrum* (DSSS). Esta consiste na modulação da fase de uma sequência pseudorrandômica pelo sinal que contém a informação de interesse. Esta sequência, também conhecida como código de espalhamento ou sequência de *chips*, apresenta um *baud rate* maior do que o sinal de dados e por meio da multiplicação desses dois sinais obtém-se o espalhamento na frequência.

Chirp Spread Spectrum

Embora a modulação DSSS seja conceitualmente simples, um dos desafios de sua implementação na prática é a necessidade de um elevado grau de sincronismo entre o transmissor e o receptor. Este requisito acaba fazendo com que dispositivos DSSS exijam um *clock* de referência extremamente preciso, o que costuma acarretar um alto custo. Além disso, o sincronismo se

Figura 7 – Modulação *Direct Sequence Spread Spectrum* (DSSS).



Fonte: Adaptado de Semtech Corporation (2015)

torna especialmente complexo em dispositivos com restrição de energia por não ser possível que permaneçam constantemente ligados. Isto posto, a modulação DSSS pode ser inviável para algumas aplicações, em especial na área de IoT.

Já a técnica adotada no *LoRa*, o CSS, causa o espalhamento do sinal por meio da variação da frequência do sinal ao longo do tempo, como apresentado na Figura 8.

A vantagem da CSS em relação ao DSSS é a redução da complexidade do sincronismo entre o transmissor e o receptor, uma vez que os *offsets* de frequência e *timing* são equivalentes em ambos os lados. Além disso, esta modulação é bastante resistente às interferências de *multipath*, *fading* e efeito *Doppler*, o que aumenta também sua robustez e consequentemente a distância em que se é possível comunicar (SEMTECH CORPORATION, 2015).

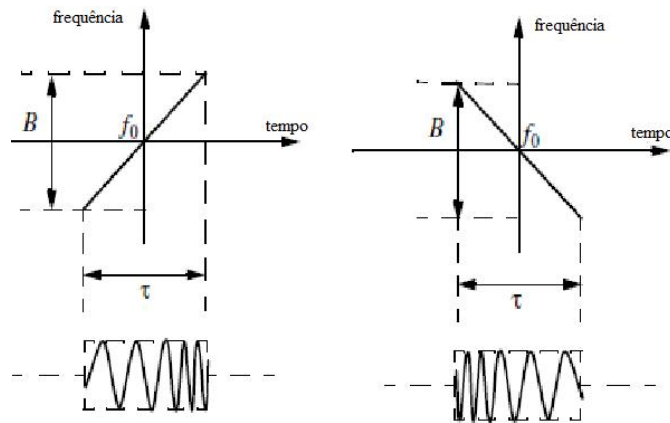
Por meio da utilização do CSS, a modulação *LoRa* apresenta um baixo consumo de energia e a alta robustez do sinal, características da técnica *spread spectrum*, bem como um custo reduzido em comparação com o DSSS. Estes fatores favoreceram seu uso em larga escala.

Sinal e parâmetros

O sinal *LoRa* é uma onda senoidal modulada em frequência. Esta apresenta uma variação ao longo do tempo que pode ser crescente ou decrescente, dando origem a um *upchirp* ou *downchirp*, respectivamente.

Cada símbolo do sinal *LoRa* varia sua frequência percorrendo em totalidade a largura de banda BW em torno da frequência da portadora. O passo no qual a frequência varia é definido pelo *Spreading Factor* (SF). Quanto maior é o fator de espalhamento, menor é esta variação. Um símbolo é descrito por 2^{SF} passos, ou *chips*, com duração de $1/BW$ segundos (SEMTECH

Figura 8 – Sinais *upchirp* (esquerda) e *downchirp* (direita).



Fonte: Adaptado de Amin *et al.* (2014).

CORPORATION, 2015).

O valor de SF define quantos *bits* são representados em um símbolo e consequentemente o número de símbolos (2^{SF}) existentes na modulação. Já o valor efetivo do símbolo, $k \in \{0,1,2,\dots,2^{SF} - 1\}$, caracteriza o *offset* de início da varredura em frequência (ELSHABRAWY; ROBERT, 2018).

Cada símbolo *LoRa* pode ser descrito pela Equação 3, na qual cada *chip* é representado por um índice $n \in \{0,1,2,\dots,2^{SF} - 1\}$ (ELSHABRAWY; ROBERT, 2018).

$$s_k = \sqrt{\frac{E_s}{2^{SF}}} \exp \left[j2\pi * (k + n) \bmod 2^{SF} * \frac{1}{2^{SF}} * n \right] \quad (3)$$

Onde:

s_k = Símbolo

E_s = Energia de símbolo

SF = Fator de espalhamento

k = Valor do símbolo

n = Índice do *chip*

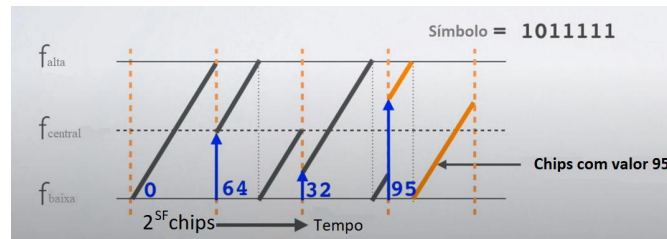
Na Figura 9, pode-se observar, como exemplo, a variação de frequência e *offsets* para diferentes símbolos com fator de espalhamento $SF7$.

Além do CSS, o padrão *LoRa* utiliza as técnicas de *whitening*, *interleaving*, mapeamento *Gray* e codificação *Hamming* para aumentar a robustez da modulação (Figura 10). Mais detalhes a respeito destas técnicas podem ser encontrados no artigo escrito por Robyns *et al.* (2018).

Com os parâmetros apresentados, é possível equacionar a taxa de dados para o *LoRa*. Pela Equação 4 (SEMTECH CORPORATION, 2015), verifica-se que o aumento do fator de

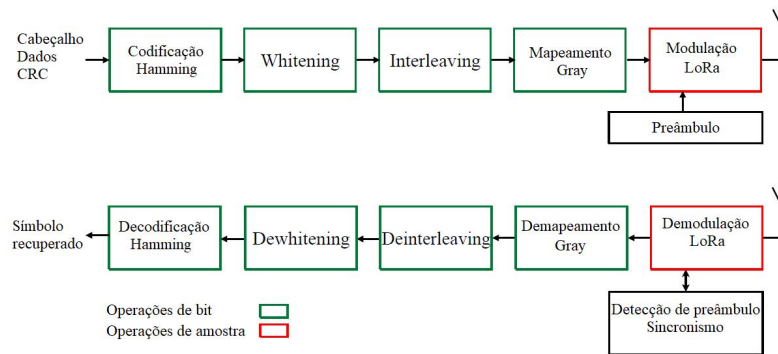
¹ Disponível em https://www.mobilefish.com/download/lora/lora_part13.pdf. Acesso em 3 de junho de 2020.

Figura 9 – Exemplo de símbolos *LoRa* com fator de espalhamento SF7.



Fonte: Adaptado do *Powerpoint LoRa/LoRaWAN: Symbol, spreading factor and chip*¹.

Figura 10 – Diagrama de blocos da modulação e demodulação *LoRa*.



Fonte: Adaptado de Afsiadis *et al.* (2019).

espalhamento e da taxa de codificação, bem como a diminuição da largura de banda, acarretam diminuição da taxa de dados. Uma descrição mais detalhada dos valores possíveis de BW, SF e CR pode ser encontrada nos *datasheets* dos rádios da linha SX12xx da Semtech (2019b).

$$R_B = \frac{SF}{2^{SF}} * BW * \frac{4}{(4 + CR)} \quad (4)$$

Onde:

R_b = Taxa de *bits* [bits/s]

SF = Fator de espalhamento

BW = Largura de banda [Hz]

CR = Taxa de codificação

O padrão *LoRaWAN* determina alguns valores específicos para esses parâmetros, sendo: 125kHz, 250kHz e 500kHz para BW; 7, 8, 9, 10, 11 e 12 para SF; 1 para CR. As combinações possíveis entre estes parâmetros estão relacionadas com a região geográfica em que o rádio opera, visto que a modulação deve respeitar a legislação vigente para a faixa livre ISM (LORA ALLIANCE, 2020). Esta questão será abordada em mais detalhes na Subseção 2.2.3. Por meio da Equação 4, pode-se calcular o melhor e o pior caso para a taxa de *bits*, 21875bps e 293bps,

respectivamente. Como os SF são ortogonais entre si, mensagens com diferentes fatores de espalhamento podem ser transmitidas, sem interferência, ao mesmo tempo. Desta forma, a capacidade total do canal é a soma das taxas de *bit* para os SF empregados (Equação 5). Por exemplo, a capacidade de um canal de 125kHz com taxa de codificação 4/5 e utilizando os SF de 7 até 12 é de 12158bps.

$$C = R_{bSF7} + R_{bSF8} + R_{bSF9} + R_{bSF10} + R_{bSF11} + R_{bSF12} \quad (5)$$

Onde:

C = Capacidade [bps]

R_{bSFx} = Taxa de *bits* para SFx [bps]

Além do alto impacto na taxa de transmissão, o aumento do parâmetro SF proporciona um ganho de sensibilidade no sinal, que passa de -124dBm (125kHz/SF7) para -137dBm (125kHz/SF12). O oposto é verificado na variação da largura de banda: uma redução de -137dBm (125kHz/SF12) para -129dBm (500kHz/SF12) ocorre com aumento de BW de 125kHz para 500kHz (SEMTECH, 2019b). Estes resultados podem ser verificados por meio da Equação 6.

$$S = NoiseFloor + NF + SNR_{req}$$

$$NoiseFloor = 10 * \log_{10}(k * T * BW * 1000)[dBm]$$

$$S = -174 + 10 * \log_{10}BW + NF + SNR_{req} \quad (6)$$

Onde:

S = Sensibilidade [dBm]

NF = Figura de ruído do rádio [dB]

SNR_{req} = SNR mínimo necessário para a demodulação do sinal [dB]

k = Constante de *Boltzmann* ($1.380649 \times 10^{-23}[JK^{-1}]$)

T = Temperatura [K]

O cálculo do valor de chão de ruído (*NoiseFloor*) se deu por meio da Equação 1 considerando SNR igual a 0dB e temperatura de 293K (SEMTECH CORPORATION, 2015). Os valores para os SNR necessários para a demodulação do sinal recebido (Quadro 1) e o valor para a figura de ruído (6dB) do rádio foram obtidos no *datasheet* do rádio SX1272 (SEMTECH, 2019b).

Quadro 1 – SNR necessário para a demodulação em função do SF escolhido.

Spreading factor SF	LoRa Demodulator SNR
7	-7.5dB
8	-10dB
9	-12.5dB
10	-15dB
11	-17.5dB
12	-20dB

Fonte: Semtech (2019b).

Outra consequência do aumento do fator de espalhamento e da diminuição da largura de banda é a extensão considerável do tempo no ar da mensagem. Isto ocorre porque a quantidade de *chips* por símbolo cresce e cada um deles apresenta duração de $1/BW$ segundos.

Apesar das modulações *spread spectrum* apresentarem a característica de operar com sinais bem próximos do nível do ruído e inclusive abaixo dele, como pode ser verificado nos valores de SNR negativos para o *LoRa* (Quadro 1), é evidente que a configuração correta dos parâmetros de modulação é imprescindível para possibilitar a comunicação entre o transmissor e o receptor, principalmente quando separados por uma longa distância. No *LoRaWAN*, a otimização destes parâmetros fica por conta do algoritmo *Adaptive Data Rate* (ADR) que manipula os valores de SF e BW dos dispositivos de modo a manter um RSSI suficiente para demodular um sinal transmitido com o menor tempo no ar possível. Este procedimento visa diminuir o gasto energético e cumprir com as limitações de uso de banda definidas pela legislação.

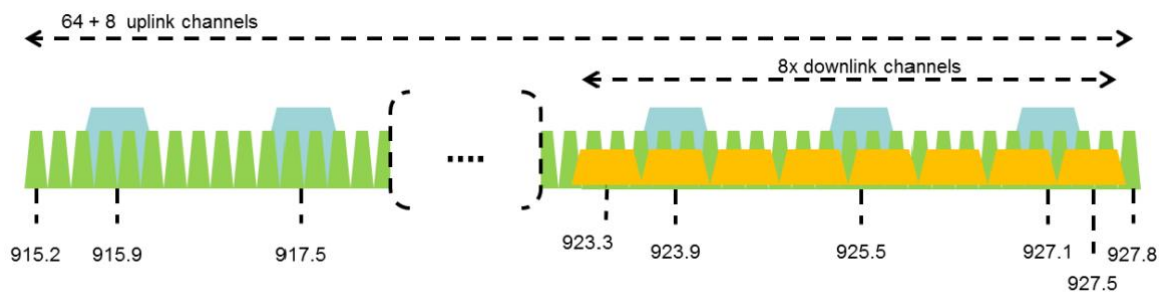
2.2.3 Parâmetros regionais

Como pode ser observado na Figura 5, na camada física da rede *LoRaWAN*, a modulação *LoRa* é aplicada sobre a banda ISM regional, que pode ter diferentes padrões. Isso ocorre porque a faixa de frequência utilizada para a banda ISM muda conforme a região. Além disso, outras regras de uso do canal podem variar. Normalmente, os países que chegam mais tarde na implantação de redes *LoRaWAN* enquadram na legislação alguma das normas de parâmetros regionais mais difundidas. Os principais padrões são o americano US902-928MHz, o australiano AU915-928MHz e os europeus EU863-870MHz e EU433MHz. Uma lista completa das normas adotadas em vários países pode ser consultada no documento “LoRaWAN Regional Parameters” (ALLIANCE, 2018). No Brasil, podem ser utilizados o padrão AU915-928MHz, na banda não licenciada, ou o padrão EU433, porém, neste caso, exige-se licenciamento. Neste trabalho, foi utilizada a norma AU915-928MHz.

AU915-928MHz

No padrão AU915-928MHz, utilizado no Brasil, a banda é dividida em três planos de frequências: dois para *upstream*, ou seja, enviar mensagens do nó para o *gateway*, e um para *downstream* - do *gateway* para o nó. Esses planos são representados na Figura 11. O Quadro 2 sumariza suas características, apresentando a banda (BW) utilizada na modulação *LoRa* e as taxas de transmissão (DR) permitidas para cada caso. Em todos os casos, é utilizada a taxa de codificação 4/5. Em verde na figura, está representado o plano que foi chamado de “Uplink 1” na tabela, em azul, “Uplink 2” e, em amarelo, “Downlink”.

Figura 11 – Frequências de canais do padrão AU915-928MHz



Fonte: Alliance (2018)

Quadro 2 – Planos de frequência - AU915-928MHz

Conjunto	Número de canais	Primeiro canal	Salto entre canais	BW - LoRa	DR
Uplink 1	64	915.2MHz	200kHz	125kHz	DR0-DR5
Uplink 2	8	915.9MHz	1.6MHz	500kHz	DR6
Downlink	8	923.3MHz	600kHz	500kHz	DR8-DR13

Fonte: Alliance (2018)

A configuração da taxa de transmissão (DR) altera o fator de espalhamento (SF) que, por consequência, juntamente à banda (BW) configurada para a modulação, define a taxa de dados conforme a Equação 4, resultando nos valores apresentados na Quadro 3. Como cada configuração de DR é destinada a somente um dos três conjuntos de canais apresentados na Quadro 2 e, portanto, tem a largura de banda da modulação (BW) já definida, a taxa de dados é alterada pela configuração do fator de espalhamento (SF). Os valores de DR 7, 14 e 15 são reservados para uso futuro. Pode ser notado que, quando SF é 11 ou 12, o valor da taxa de transmissão fica abaixo da esperada pela Equação 4. Isso ocorre porque, para estes fatores de espalhamento, cujo tempo de símbolo é muito longo, é necessário ativar um otimizador, chamado *Low Data Rate Optimizer*, que diminui o número de *bits* por símbolo (SEMTECH, 2019b).

No Quadro 3, também é informada a limitação do tamanho do *payload*, isto é, a quantidade de *bytes* da camada de aplicação, para cada um dos possíveis valores de DR. Esta definição do protocolo *LoRaWAN* existe para limitar o tempo de ocupação do canal pelos dispositivos, embora, no Brasil, não exista uma limitação legal de tempo de permanência (*dwell time*) nem de ciclo de trabalho (*duty cycle*) para dispositivos com modulação *LoRa*, que se enquadram no item 10.3 do ato 14448 da Anatel. Mesmo assim, pode ser escolhido limitar o tempo de permanência a 400 milissegundos, reduzindo o *payload* máximo aos valores da última coluna da tabela.

Quadro 3 – Configurações de data rate- AU915-928MHz

DR	BW	SF configurado	Bit rate	Payload máx.	Payload máx. (<i>dwell time</i> = 400ms)
0	125kHz	12	250 bps	51 bytes	Indisponível
1	125kHz	11	440 bps	51 bytes	Indisponível
2	125kHz	10	980 bps	51 bytes	11 bytes
3	125kHz	9	1760 bps	115 bytes	53 bytes
4	125kHz	8	3125 bps	242 bytes	125 bytes
5	125kHz	7	5470 bps	242 bytes	242 bytes
6	500kHz	8	12500 bps	242 bytes	242 bytes
8	500kHz	12	980 bps	53 bytes	53 bytes
9	500kHz	11	1760 bps	129 bytes	129 bytes
10	500kHz	10	3900 bps	242 bytes	242 bytes
11	500kHz	9	7000 bps	242 bytes	242 bytes
12	500kHz	8	12500 bps	242 bytes	242 bytes
13	500kHz	7	21900 bps	242 bytes	242 bytes

Fonte: Alliance (2018)

A potência isotrópica radiada equivalente (EIRP) dos transmissores também é limitada pela norma regional e, no caso da AU915-928, o valor máximo permitido é de 30dBm ou 1W.

O Quadro 4 apresenta valores padrão para os intervalos de tempo das janelas de recepção do nó. Como citado na Subseção 2.2.11, essas temporizações são usadas após cada *uplink* por um dispositivo classe A ou classe B para receber uma mensagem de *downlink*.

Quadro 4 – Planos de frequência - AU915-928MHz

Janela de recepção 1 para dados de aplicação	1s
Janela de recepção 2 para dados de aplicação	2s
Janela de recepção 1 para processo de <i>join</i>	5s
Janela de recepção 2 para processo de <i>join</i>	6s

Fonte: Alliance (2018)

2.2.4 End devices

Os *end devices*, por meio de mensagens utilizando modulação *LoRa*, se comunicam com um ou mais *gateways* diretamente, ou seja, apresentam uma comunicação *single-hop*. Na rede,

cada dispositivo é identificado por uma sequência numérica de 64 *bits* chamada *device Extended Unique Identifier (EUI)*, gravada pelo fabricante. Para criar um EUI, o fabricante deve ter um OUI (*Organizationally Unique Identifier*) cedido pelo Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE).

Em relação ao *hardware*, um *end device* é composto por chip capaz de realizar a modulação *LoRa* e, na maioria das aplicações, sensores e/ou atuadores controlados por um microcontrolador que apresenta baixo consumo, com *firmware* otimizado para a economia de energia, pois geralmente estes dispositivos são alimentados via bateria. Atualmente, a maioria dos dispositivos utiliza os *chips* da linha SX12xx da *Semtech* para realizar a modulação *LoRa*.

2.2.5 Gateway

Os *gateways* são responsáveis por encaminhar mensagens dos *end devices* para o *Network Server* e vice-versa. Sendo assim, são capazes de se comunicar por modulação *LoRa* e por IP via *wi-fi*, *Ethernet* ou conexão celular. As mensagens que trafegam por estes *gateways* têm sua integridade verificada por meio da verificação cíclica de redundância CRC, impedindo que pacotes inválidos sejam repassados. Às mensagens de *uplink*, são anexados metadados, como estampa de tempo e RSSI. Ademais, os pacotes, tanto de *uplink* quanto de *downlink*, não são interpretados pelo *gateway*.

Uma mensagem transmitida por um nó pode ser ouvida por mais de um *gateway*, uma vez que não há associação fixa entre estes dispositivos. Se isso acontece, cada um dos *gateways* encaminha o pacote ao *Network Server* e o controle de informações duplicadas fica por conta deste, que deve encaminhar somente um dos pacotes ao *Application Server* - processo chamado de deduplicação. O envio das mensagens por múltiplos *gateways*, além de reduzir a probabilidade de perda de pacotes, pode ser explorado para disponibilizar dados de localização de *end devices* por meio de triangulação, caso a localização dos *gateways* seja conhecida. A escolha de qual *gateway* é encarregado de enviar mensagens de *downlink* para um determinado *end device* também se dá no *Network Server* e é feita com base nos valores de RSSI das mensagens. Desta forma, o *gateway* com melhor enlace em relação ao *end device* realiza a transmissão do *downlink*.

Os *gateways* comerciais possuem algumas variações em relação ao número de canais que ouvem simultaneamente. O mínimo são oito, podendo, em passos de oito canais, chegar até a sessenta e quatro. Podem também ser voltados ao ambiente interno ou externo, apresentando diferenças na robustez física e antena escolhida.

Um *gateway* simples pode ser obtido combinando-se uma placa *Raspberry Pi* com um *shield LoRa*. Um *software* básico conhecido como *LoRaWAN packet forwarder* é disponibilizado gratuitamente pela própria *Semtech* para este fim. Neste trabalho, foi utilizada esta abordagem, com um *shield LoRa* Radioenge RD43HATGPS, produzido no Brasil, acoplado a uma placa *Raspberry Pi 3B +*. O *gateway* utilizado é apresentado em mais detalhes no Capítulo 3.

2.2.6 Network Server

O *Network Server* é o principal responsável pelo gerenciamento da rede e faz isso por meio de mensagens da camada MAC (Controle de Acesso ao Meio) trocadas com os *end devices*. Ele deve controlar dinamicamente os parâmetros da modulação *LoRa* – SF e taxa de transmissão de dados – utilizados pelos nós a fim de otimizar a comunicação, especialmente em relação à economia de energia, frente a mudanças no ambiente como deslocamento de dispositivos ou aumento de ruído. Outra função do *Network Server* é encaminhar as mensagens de aplicação dos *end devices* para o *Application Server* ao qual eles pertencem. Ele não tem, efetivamente, acesso aos dados das aplicações, uma vez que estes são criptografados por meio de chaves padrão de criptografia avançada (AES), permitindo a concepção de redes *LoRaWAN* públicas seguras. Mais especificamente, o *Network Server* deve ser capaz de executar as seguintes ações:

- Checar o endereço dos dispositivos.
- Autenticar e contabilizar pacotes.
- Gerar confirmação de mensagens recebidas.
- Adaptar a taxa de transmissão de dados dinamicamente por meio do protocolo ADR.
- Atender requisições da camada MAC provenientes do nó.
- Repassar pacotes de *uplink* para o servidor de aplicação correspondente.
- Agendar e encaminhar pacotes de *downlink* para o *end device* por meio do *gateway* selecionado a partir das medidas de RSSI.
- Trafegar mensagens originadas no procedimento de *join* entre o *end device* e o *Join Server*.

2.2.7 Join Server

O *Join Server* viabiliza o método de integração de um nó à rede chamado de *Over-The-Air Activation* (OTAA). A partir de um *join-request* enviado por um nó e repassado pelo *Network Server*, o *Join Server* valida o dispositivo e deriva as chaves de criptografia de sessão para ambos *Network* e *Application Server*. Se o processo de *join* obtiver sucesso, uma resposta (*joinAns*) é enviada pelo *Join Server* ao *Network Server*, o qual encaminha um *join-accept* ao *end device*, atribuindo-lhe um endereço na rede — chamado no protocolo *LoRaWAN* de *devAddr*. Nessa resposta, o *Join Server* também indica ao *Network Server* qual o *Application Server* ao qual as mensagens desse *end device* devem ser encaminhadas.

2.2.8 Application Server

O servidor de aplicação recebe do *Network Server* somente os dados que lhe são pertinentes, ou seja, aqueles originados pelos *end devices* que nele estão cadastrados. No servidor de aplicação, os dados podem ser descriptografados e interpretados. Este elemento também é responsável por gerar e encriptar as mensagens de *downlink*.

2.2.9 Device Commissioning

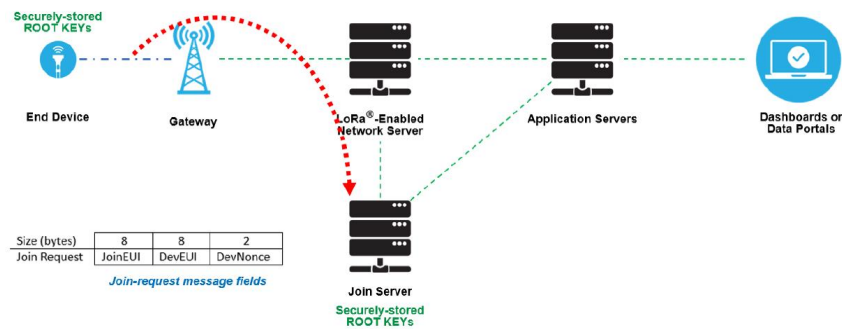
Ao iniciar a operação, um dispositivo deve vincular-se à rede para que seja autenticado e possa operar seguramente. Para isto, ele deve escolher uma das duas possibilidades de ativação: *Over-The-Air Activation* (OTAA) ou *Activation by Personalization* (ABP). A primeira executa o processo de *join*, descrito na sequência (Subseção 2.2.10), no qual as chaves de sessão e *devAddr* são gerados dinamicamente. Quando selecionada a segunda opção, os dispositivos não passam por esse processo e, normalmente, as informações necessárias para comunicação devem ser gravadas durante a fabricação ou por uma configuração. O fato de ter o ID e as chaves estáticos e predefinidos faz com que o dispositivo esteja vinculado a uma única rede/serviço e que suas chaves de sessão não possam ser renovadas periodicamente como no OTAA, o que o torna menos seguro e flexível. Porém, como vantagem, a ativação por ABP permite que o dispositivo esteja funcional a partir do instante em que é ligado. Neste trabalho, foi escolhida a ativação OTAA para os *end devices*, por facilitar o processo de produção e configuração inicial dos dispositivos - todos podem ter o mesmo *firmware* gravado e não é necessária uma configuração inicial especial

para cada *dispositivo*.

2.2.10 Join Procedure

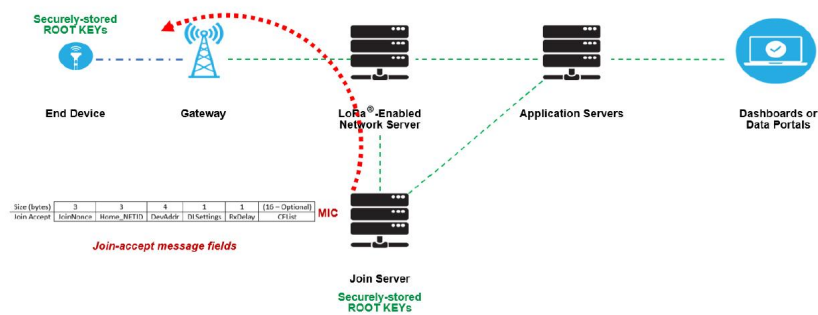
O processo de *join* permite que um dispositivo seja autenticado ao se conectar à rede *LoRaWAN* e que as chaves de criptografia sejam geradas dinamicamente e periodicamente. Estas medidas são tomadas para reforçar a segurança da rede, uma vez que estes dispositivos precisam estar previamente cadastrados e atrelados a um *Application Server* para que ocorra a autorização. O processo, apresentado nas Figuras 12 e 13, tem início quando o *end device* envia ao *Network Server* uma mensagem MAC conhecida como *join-request* e esta é repassada ao *Join Server*. Nela se encontram três dados, o *JoinEUI* e o *DevEUI*, que identificam o *Join Server* e o dispositivo, respectivamente, e o *DevNonce*, um contador incrementado a cada requisição e utilizado para evitar requisições de *join* repetidas, que poderiam ser empregadas em um ataque na tentativa de explorar vulnerabilidades da rede.

Figura 12 – Procedimento de requisição de *join*.



Fonte: Semtech (2019a)

Figura 13 – Procedimento de aceitação de *join*.



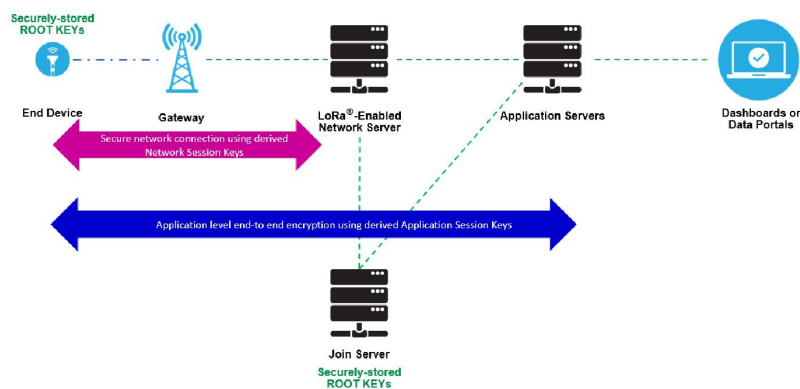
Fonte: Semtech (2019a)

Na mensagem *join-request* se encontram a identificação da rede (*Home_NETID*) e o endereço que o dispositivo deve adotar nela (*DevAddr*). Assim como a mensagem anterior,

a *join-accept* conta com o campo de controle “*JoinNonce*”, cujo papel se assemelha ao do *DevNonce*. Além destes, são enviados os campos *DLSettings*, *RxDelay* e opcionalmente o *CFList* para alterar certas configurações do dispositivo para que ele se integre à rede. Após a aceitação do dispositivo na rede, são geradas as chaves de sessão utilizadas para criptografar os dados da aplicação (*AppSKey*) e da rede (*NwkSKey*). A *AppKey* é uma chave comum que deve ser conhecida tanto pelo dispositivo quanto pelo *Join Server*, previamente ao processo de *join*. Desta forma, as chaves podem ser geradas internamente no dispositivo e no *Join Server*. Este último irá então encaminhar a *NwkSKey* e a *AppSKey* para o *Network* e *Application Servers*, respectivamente.

A Figura 14 ilustra a abrangência de cada uma das criptografias. A *NwkSKey* codifica os dados ainda no dispositivo para serem decodificados pelo *Network Server*. A *AppSKey* também codifica os dados gerados pelo dispositivo, mas para serem decodificados ao chegar ao *Application Server* correspondente. Desta forma, nem os *gateways*, nem o servidor de rede tem acesso aos dados das aplicações.

Figura 14 – Chaves de criptografia na rede *LoRaWAN*.



Fonte: Semtech (2019a)

2.2.11 Classes de dispositivos

O protocolo *LoRaWAN* define três classes de operação (A, B e C) possíveis para um dispositivo. Estas representam um compromisso entre gasto energético e latência do *downlink*. Na classe A, o dispositivo pode receber mensagem do servidor apenas em um curto período de tempo, dividido em duas janelas de recepção, que precede seu próprio *uplink*, permanecendo em modo de baixo consumo o restante do tempo. Na classe B, além de apresentar o mesmo comportamento da classe A, o dispositivo abre janelas de recepção periódicas, sincronizados via

beacon, para a recepção de informações vindas do servidor sem a necessidade de *uplink*. Já um dispositivo classe C apresenta-se sempre em modo de recepção, exceto quando está transmitindo uma mensagem. Todos os dispositivos devem ser capazes de operar em classe A e aqueles que oferecem suporte à classe C devem também suportar a operação em classe B. Na aplicação de monitoramento desenvolvida, a maior parte das mensagens de *downlink* ocorrem com o veículo em percurso e neste estado as mensagens de *uplink* são também mais frequentes. As demais mensagens do servidor para o nó têm função de configuração e não requerem uma latência menor do que o período da mensagem de *keep-alive* (Subseção 3.4.2). Desta forma, visto também que o dispositivo é alimentado via bateria, a classe A foi escolhida como modo de operação.

2.3 CAYENNE LPP

O padrão *Cayenne Low Power Payload (LPP)*² foi empregado na formatação dos dados enviados dos dispositivos para a aplicação. Este protocolo foi desenvolvido especialmente para uso em redes LPWAN, visando generalizar a identificação dos sensores e respectivos tipos de dado, ao mesmo tempo em que atende a limitação de tamanho de *payload* geralmente presente em redes que operam com baixo consumo energético.

Cada dado de sensor enviado contém dois *bytes* de identificação, um para distinção do sensor na aplicação – canal de dado e o outro para definir o tipo de informação – tipo de dado. Na sequência, encontram-se os dados propriamente ditos.

Quadro 5 – Formatação de *payload* Cayenne LPP.

1Byte	1Byte	N bytes	1 Byte	1 Byte	M bytes	...
Canal Dado1	Tipo Dado1	Dado1	Canal Dado2	Tipo Dado2	Dado2	...

Fonte: Adaptado de myDevices (2020).

Enquanto o valor colocado no campo canal de dado é específico da aplicação para possibilitar a identificação da medição, sobretudo quando há mais de um sensor do mesmo tipo, o tipo de dado é baseado na norma *IPSO Alliance Smart Objects Guidelines* para identificação de objetos, com uma adaptação na faixa de ID para que este valor ocupe apenas um *byte*.

Os tipos de dados efetivamente utilizados na aplicação deste trabalho encontram-se descritos no Quadro 6.

O protocolo permite também a criação de novos tipos de dados. No Quadro 6, os tipos de dados criados pelos autores podem ser identificados pelo valor de tipo de dado iniciando em

² Disponível em: <https://developers.mydevices.com/cayenne/docs/lora/#lora-cayenne-low-power-payload>. Acesso em: 29 de maio de 2020.

Quadro 6 – Tipo de dado Cayenne LPP.

Tipo	Tipo de dado	Tamanho [byte]	Resolução
Tensão	116	2	0.01V
Unix timestamp	133	4	1 s
Byte genérico	201	1	1
Velocidade	202	2	0.01 km/h
GPS error	203	2	0.01
GPS (Latitude e longitude)	204	8	0.0000001°

Fonte: Cayenne(2020).

2. Apesar de tornar o protocolo menos genérico, optou-se pela customização de alguns campos de modo a facilitar a decodificação das informações e, no caso do GPS, aumentar a precisão e reduzir a quantidade de *bytes* enviados. O *Cayenne LPP* originalmente possui um tipo de dado para geoposicionamento, mas sua resolução para latitude e longitude é de apenas 0.0001°, menor do que o dispositivo GPS é capaz de fornecer e, além disso, a informação de altitude, irrelevante para a aplicação desenvolvida, também é adicionada ao pacote, totalizando 9 *bytes*. O dado personalizado criado utiliza apenas 8 *bytes* para atingir uma precisão de 0.0000001°.

2.4 GPS

Os GNSS (Sistema Global de Navegação por Satélite) são sistemas de rádio navegação, baseados em ondas de rádio emitidas por uma constelação de satélites, capazes de prover informações de posicionamento sobre qualquer ponto da superfície terrestre. O Sistema de Posicionamento Global (GPS) é um dos mais populares entre estes sistemas e foi desenvolvido, inicialmente para uso militar, nos EUA em 1960 pelo DoD (*Department of Defense*). Este sistema faz uso de uma constelação de vinte e quatro satélites conhecida como NAVSTAR. Pode-se citar também o GALILEO, europeu, o GLONASS, russo e o BeiDou, chinês, como outros exemplos de GNSS (KAPLAN; HEGARTY, 2006).

Estes sistemas funcionam com base nos conceitos de triangulação e tempo de chegada (TOA) para determinação das coordenadas de geolocalização. Por meio de uma estampa de tempo na mensagem e sabendo-se a velocidade de propagação da onda de rádio, é possível calcular a distância entre o satélite e o dispositivo. Com a distância em relação a três satélites, já seria possível realizar a triangulação, mas, para compensar desvios no sincronismo de tempo, mais satélites são considerados no cálculo (KAPLAN; HEGARTY, 2006).

A comunicação entre o satélite e o dispositivo é unidirecional, ou seja, o dispositivo apenas recebe mensagens do satélite, o que permite a coexistência de um número ilimitado

deles. Deste modo, todo o cálculo de posicionamento fica por conta do receptor. Embora o processamento tenha evoluído desde a criação do sistema fazendo com que a aquisição das coordenadas se tornasse mais rápida, o consumo destes dispositivos ainda é relativamente alto, podendo ser crítico para sistemas de baixo consumo e alimentados por bateria, cenário bastante comum em IoT.

2.4.1 Protocolo NMEA

O protocolo de interface desenvolvido pela *National Marine Electronics Association* (NMEA, 2002) define ambos requisitos elétricos de sinal e formato das mensagens para viabilizar a comunicação, com foco na configuração “um produtor, múltiplos consumidores”, entre dispositivos eletrônicos comumente empregados em aplicações marítimas, entre eles, o GPS. O estilo de formatação de mensagens NMEA acabou se popularizando e se tornando o principal padrão de interfaceamento com GPS comerciais para aplicações gerais.

O padrão estipula o uso de mensagens com caracteres ASCII (Código Padrão Americano para o Intercâmbio de Informação) seguindo o formato “\$aacc,c-c*hh<CR><LF>”, cujo significado, para cada bloco de caracteres, é apresentado no Quadro 7.

Quadro 7 – Formatação NMEA.

ASCII	Descrição
\$	Começo de uma sentença.
aacc	Campo de endereçamento. O "aa" identifica o produtor, enquanto o "ccc" descreve a informação contida na mensagem.
,	Delimitador. Este caractere delimita o começo de um novo campo, com exceção do campos de endereço e CRC.
c-c	Este campo representa a sequência de informação efetivamente transmitida. Consiste de uma série de dados, também em ASCII, separada por caracteres delimitadores. Estes dados não possuem identificadores e deverão ser interpretados por meio de sua ordem de acordo com o que é estabelecido para uma mensagem do tipo “ccc”.
*	Delimitador de <i>check sum</i> . Este caractere representa que os dois caracteres subsequentes contêm o check sum do pacote.
hh	O <i>check sum</i> é calculado por meio de uma operação OR exclusivo entre todos os bytes da dado da mensagem, ou seja, entre os delimitadores \$ e *. Cada nibble do resultado é então convertido para um caractere ASCII (0-9, A-F (upper case)).
<CR><LF>	Os caracteres <i>carriage return</i> e <i>line feed</i> indicam o fim do pacote.

Fonte: (NMEA, 2002).

Especificamente, para mensagens GPS os caracteres “aa” de identificação do produtor são “GP”. Quanto aos identificadores de informação da mensagem, os mais comuns são os apresentados no Quadro 8. Algumas informações são bem particulares como dados sobre data (ZDA), velocidade (VTG) e informações sobre o próprio sistema de comunicação (MSS e GSV),

enquanto outros são variações das informações de localização e data.

Para a aplicação de rastreamento de veículos, são necessários dados de posicionamento, estampa de tempo e velocidade (em km/h). Estas podem ser obtidas por meio das mensagens RMC e VTG. Além disso, valores de erros de fixação presentes na mensagem GSA podem ser utilizados para checar a validade dos dados obtidos.

Quadro 8 – Mensagens NMEA para GPS (QUEC-TEL, 2016)

Campo "ccc"	Descrição	Dados
GGA	Dados de fixação do sistema de posicionamento global	Horário, posição e status de fixação.
GLL	Latitude e Longitude	Latitude, longitude, UTC da fixação de posição e status da fixação.
GSA	Diluição de precisão e satélites ativos no GPS	Modo de operação do receptor GPS, satélites utilizados e DOP.
GSV	GPS Satélites visíveis	Satélites ao alcance, elevação, azimute e valores de SNR.
MSS	Sinal do receptor MSK	SNR, intensidade do sinal, frequência e taxa de bit do receptor.
RMC	Sentença GPS mínima recomendada	Horário, data, posição, direção do curso e velocidade.
VTG	Curso e velocidade sobre a superfície	Curso e velocidade em relação ao solo.
ZDA	Data e hora	UTC, dia, mês, ano e fuso horário

Fonte: (NMEA, 2002).

O protocolo NMEA permite ainda mensagens proprietárias, sob a condição de que estas devem fazer uso obrigatório do identificador “P” ao início da sentença, fazendo com que o campo de endereçamento possua apenas 4 caracteres disponíveis.

2.5 ACELERÔMETRO

Sensores de aceleração, ou acelerômetros, são dispositivos que fornecem um sinal elétrico proporcional à aceleração linear à qual são submetidos. São dispositivos inerciais, isto é, medem a força inercial provocada pela aceleração do corpo ao qual estão fixados (ALBUQUERQUE, 2009). Eles podem ser usados em aplicações de monitoramento de condições de máquinas a partir de vibração, medição de ângulos de lança em guindastes, medição de ângulos de inclinação em máquinas de construção, sistemas de navegação inerciais de navios, aviões e satélites, sistemas de liberação de *airbag*, sistemas de suspensão ativa, monitoramento de atividade sísmica, monitoramento médico de pacientes, sistemas de realidade virtual, entre outras (ALBUQUERQUE, 2009; BEEBY *et al.*, 2004).

Todos os acelerômetros compartilham a mesma estrutura básica que consiste em uma

massa inercial suspensa por uma mola. Eles diferem nos mecanismos utilizados para medir a posição relativa dessa massa conforme ela se altera pela aplicação de uma aceleração externa (MALUF; WILLIAMS, 2004). A relação entre essa posição e a aceleração é dependente das variáveis mecânicas como o coeficiente de elasticidade da mola e o coeficiente de viscosidade do meio – geralmente escolhido de forma a aumentar o amortecimento e evitar oscilações em estado transiente (ALBUQUERQUE, 2009).

Métodos comuns de detecção da posição relativa da massa são o capacitivo e o piezoelétrico. No primeiro, a massa forma um dos lados de um capacitor de placas paralelas. Assim, a alteração da distância entre as placas altera a capacitância. Um circuito eletrônico é utilizado para medir essas variações de capacitância, da ordem de pF , e transformar em uma saída de tensão amplificada. No segundo método, a massa é piezoelétrica ou contém um filme piezoelétrico anexado, fornecendo uma tensão diretamente proporcional à deformação do mesmo que também é posteriormente amplificada (MALUF; WILLIAMS, 2004).

Com a expansão do campo dos MEMS (em português, sistemas microeletromecânicos), em especial transdutores microeletromecânicos, o custo de produção dos acelerômetros abaixou e tornou possível a produção de componentes menores, com uma maior gama de aplicações (BEEBY *et al.*, 2004). Afinal, um requisito importante em um acelerômetro é que sua massa não pode ser significativa em relação ao corpo cuja aceleração se deseja medir.

Para confecções de acelerômetros com medições em todas as dimensões, x, y e z, três acelerômetros são combinados de forma ortogonal. As principais especificações de um acelerômetro comercial são a escala completa, normalmente fornecida em relação à aceleração da gravidade G , a sensibilidade, em V/G , a resolução, também em G , a largura de banda em Hz, e a sensibilidade cruzada entre os eixos, isto é, quanto a medição em um eixo é afetada pela aceleração de seus eixos ortogonais. Além disso, em algumas aplicações, a imunidade ao choque pode ser importante. A escolha do acelerômetro ideal depende da aplicação a que é destinado. Por exemplo, enquanto o monitoramento de motores industriais necessita um fundo de escala próximo a $1G$, uma resolução muito alta, da ordem de $100\mu G$ e uma largura de banda maior que 10kHz, sensores de batida para air-bag necessitam uma escala maior, de até 50G, com uma largura de banda de somente $1kHz$ (MALUF; WILLIAMS, 2004).

Como a aplicação para que o acelerômetro foi usado neste projeto não possuía nenhum requisito muito crítico, este componente não teve critério decisivo na escolha do dispositivo de rastreamento IoT utilizado como base. O acelerômetro de três eixos embutido no Dragino

LGT-92 é contido no módulo MPU-9250, que também contém giroscópio e bússola. Segundo o *datasheet* (INVENSENSE, 2016), esse acelerômetro apresenta escala configurável (entre $\pm 2g$, $\pm 4g$, $\pm 8g$ e $\pm 16g$), largura de banda de até 1,13kHz, conversores analógico-digital de 16 bits, sensibilidade cruzada de cerca de 2% e capacidade de suportar choques de até 10.000G.

Neste trabalho, o acelerômetro foi utilizado para detectar se o veículo está parado ou em movimento, a partir da vibração existente no mesmo pela movimentação no asfalto. Além disso, é medido o ângulo de inclinação do patinete em relação à superfície, para identificação de possíveis acidentes caso o patinete se encontre na horizontal por um determinado tempo.

2.6 NODEJS

O servidor de aplicação deste trabalho foi desenvolvido para *NodeJs*, um ambiente de código aberto e multiplataforma para execução de *scripts* em linguagem *Javascript*. Um programa sendo executado pelo *Node.JS* utiliza somente um processo do sistema operacional, sem criar uma nova *thread* para cada requisição, o que é possibilitado por um conjunto de funções de entrada e saída não bloqueantes (FOUNDATION, 2021). Com este núcleo não bloqueante, o *Node.JS* é voltado à programação assíncrona, isto é, o programa sendo executado não espera até que operações de entrada e saída por meio da rede ou no disco sejam finalizadas para seguir para o próximo comando. Em vez disso, é dado prosseguimento ao *script* e os dados referentes a essa operação são tratados após a resposta ter sido recebida – o tratamento é especificado por meio do registro de funções de *callback*. Sendo assim, pode-se dizer que a linguagem *Javascript*, orientada a funções, e a execução assíncrona pelo ambiente *Node.Js* favorecem um paradigma de programação orientada a eventos (TILKOV; VINOSKI, 2010).

2.7 BANCO DE DADOS

Segundo Elmasri e Navathe (2005), um banco de dados é uma coleção lógica e coerente de dados com algum significado inerente, sendo projetado, construído e povoado por dados para uma proposta específica, atendendo um grupo de usuários definido e aplicações preconcebidas de acordo com o interesse desse grupo. As principais diferenças, segundo o mesmo autor, da abordagem de um banco de dados em relação ao processamento de arquivos são: a natureza autodescritiva do banco de dados; abstração dos dados e isolamento entre estes e os programas; suporte para as múltiplas visões dos dados; compartilhamento de dados e processamento de tran-

sações de multiusuários. A abstração e descrição padronizada de dados de forma que possibilite o compartilhamento é um requisito deste projeto e, portanto, a utilização de um banco de dados foi escolhida para o armazenamento das informações do sistema.

A classe mais tradicional de banco de dados é a de bancos de dados relacionais, baseados no modelo relacional introduzido por E. F. Codd em 1970. Esses sistemas são também conhecidos como sistemas SQL por utilizarem em maioria Linguagem de Consulta Estruturada (SQL) como linguagem de consulta (HALEEMUNNISA; KUMUD, 2016). A tendência recente é a transição para bancos de dados não relacionais (RINALDI *et al.*, 2019), ou NoSQL, que são bancos de dados livres de esquemas, facilmente escaláveis horizontalmente (replicáveis em outros nós computacionais) e livres do processo de *join* para busca de dados relacionados (RAUTMARE; BHALERAO, 2016). Esses sistemas surgiram para lidar com a quantidade massiva de dados provenientes do crescimento da IoT e da *Big Data*, no entanto, diversos trabalhos demonstram que não há um vencedor definitivo para o armazenamento de dados de sensores (HALEEMUNNISA; KUMUD, 2016).

Rautmare e Bhalerao (2016) compararam os tempos de inserção e consulta entre um banco de dados SQL e um NoSQL para uma aplicação IoT de pequena escala, conforme a quantidade de registros e *threads*, concluindo que em alguns cenários o banco de dados NoSQL tem um tempo de resposta menor em comparação ao banco relacional, porém este apresentou um comportamento mais estável. Os tempos de resposta, em todos os testes, não diferiram de forma significativa para uma aplicação não crítica em relação ao tempo. Rinaldi *et al.* (2019) demonstraram que a escolha de um modelo de dados apropriado pode impactar significativamente a performance de um banco de dados não relacional utilizado para implementação de um TSDB (em português, banco de dados para séries de tempo), uma vez que a estrutura dos metadados associados aumentou o tempo de consulta em até 6 vezes nos testes realizados. Por outro lado, os resultados apresentados por Haleemunnisa e Kumud (2016) mostram que os sistemas NoSQL e NewSQL (que misturam os dois paradigmas) podem apresentar um desempenho melhor em diversas situações.

Para este trabalho, por se tratar de uma aplicação não crítica e sem uma quantidade massiva de dados, o banco de dados relacional foi escolhido por apresentar maior robustez e segurança devido aos vários anos de desenvolvimento, além dos autores terem mais familiaridade com essa topologia. No entanto, a arquitetura de *software* utilizada, isolando os serviços de comunicação com o banco de dados para cada entidade e representando cada uma delas por

um objeto modelo na linguagem de programação escolhida, por meio do mapeamento objeto-relacional (ORM), facilita uma possível transição para um banco de dados não relacional.

2.8 API RESTFUL

Quando um *software* deve ser usado por outro *software*, ele precisa de uma Interface de Programação de Aplicação (API). As APIs fornecem um jeito simples de se conectar a um programa e interagir com o mesmo. Mais precisamente, elas são usadas para se construir sistemas distribuídos cujos componentes são fracamente acoplados. Serviços web são métodos de comunicação padronizada entre dois computadores por meio de uma rede para integração de aplicações Web, geralmente disponibilizado a partir da conexão a um Identificador Uniforme de Recurso (URI). Sendo assim, uma *API Web* é um modelo de desenvolvimento para um serviço web. Existem diversas padronizações para APIs Web. Uma delas faz uso do modelo REST (Transferência Representacional de Estado), comumente chamada de RESTful API (BIEHL, 2016).

Fielding (2000) definiu as APIs do tipo RESTful pela primeira vez, determinando, entre outras características, que elas devem utilizar a topologia cliente-servidor e não devem ter considerações de estado, isto é, cada interação entre o cliente e o servidor deve conter toda a informação necessária e não deve depender de transações anteriores. Esta última propriedade induz a visibilidade, confiabilidade e escalabilidade aos programas, segundo o autor. Além disso, o recurso central que distingue esse tipo de API das demais é a ênfase na uniformidade das interfaces entre os componentes. Perdendo-se em eficiência, ganha-se simplicidade, visibilidade das interações e desacoplamento entre a implementação e os serviços prestados.

Uma forma de implementar uma API Restful é utilizar métodos de requisição do protocolo *Hypertext Transfer Protocol* (HTTP) para representar as interações com os recursos do serviço. Dessa forma, o método “GET” serve para ler as representações dos recursos, o método “PUT” para criar um novo recurso, o método “DELETE” para excluir o recurso do serviço, o método “POST” para modificar as informações do recurso e o método “HEAD” para obter meta-informações sobre o recurso. A interação com cada tipo de recurso é feita a partir de uma URI específica atrelada ao servidor que executa a API (SUBRAMANIAN; RAJ, 2019).

Neste trabalho, uma API do tipo RESTful foi desenvolvida para possibilitar a interação com os recursos do sistema de monitoramento dos patinetes, desacoplando o mesmo do desenvolvimento da aplicação final do cliente. A arquitetura do sistema é apresentada na Seção 3.6 e a

documentação da API no Apêndice E.

3 DESENVOLVIMENTO DO SISTEMA LORAWAN DE MONITORAMENTO DA FROTA DE VEÍCULOS DO LACTEC

Para desenvolver o projeto, a abordagem da solução baseada em problemas foi seguida. Nesse método, primeiramente são identificados os problemas que o cliente precisa solucionar e quais são as dores causadas pela ausência da solução. Em seguida, são percebidas necessidades tecnológicas que poderiam resolver esses problemas. Por fim, os requisitos técnicos funcionais e não funcionais que suprem essas necessidades são elaborados, gerando uma especificação de requisitos que pode originar um projeto que resolve o problema em questão. Após o projeto da solução, segue-se à construção ou implementação e, enfim, os testes. Todas essas etapas podem ser realizadas de forma iterativa, isto é, não é necessário ter todos os requisitos para que se siga ao projeto – o que identificaria um modelo de desenvolvimento em cascata. A cada iteração, novos requisitos podem ser identificados, gerando adições e adaptações no projeto e na implementação, além de novos testes a serem realizados. O documento de especificação de problemas, necessidades e requisitos se encontra no Apêndice A.

3.1 SISTEMA ATUAL DE VEÍCULOS DO LACTEC E PROPOSTAS DE MELHORIAS

O sistema de veículos alternativos do Lactec foi implantado para facilitar a locomoção dos funcionários, incluindo bolsistas e estagiários, entre as sedes da empresa. A norma administrativa NA-503 estabelece os procedimentos para a utilização adequada das bicicletas e patinetes elétricos. Nas disposições da norma, encontram-se as seguintes informações, dispostas aqui de forma resumida:

1. Cabe à área de gerência de instalações da empresa, além de outras atribuições: manter os veículos elétricos em condições de utilização, providenciando manutenções para maximizar a vida útil dos veículos e garantir a disponibilidade dos mesmos; proceder verificações e tomar providências quanto a eventuais danos causados aos veículos; manter uma base de informações referente à operação e manutenção dos veículos; realizar estudos anuais para definir a necessidade de substituição ou locação de novos veículos elétricos.
2. A análise de eventuais processos envolvendo veículos elétricos, como multas, acidentes de trânsito ou danos causados ao equipamento por erro, imperícia ou negligência do condutor é atribuída à gerência do segmento a que o usuário do veículo pertence.

3. É responsabilidade do condutor do veículo elétrico zelar pela segurança e preservação do veículo elétrico, utilizando-o exclusivamente em serviço e dentro das rotas estabelecidas na norma (Figura 15) e, ao final da utilização, conectá-los à estação de carregamento disponível na unidade do Lactec (Fotografia 1). Em caso de acidentes, cabe ao usuário comunicar a ocorrência ao setor de instalações, sem abandonar o local do ocorrido.

Figura 15 – Rotas obrigatórias para uso dos veículos alternativos do Lactec.



Fonte: Intranet do Lactec.

Fotografia 1 – Estação de carregamento na sede do Lactec.



Fonte: Autoria própria.

Como pode ser observado, para execução das competências pelos responsáveis definidos na norma, é necessária uma base de informações a respeito do funcionamento dos veículos elétricos e do uso deles pelos funcionários.

O sistema eletrônico das estações de carregamento dos veículos realizam o controle de acesso dos mesmos pelos usuários a partir da leitura de seus crachás. As informações coletadas

nas estações são armazenadas e, assim, é possível saber quando e por quem um patinete foi retirado em uma estação e quando ele foi devolvido em uma outra estação. No entanto, não existia nenhuma informação gerada durante o percurso com o veículo. Portanto, seria impossível saber se um usuário saiu da rota pré-estabelecida, por exemplo. Além disso, nenhum dado era salvo em relação às baterias dos veículos, impossibilitando a análise do comportamento de carga e descarga para detectar eventuais padrões indicativos de necessidades de manutenções preditivas ou de manutenções corretivas antes que prejuízos maiores fossem causados. Por fim, as estatísticas de uso dos patinetes eram geradas somente pela empresa fornecedora do sistema, Emove¹, e eram, portanto, limitadas aos parâmetros fornecidos pela mesma, sendo eles: quantidade de retiradas em cada estação no período, tempo médio de utilização, separado por categorias (menor que 30 minutos, até 1 hora e mais que 1 hora), quantidade de utilizações em dias de semana, quantidade de utilizações em fins de semana, quantidade de utilização por faixa horária (de uma em uma hora) e quantidade de utilizações por usuário. Sendo assim, era impossível realizar buscas mais elaboradas para definir, por exemplo, quais usuários utilizaram um determinado veículo dentro de um determinado período e como procedeu esse uso.

Para solucionar o problema da falta de monitoramento dos veículos em percurso, o sistema de rastreamento apresentado neste trabalho foi proposto, com armazenamento das informações de GPS dos dispositivos. Com este sistema, se um veículo sai da rota pré-estabelecida, um aviso sonoro é emitido pelo dispositivo IoT e uma notificação é enviada a um responsável cadastrado. As informações das baterias, do veículo e do dispositivo IoT, são armazenadas tanto durante o percurso quanto durante o carregamento nas estações. Com todas as informações no banco de dados disponíveis para o Lactec, variados tipos de análise para definições mercadológicas sobre o futuro do sistema e melhoria da experiência dos usuários, ou até investigações sobre usos indevidos do sistema, podem ser realizadas, não limitando-se somente às estatísticas informadas pela fornecedora dos veículos.

Além da solução dos problemas citados, outras funcionalidades foram propostas, como: a detecção de possíveis acidentes a partir da permanência do veículo em posição horizontal por um determinado tempo, com geração de notificação para o e-mail de um responsável cadastrado, adiantando o contato da gerência de instalações que deve ser realizado pelo condutor; a contagem do tempo de percurso em que o veículo está parado e em movimento, para fins de análise do tempo de espera em semáforos e de deslocamento efetivo; a determinação de uma velocidade

¹ Site da empresa: <https://www.emove.mobi/>. Acesso em 18 de julho de 2021.

limite que deve ser respeitada pelos usuários, com emissão de aviso sonoro pelo dispositivo de monitoramento caso ela seja ultrapassada.

Como já foi citado, uma descrição mais detalhada dos problemas, necessidades tecnológicas e requisitos técnicos para implementação dessas funcionalidades é apresentada no Apêndice A.

3.2 ARQUITETURA DO SISTEMA

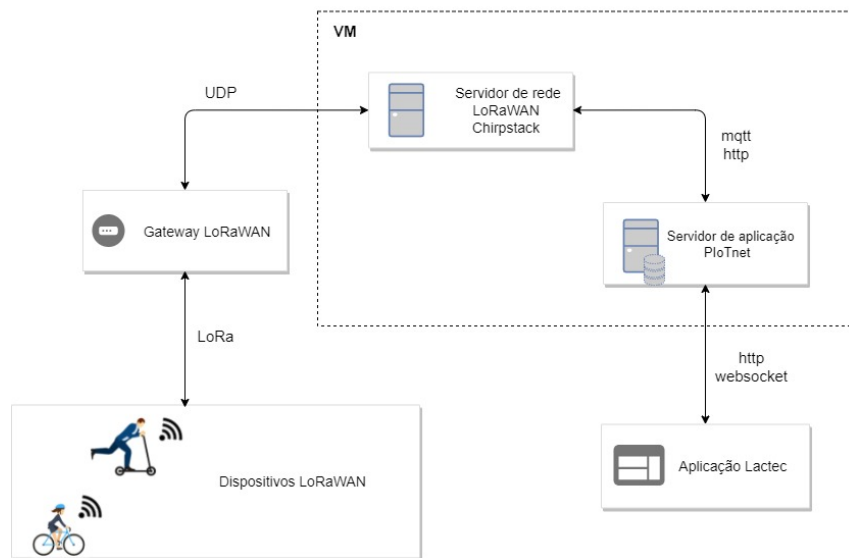
Com base nas justificativas já citadas neste trabalho e nos requisitos apresentados no Apêndice A, decidiu-se pelo esquema apresentado na Figura 16 para a arquitetura do sistema. Como já mencionado, a pilha *LoRaWAN* escolhida para o servidor foi a da comunidade *Chirpstack*. Sua estrutura tem uma peculiaridade: o serviço do *Network Server* encaminha todos os dados de aplicação para um mesmo serviço de *Application Server*, para permitir algumas funcionalidades comuns a todas as aplicações por meio de uma página *Web* e possibilitar um gerenciamento completo do sistema. Sendo assim, os demais servidores de aplicação que vão utilizar essa infraestrutura *LoRaWAN* devem se comunicar, neste caso, com esse servidor de aplicação da *Chirpstack*, e não com o *Network Server* como é previsto na arquitetura *LoRaWAN*. Esse servidor de aplicação possui diversas interfaces, como apresentado na arquitetura da pilha *Chirpstack* presente na Figura 17.

Em nosso diagrama de arquitetura (Figura 16), o *Chirpstack Network Server* e o *Chirpstack Application Server* foram encapsulados no mesmo bloco chamado “Servidor de rede *LoRaWAN Chirpstack*”, pois é dessa forma que eles serão enxergados por nossa aplicação, que simplesmente se comunica com esse bloco por meio de um *broker* MQTT e uma API em HTTP. Foi decidido que o escopo do projeto não iria incluir o desenvolvimento do *front end*, isto é, de uma aplicação *desktop*, *mobile*, ou página *Web* para o usuário final. Assim, o servidor de aplicação desenvolvido teria uma interface HTTP para acesso às funcionalidades do sistema, além de uma interface *WebSocket* para disponibilização dos dados em tempo real. Dessa forma, o modo como os dados são apresentados e tratados pode ser decidido pelo cliente do sistema.

O bloco “Servidor de aplicação *PIoTnet*” foi totalmente construído pelos autores, conforme explicado na Seção 3.6. Para realização dos testes do sistema e apresentação dos dados neste trabalho foi utilizado o *software* Grafana, que permite o acesso aos dados de diversas formas, incluindo por APIs do tipo Restful, usada neste caso. Os desenvolvimentos de *hardware* e *firmware* para o bloco “Dispositivos *LoRaWAN*” são apresentados na Seção 3.4. Para implantação

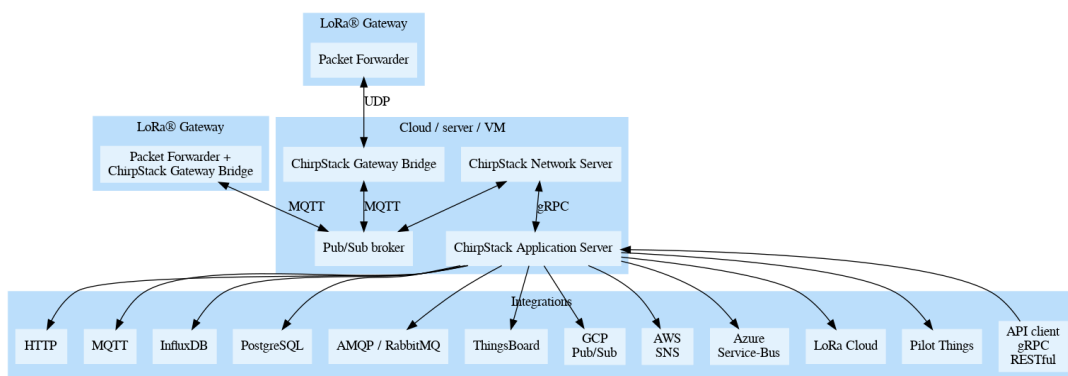
do sistema, toda a parte de *back-end* do servidor, incluindo todos os *softwares* da pilha *Chirpstack* e o servidor de aplicação próprio, foram instalados em uma máquina virtual do Lactec com sistema operacional *Ubuntu Server 18.04*. Um espaço de 10GB em disco foi reservado para o armazenamento de dados do sistema. Para uma persistência de cerca de 5 anos, considerando 10 percursos de 30 minutos sendo realizados por dia com a maior taxa possível de envio de dados, 350MB já teria sido suficiente.

Figura 16 – Diagrama de bloco da arquitetura do sistema de monitoramento de veículos.



Fonte: Autoria própria.

Figura 17 – Arquitetura da pilha Chirpstack.



Fonte: Chirpstack Community.

3.3 PROTOCOLO

Como o padrão *LoRAWAN* não define a camada de aplicação, foi necessário estabelecer um protocolo próprio para a comunicação entre o dispositivo de monitoramento e o servidor de aplicação.

Para isso, explorou-se o conceito de portas presente no *LoRaWAN*. Conforme a aplicação, as portas de número 1 a 223 estão livres para uso (LORA ALLIANCE, 2020). Em nosso sistema, as portas partindo de valores mais baixos e mais altos foram empregadas em *uplinks* e *downlinks* respectivamente. Além disso, as mensagens originadas no dispositivo com destino ao servidor seguem o protocolo *CayenneLPP* apresentado na Seção 2.3.

Foram criados três tipos de mensagem de *uplink*, um para cada estado do dispositivo de monitoramento (Figura 27). As informações de portas (*LoRaWAN* e *CayenneLPP*), configuração de confirmação de recebimento da mensagem *LoRaWAN* e detalhamento dos dados enviados se encontram no Quadro 9.

Quadro 9 – Mensagens de *uplink*.

Uplinks	Porta LoRaWAN	Confirmada	Dado	Tipo de dado CayenneLPP	Porta CayenneLPP	Descrição	
Keep Alive	4	Sim	Bateria do node	Voltage	0	Lida pelo do ADC	
			Timestamp	Unix timestamp	0	Lida do RTC	
Recarga	3	Não	Bateria do node	Voltage	0	Lida pelo do ADC	
			Bateria do veículo	Voltage	1	Lida pelo do ADC	
Rota	2	Não	Timestamp	Unix timestamp	0	Lida do RTC	
			Latitude e longitude	GPS	0	Recebida na mensagem de GPS NMEA RMC	
			Bateria do node	Voltage	0	Lida pelo do ADC	
			Bateria do veículo	Voltage	1	Lida pelo do ADC	
			Velocidade	Velocity	0	Recebida na mensagem de GPS NMEA VTG	
			Erro do GPS	GPSError	0	Módulo dos erros hdop, pdop e vdop recebidos na mensagem de GPS NMEA GSA	
			Flags	Generic Byte	0	Bit 0	1 - em movimento 0 - parado
						Bit 1	0 - GPS fixado 1 - GPS não fixado
						Bit 2	1 - possível acidente 0 - sem acidente

Fonte: Autoria própria.

Para acionar o *buzzer* ao se detectar alguma anormalidade na operação do veículo, alterar valores de parâmetros de funcionamento do sistema e reiniciar o dispositivo, os comandos de *downlink* detalhados no Quadro 10 foram desenvolvidos. Diferentemente das mensagens *uplinks*, os *downlinks* não usam o protocolo *CayenneLPP*. A identificação da informação é feita somente com base na porta *LoRaWAN*.

Todas as mensagens de *downlink* possuem confirmação de recebimento. No caso das mensagens de *uplink*, apenas a mensagem de *keep-alive* possui confirmação. A perda de algumas mensagens de recarga e rota é menos crítica do que a perda de um comando de configuração. Além disso, estas mensagens possuem um período de envio bem reduzido e a exigência de

Quadro 10 – Mensagens de *downlink*.

Downlinks	Porta LoRaWAN	Confirmada	Dado	Descrição		
Configuração de limiar de erro do GPS	217	Sim	Limiar para considerar o erro de fixação do GPS como aceitável ou não	byte 0-3	Float	
Alarme	218	Sim	Estado do buzzer	byte 0	1 - aciona o buzzer 0 - desaciona o buzzer	
Veículo fora de área	220	Sim	Informação de área ilegal	byte 0	1 - área ilegal 0 - área legal	
Configuração de período de transmissão	221	Sim	Configuração	byte 0-1	Período de rota [s]	Enviar 0 para manter o valor atual
				byte 2-3	Período de recarga [s]	
				byte 4-5	Período de keep alive [s]	
Configuração de limite de velocidade	222	Sim	Velocidade	byte 0-3	Float	
Firmware reset	223	Sim	-	-	Reseta o dispositivo	

Fonte: Autoria própria.

confirmação de cada pacote pode inclusive prejudicar o desempenho da rede. A confirmação da mensagem de *keep-alive* é viável neste caso, visto que é transmitida com um período muito maior do que as demais.

3.4 DISPOSITIVO DE MONITORAMENTO

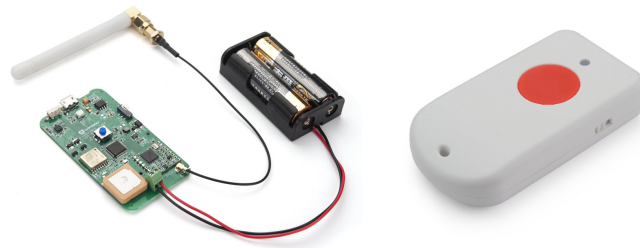
3.4.1 Adaptação do *hardware*

O dispositivo acoplado ao veículo foi baseado na solução comercial aberta LGT92 da *Dragino Technology Co.* Este módulo apresenta de forma integrada os componentes que possibilitam a comunicação por meio de *LoRa*, a coleta de informações de geolocalização e dados de aceleração, sendo os principais deles:

- Acelerômetro de 9 eixos do tipo microeletromecânico capacitivo, MPU9250.
- Módulo de comunicação GPS L76-L.
- Microcontrolador L072CZT6, de arquitetura ARM, com 192kB de memória Flash, 20kB de memória RAM e 6kB de memória EEPROM.
- Circuito integrado SX1276 para comunicação via modulação *LoRa*.

Algumas modificações foram feitas para a inserção de novas funcionalidades, como o aviso sonoro e a leitura de tensão. Além destas, alterações objetivando a integração do dispositivo com o veículo também foram necessárias.

Figura 18 – Módulo Dragino LGT92.



Fonte: Dragino (2019a).

Alimentação

O LGT92 possui duas versões, uma alimentada por pilhas AA (LGT-92-AA) e outra com bateria de lítio recarregável (LGT-92-Li). Esta última foi escolhida visando uma estratégia de *backup* para o sistema de monitoramento caso a alimentação do veículo apresente alguma falha, como, por exemplo, descarregamento intenso por causa desconhecida, defeitos na própria bateria ou até uma tentativa de desacionar o monitoramento, cortando-se a alimentação pela bateria do veículo, em caso de furto. O valor nominal da bateria do veículo é de 40V, mas, ao ser totalmente carregada, pode atingir 42V. Já o dispositivo de monitoramento requer uma tensão de 5V para alimentar o CI CN3052A responsável pelo carregamento da bateria de 3,3 V. Um conversor DC-DC *step down* comercial baseado no CI LM2596 da Texas (Figura 19) foi utilizado para este fim.

Figura 19 – Módulo DCDC *step down*.



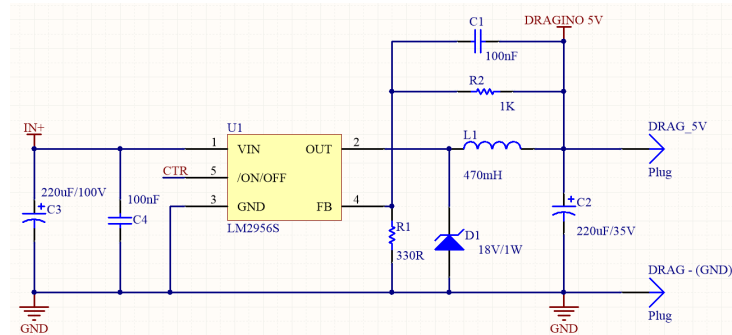
Fonte: Filipe Flop².

O *trimpot* foi substituído por um resistor de 330Ω (R1 na Figura 20), visto que a tensão

² Disponível em <https://www.filipeflop.com/produto/regulador-de-tensao-lm2596-conversor-dc-dc-step-down/>. Acesso em 3 de junho de 2020.

de saída deve estar fixa em 5V, e o circuito foi passado para outro *layout* contendo modificações de modo a poupar espaço.

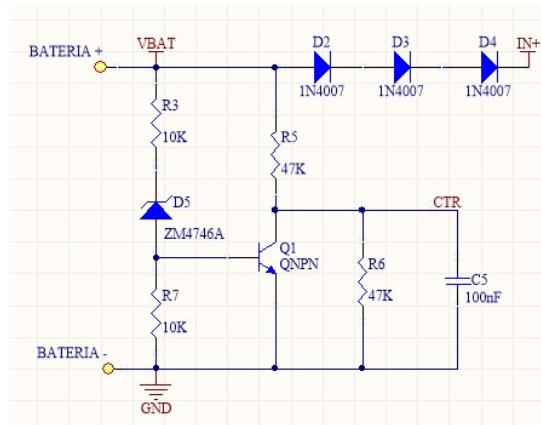
Figura 20 – Circuito DCDC *step down* fixo para 5V.



Fonte: Autoria própria.

Para proteger a bateria do veículo de ser descarregada abaixo do valor de 30V, um circuito de proteção de subtensão (TEXAS INSTRUMENTS, 2016) foi adicionado à lógica de alimentação (Figura 21). Além disso, como a máxima tensão de entrada do LM2596 é 40V, foram adicionados três diodos em série entre a bateria e a entrada de alimentação do CI para evitar a sobretensão, uma vez que os diodos geram uma queda de aproximadamente 2,1V quando existe corrente circulando da bateria para o circuito.

Figura 21 – Circuito de proteção de subtensão.



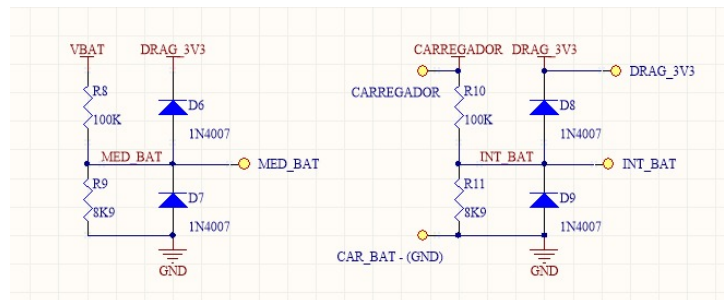
Fonte: Autoria própria.

Monitoramento de tensão

O circuito para a leitura da bateria do próprio módulo por meio de um pino ADC já estava implementado no projeto do LGT92 e otimizado para minimizar o consumo de bateria pelo divisor de tensão (DRAGINO, 2019b). No entanto, o valor da bateria do veículo e a conexão

desta na estação de carregamento também precisavam ser monitorados. Para isso, divisores de tensão em conjunto com diodos de proteção contra picos positivos e negativos foram acoplados aos pinos PA1 e PA3 do microcontrolador STM32L072cz, capazes de realizar leitura analógica. Os dois circuitos de atenuação e proteção são apresentados no esquemático da Figura 22.

Figura 22 – Circuito de monitoramento de tensão da bateria do veículo e estação de carregamento.



Fonte: Autoria própria.

Aviso sonoro

O aviso sonoro se dá por meio do acionamento de um *buzzer*. Este deve ser alimentado pela mesma bateria de 3,3V que alimenta o dispositivo para que seja capaz de emitir som mesmo quando a bateria do patinete não esteja fornecendo corrente. Além disso, o componente apresenta limitação quanto ao tamanho, visto que será colocado dentro da carcaça do patinete.

Tanto *buzzers* de acionamento por Modulação por Largura de Pulso (PWM) quanto de sinal contínuo foram testados. Maiores informações sobre os modelos testados encontram-se no Quadro 11.

Quadro 11 – Comparação de modelos de *buzzer*.

Modelo	MSR320R	GT0905A	AC1205G-N	TFM-51FA	CEM1203-42
Tensão [V]	3 ~ 20	4 ~ 6	4 ~ 8	0 ~ 30	3 ~ 5
Intensidade sonora [dB(A)]	73 ~ 86	85	85	75	85 ~ 95
Corrente [mA]	3 ~ 16	80	50	2	35
Modo	Contínuo	PWM	PWM	PWM	PWM

Fonte: Autoria própria.

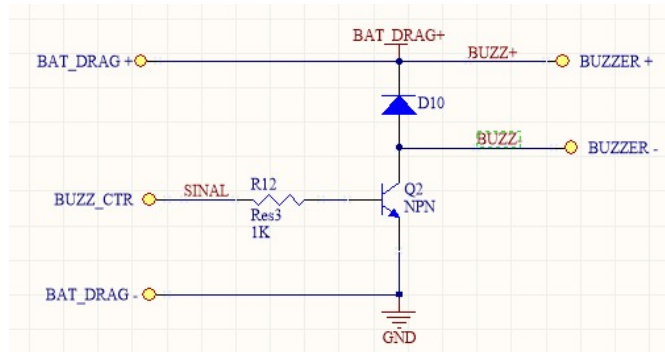
O modelo escolhido foi o CEM1203-42 da RoHS, uma vez que além de atender o requisito de faixa de tensão de operação, possui a maior intensidade sonora com um consumo de corrente relativamente baixo.

Estes modelos foram testados também empiricamente em relação à intensidade do som e o modelo escolhido apresentou também o volume mais alto nas condições de alimentação

definidas. A frequência de PWM utilizado foi ajustada para 2115Hz, um valor um pouco mais alto do que 2048Hz sugerido no datasheet (ROHS, 2006), mas chegou-se à conclusão de que a maioria das amostras respondia melhor a este valor do que ao nominal.

O circuito representado na Figura 23 foi desenvolvido para acionamento do *buzzer* pelo microcontrolador.

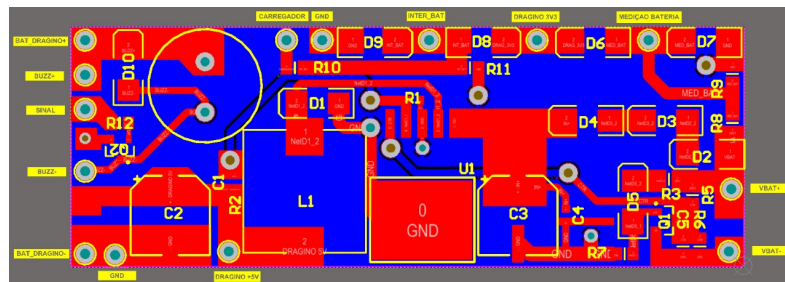
Figura 23 – Circuito de acionamento do *buzzer* CEM1203-42.



Fonte: Autoria própria.

Uma placa de circuito impresso contendo os circuitos descritos anteriormente foi desenvolvida e seu *layout* é apresentado na Figura 24.

Figura 24 – Circuito de alimentação, acionamento do *buzzer* e acoplamento com o ADC integrados na mesma PCB.



Fonte: Autoria própria.

Antena

Para aprimorar o acoplamento mecânico da antena de comunicação *LoRa* na carcaça do patinete, a antena original (Figura 25) do LGT92 foi removida e um conector ipex foi soldado para a utilização da antena SRFC011 para faixa de 900MHz (Figura 26).

Figura 25 – Antena original do módulo LGT92-Li.



Fonte: A autoria própria.

Figura 26 – Antena mais adequada para a associação mecânica com o veículo.



Fonte: A autoria própria.

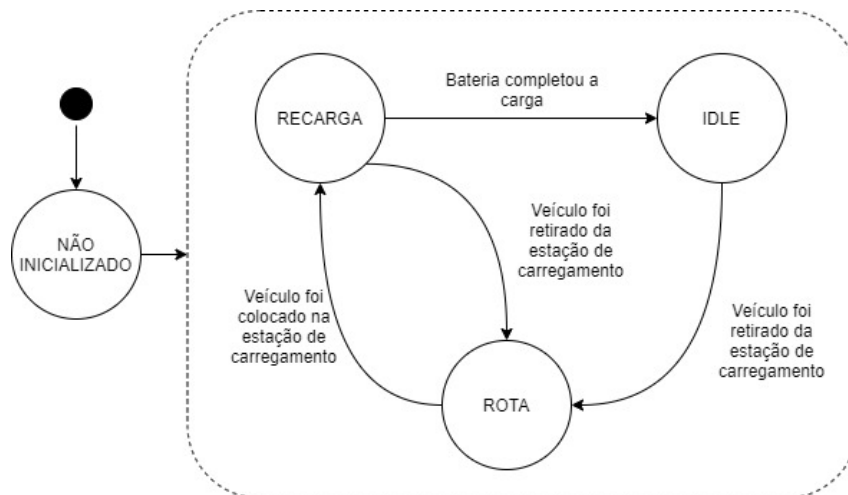
Reset

Para viabilizar a depuração do *firmware*, o pino de *reset* do STM32L072cz precisou ser estendido para conexão na ferramenta STLINK. A princípio, o link do microcontrolador se encontrava instável. Este problema foi resolvido por meio da adição do capacitor de desacoplamento em paralelo com o pino de *reset* sugerido pela ST (STMICROELECTRONICS, 2017), que não havia sido implementado no projeto da Dragino.

3.4.2 Firmware

Por meio da análise do sistema de mobilidade elétrica do Lactec, concluiu-se que o uso dos veículos, em condições normais, pode ser descrito por três estados: “rota”, “recarga” e “idle” (Figura 27). Cada um desses estados define quais variáveis do sistema devem ser monitoradas.

Figura 27 – Diagrama de estados do *firmware* do dispositivo de monitoramento.



Fonte: Autoria própria.

Ao inicializar, o sistema busca recuperar o estado anterior da aplicação por meio da leitura de informações que são salvas na EEPROM a cada alteração de estado/parâmetro de configuração. Caso o sistema esteja sendo inicializado pela primeira vez, ou um erro de CRC seja detectado na leitura dos dados de ambas regiões de salvamento, principal e redundância, os valores padrão são utilizados. Os dados armazenados e os valores padrão encontram-se especificados no Quadro 12. Também a cada inicialização, o GPS é acionado para que o sincronismo do relógio interno seja feito. Esse processo de sincronização ocorre pelo menos uma vez por dia.

Quadro 12 – Parâmetros de configuração e estado da aplicação do dispositivo de monitoramento salvos na EEPROM.

Dado	Tipo	Valor padrão
ROUTE_DATA_TX_PERIOD [ms]	uint32	5000
RECHARGE_DATA_TX_PERIOD [ms]	uint32	60000
KEEP_ALIVE_TX_PERIOD [ms]	uint32	3600000
BUZZER_STATE	uint32	NO_BEEP
LAST_STATE	uint32	IDLE
VELOCITY_LIMIT [m/s]	float	20,0
GPS_ERROR_THRESHOLD	float	3,0

Fonte: Autoria própria.

A retirada do veículo de uma das estações por um colaborador é a condição de transição

para o estado “rota”. Neste estado, os processos de GPS (Apêndice B) e do acelerômetro (Seção 3.4.2) devem ser iniciados para a coleta dos dados de movimentação, orientação, localização e velocidade. Leituras periódicas dos ADCs da bateria do dispositivo e do veículo também devem ser realizadas. Uma descrição mais detalhada dos dados monitorados foi apresentada anteriormente no Quadro 9.

Neste estado, em que o veículo está sendo efetivamente utilizado por um usuário, é possível disparar avisos sonoros. O acionamento do *buzzer* no caso de excesso de velocidade e bateria baixa é feito por iniciativa do próprio dispositivo que monitora esses valores constantemente. Já a detecção de transpasse de área é feita no servidor de aplicação e informada via *downlink*. O acionamento do alarme também se dá desta forma, visto que parte da iniciativa de um funcionário monitorando o sistema de mobilidade.

A diferenciação entre as possíveis notificações sonoras é feita por meio do número de bipes e sua duração. O retorno para a estação, estados “*idle*” e “*recarga*”, leva ao desacionamento do *buzzer*. Além disso, nesta etapa de desenvolvimento do sistema, o dispositivo tem autonomia para desacionar também os alertas iniciados por *downlink*, caso o servidor não seja capaz de fazê-lo, por exemplo, por alguma falha na comunicação. Esta escolha foi feita visando favorecer a experiência do usuário, evitando situações inconvenientes até que o sistema se prove confiável. Apenas um aviso soa por vez com base na prioridade da informação. A descrição das notificações sonoras e suas condições de acionamento e desacionamento encontram-se no Quadro 13 em ordem crescente de prioridade, ou seja, a notificação de bateria baixa é a menos prioritária.

Quadro 13 – Descrição dos alertas sonoros e condições de acionamento/desacionamento.

Número de bipes	Duração	Descrição	Acionamento		Desacionamento				
1	Curto	Bateria do veículo baixa	Estado de rota	Leitura do ADC abaixo de 34V	Transição de estado para <i>idle</i> ou recarga	-			
2	Curto	Acima do limite de velocidade		Erro do GPS abaixo do valor de limiar configurado e velocidade acima do valor de limiar configurado			OU	Sequência de dados de GPS com erro acima do aceitável	
3	Curto	Fora da área permitida		Downlink					Downlink
1	Longo	Alarme		Downlink					Downlink

Fonte: Autoria própria.

Ao ser colocado na estação, o dispositivo de monitoramento passa para o estado “*recarga*” ou “*idle*”, dependendo do nível de bateria do veículo. A aplicação permanece no estado de “*recarga*”, enviando periodicamente os valores de tensão das baterias, enquanto o valor da tensão da bateria do veículo for menor que 41,5V. Após atingir esse valor, o estado muda para “*idle*”, em que uma mensagem de *keep-alive*, contendo apenas a carga da bateria do dispositivo, é enviada em um período bem maior do que nos outros estados. O envio deste pacote tem como objetivo indicar que o dispositivo de monitoramento do veículo em questão se encontra ativo.

Uma descrição mais detalhada dos pacotes enviados nestes modos foi feita também no Quadro 9.

As mensagens de *downlink* podem ser enviadas ao dispositivo em qualquer um dos estados, mas dado que o dispositivo opera na classe A do padrão *LoRaWAN*, e só pode receber um pacote após um *uplink*, alguns estados, podem apresentar uma latência bem maior em comparação com os outros, como é o caso do “*idle*”, que por padrão, pode apresentar uma latência de 1 hora.

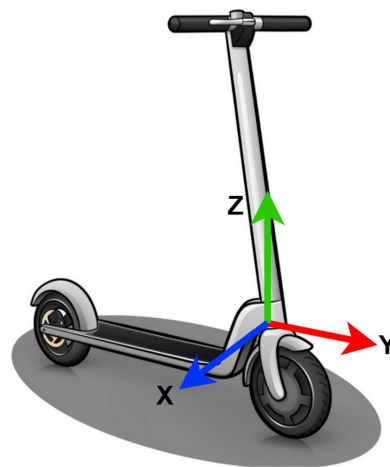
Cabe destacar que o desenvolvimento do *firmware* se deu a partir do código base para implementação de nós *LoRaWAN* em microcontroladores ST fornecido pela Semtech³. Uma visão mais detalhada dos estados e processos da aplicação integrada à pilha I-CUBE-LRWAN pode ser encontrada no Apêndice B.

A ferramenta de versionamento Git foi utilizada para controle das alterações feitas no código fonte e um repositório remoto está hospedado no GitHub na url <<https://github.com/GeovanaSlla/PIoTNetEndNode>>. A autorização dos autores é necessária para acesso ao repositório.

Detecção de movimento

O estado do veículo, quanto a se está parado ou em movimento, pode ser inferido por meio do monitoramento da vibração deste.

Figura 28 – Eixos do acelerômetro no patinete.



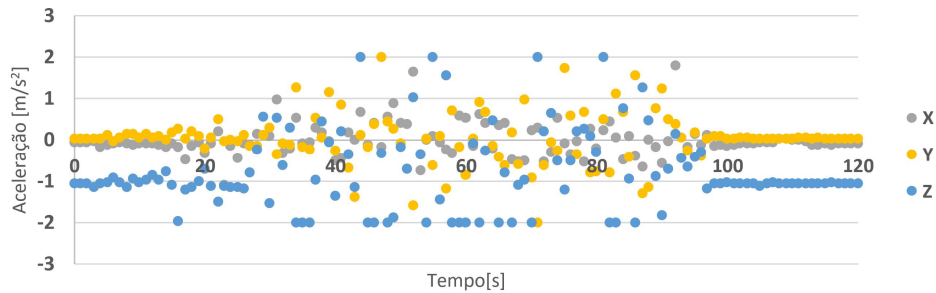
Fonte: Adaptado de Vectorstock⁴.

³ Disponível em: <https://www.st.com/en/embedded-software/i-cube-lrwan.html>. Acesso em: 27 de junho de 2021.

⁴ Disponível em <https://www.vectorstock.com/royalty-free-vectors/scooter-drawing-vectors>. Acesso em 3 de junho de 2020.

O dispositivo de monitoramento foi acoplado a um patinete com os eixos do acelerômetro como na Figura 28 e os valores de aceleração foram coletados. No gráfico, pode-se verificar que veículo iniciou a movimentação após cerca de 20 segundos e parou após cerca de 100 segundos do instante inicial representado.

Gráfico 1 – Aceleração em cada eixo com o veículo em movimento e parado.



Fonte: Autoria própria.

Por meio da distribuição normal dos valores coletados pode-se notar o aumento do desvio padrão que ocorre quando o veículo encontra-se em movimento.

Gráfico 2 – Distribuição normal das acelerações com o veículo parado.

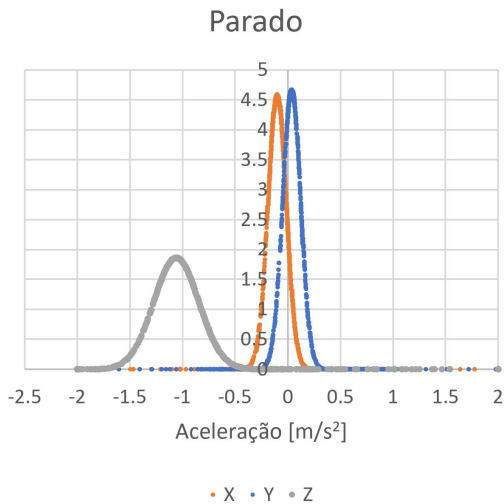
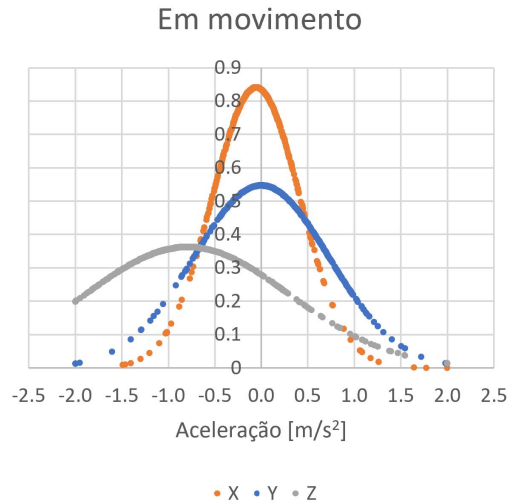


Gráfico 3 – Distribuição normal das acelerações com o veículo em movimento.



Fonte: Autoria própria.

Quando o veículo está em percurso, os valores de aceleração para cada eixo são lidos a cada 250ms e os últimos 10, compreendendo assim um período de 2,5 segundos, são armazenados em memória para o cálculo do desvio padrão por meio da Equação 7.

$$s = \sqrt{\frac{\sum(x_i - \bar{x})^2}{n - 1}} \tag{7}$$

Onde:

s = Desvio padrão

x_i = i -ésima amostra

\bar{x} = Média das amostras

n = Número de amostras

O valor obtido é então comparado com um limiar definido para a decisão do estado. Se o valor calculado for maior que o limiar, então o veículo está em movimento.

Detecção de acidente

Dado que o patinete tem como característica um equilíbrio instável em relação ao eixo X e que os valores de aceleração nesta direção apresentam o menor desvio padrão em uso regular, outra análise foi feita sobre estes dados para detectar situações anormais de utilização. O valor médio de aceleração no eixo X é atualizado a cada medição e se este permanecer fora de seu valor médio esperado ($a_x \approx 0m/s^2$), com uma diferença mínima estabelecida, por um período de 10 segundos, um alerta é disparado. Este valor anormal de aceleração determina que o patinete encontra-se em posição horizontal, o que pode significar um acidente ou manuseio incorreto do veículo.

3.4.3 Acoplamento

Como o dispositivo de monitoramento requer conexão com a bateria do veículo, foi necessário encontrar um local em que isso fosse possível e seguro. Além de que, como o *hardware* não possui nenhum tipo de encapsulamento robusto capaz de resistir à água, poeira, entre outros, optou-se por inserir o dispositivo dentro da estrutura do veículo. O local escolhido no patinete encontra-se destacado em vermelho na Fotografia 2.

Na Fotografia 3, pode-se visualizar como foi feito o encaixe do *hardware* na estrutura frontal do patinete. As antenas de GPS e do rádio *LoRa* encontram-se destacadas em amarelo e vermelho, respectivamente; o *buzzer*, em azul.

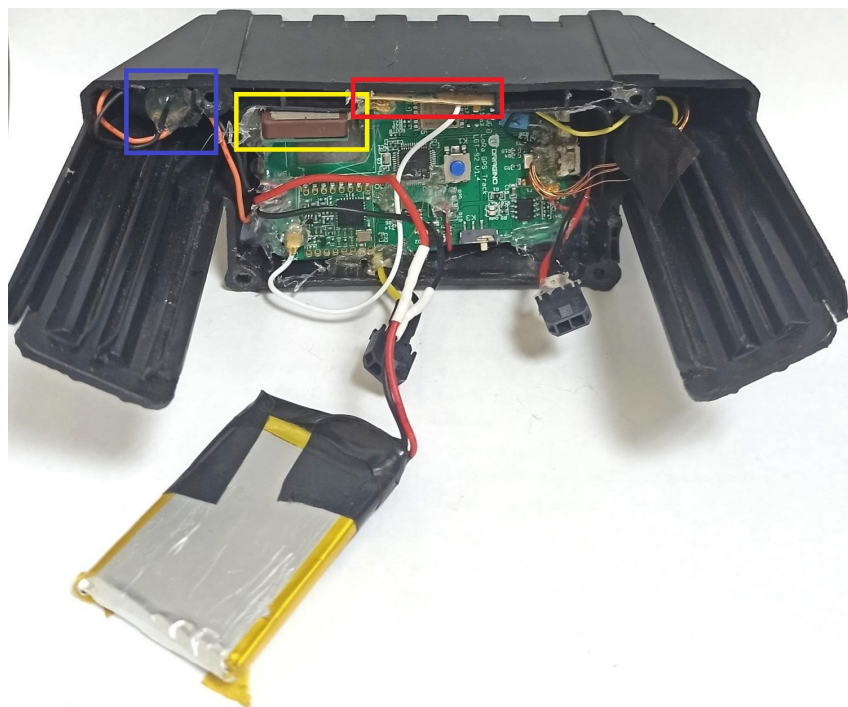
Para a conexão com a bateria e com os pinos de contato do carregador da estação, foi necessário modificar o cabeamento do patinete para a inserção de dois conectores do tipo *micro fit* destacados em amarelo nas Fotografias 4 e 5.

Fotografia 2 – Local escolhido para a instalação do dispositivo de monitoramento no patinete.



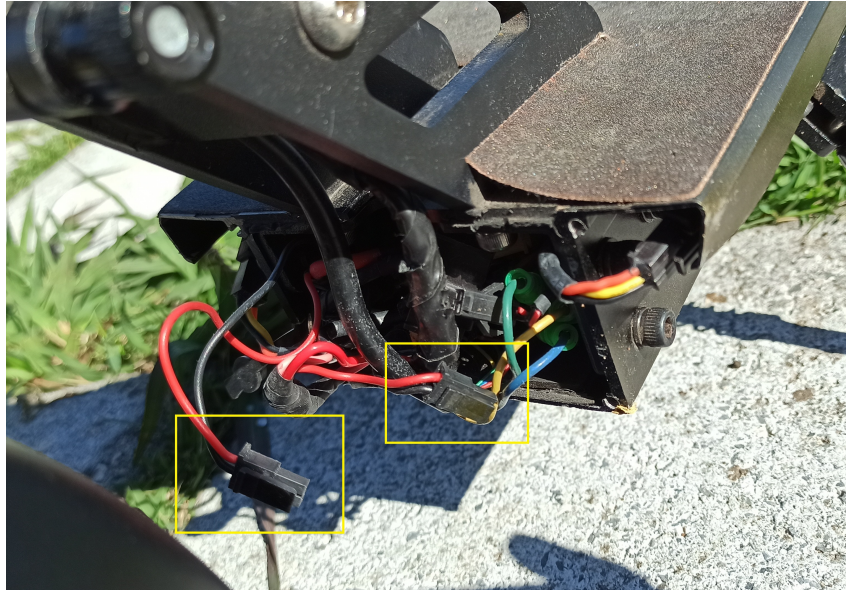
Fonte: Autoria própria.

Fotografia 3 – Acoplamento entre o hardware e estrutura frontal do patinete.



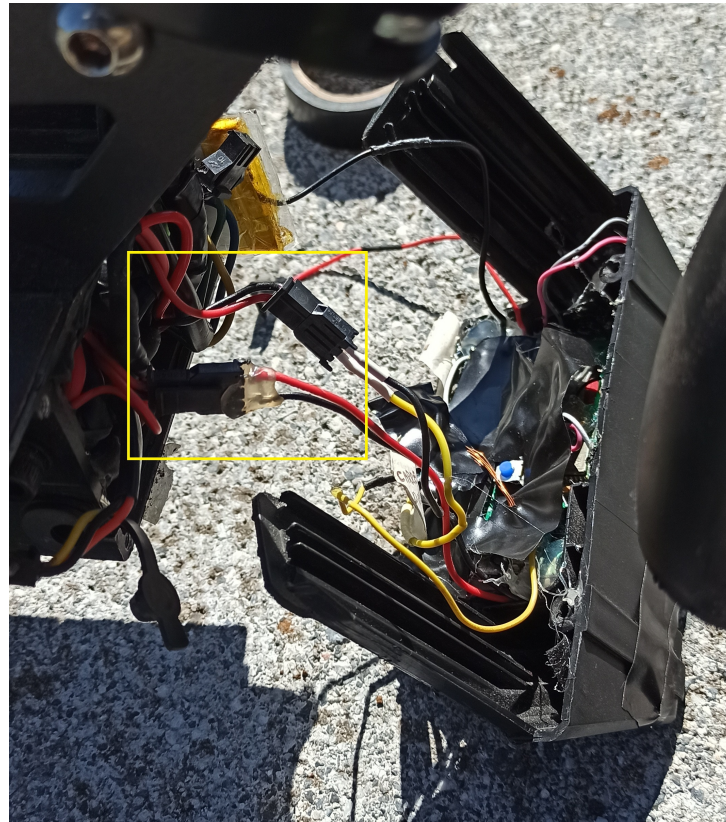
Fonte: Autoria própria.

Fotografia 4 – Conectores *micro fit* para conexão do dispositivo com o patinete.



Fonte: Aatoria própria.

Fotografia 5 – Conexão do dispositivo com o patinete.



Fonte: Aatoria própria.

3.5 GATEWAYS

Um sistema formado por uma placa *Raspberry 3B+* e o *shield LoRa RD43HATGPS* da Radioenge foi montado para ser utilizado como *gateway*. O *software* utilizado foi o fornecido pela própria *Semtech*.

Partindo de uma *Raspberry* com o sistema operacional *Raspbian SO*, pode-se instalar os *softwares* “LoRa gateway” e “LoRa network packet forwarder” disponíveis no *Github*⁵. Algumas configurações adicionais, como, por exemplo, habilitar a SPI e pino de *reset* são necessárias. Outra opção é gravar a imagem disponibilizada pela Radioenge⁶ contendo o *software* do *gateway* e o *Raspbian SO* já integrados e devidamente configurados para executar serviço do *gateway LoRaWAN* utilizando o RD43HATGPS já na inicialização do sistema.

Em ambos os casos, é necessário modificar o arquivo *global_conf.json* para configurar os canais de frequência utilizados pelo *gateway* na comunicação *LoRa*. Por meio deste arquivo *json*, a terceira sub-banda da região AU915 foi configurada, bem como o endereço IP do *Network Server Chirpstack*. O *gatewayID* também deve ser informado, mas por meio do arquivo *local_config.json*.

Para a instalação dos dispositivos, três prédios do Lactec estavam disponíveis, um em cada unidade. Na Figura 29 encontram-se a sede, o Cehpar e o LEME destacados com marcadores circulares laranja, azul e verde, respectivamente.

Figura 29 – Pontos disponíveis para instalação dos gateways LoRaWAN.



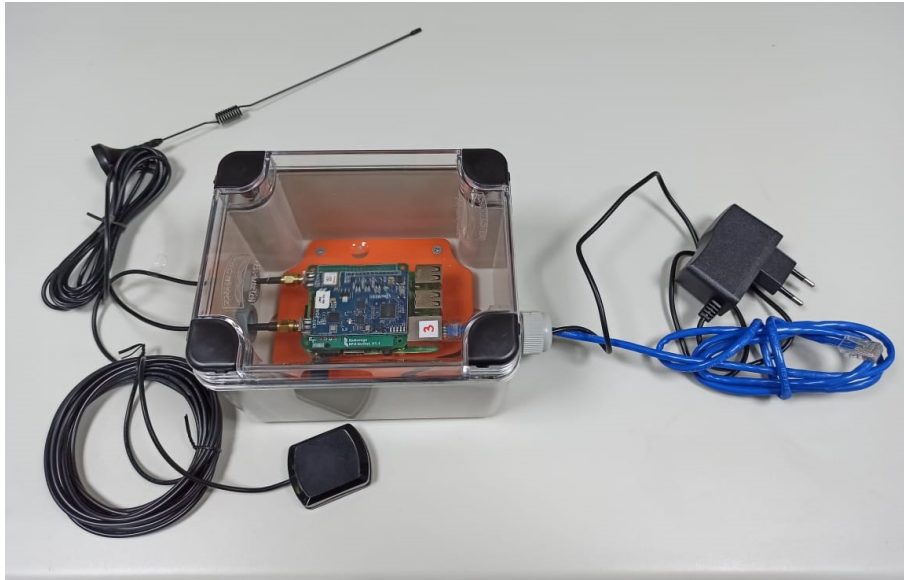
Fonte: *Google Earth*.

⁵ Disponível em: <https://github.com/Lora-net>. Acesso em: 22 de maio de 2020.

⁶ Disponível em: <https://www.radioenge.com.br/solucoes/iot/18-gateway-lorawan.html>. Acesso em: 22 de maio de 2020.

O primeiro *gateway* foi instalado na cobertura do prédio sede. Como os *gateways* foram instalados em ambiente externo, foi necessária a utilização de um gabinete vedado para proteção (Fotografia 6).

Fotografia 6 – Gabinete para o gateway LoRaWAN.



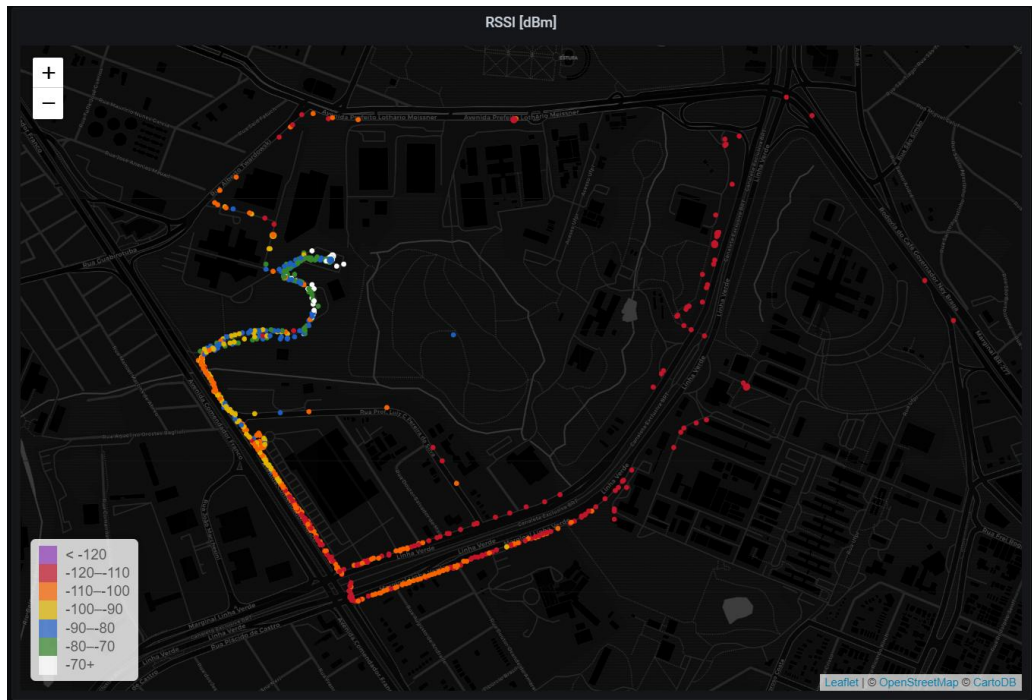
Fonte: Autoria própria.

A região designada para o uso dos veículos (Figura 15) foi percorrida e constatou-se que mais *gateways* seriam necessários para cobrir completamente a região de interesse. O resultado deste teste encontra-se na Figura 30. Isto já era esperado devido à simulação (Figura 31) feita com auxílio da ferramenta *Radio Mobile*⁷, um *freeware* mantido pela VE2DBE para simulações de cobertura de rede e enlaces.

Para a simulação, o *software* requer dados de posicionamento: geolocalização e altura de instalação das antenas, neste caso, correspondente às alturas dos edifícios. Estes dados foram obtidos por meio da ferramenta *Google Earth*. Valores de potência, ganhos de antenas e sensibilidade do rádio também são necessários. Os valores de potência e de ganho foram extraídos dos *datasheets*. Já quanto ao valor de sensibilidade para a modulação *LoRa*, apesar deste estar disponível na documentação, optou-se por utilizar um valor maior (de menor sensibilidade). O valor calculado para BW125kHz e fator de espalhamento SF9, referente à configuração DR3 utilizada para transmissão de dados da aplicação, é de -129dBm. O valor utilizado foi de -120dBm dado que, na prática, um RSSI mais baixo que este valor não foi observado ao longo dos testes. Este fato indica que o ruído ambiente se encontra próximo a este valor.

⁷ Disponível em: <http://www.ve2dbe.com/english1.html>. Acesso em: 31 de maio de 2020.

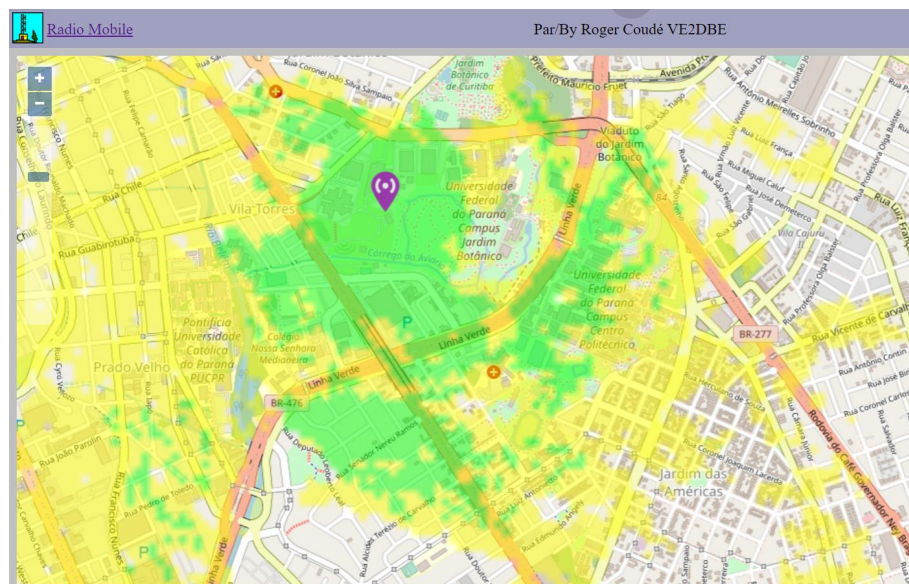
Figura 30 – Dados de RSSI coletados para avaliação da cobertura do gateway LoRaWAN instalado na cobertura do prédio sede do Lactec.



Fonte: Autoria própria. Gerada no software Grafana.

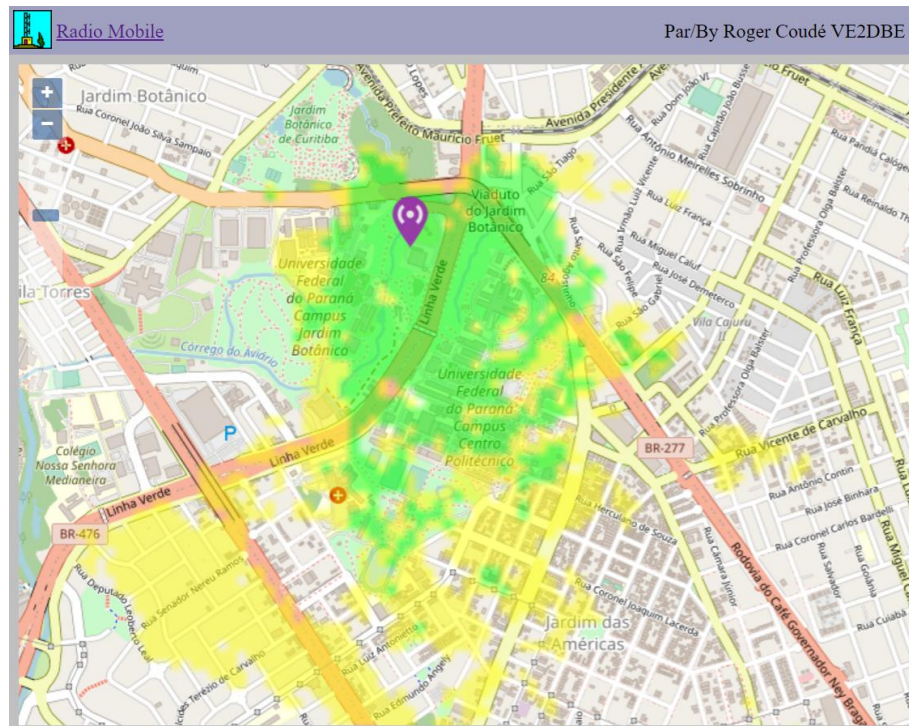
A simulação foi feita para os outros dois posicionamentos, Leme (Figura 32) e Cehpar (Figura 33). Os valores utilizados em todas as simulações encontram-se na íntegra no Anexo A. Apesar das simulações de cobertura para o posicionamento no Leme e no Cehpar não apresentarem tanta diferença, o primeiro foi escolhido para a instalação do segundo gateway. Esta decisão

Figura 31 – Simulação da cobertura do gateway LoRaWAN instalado na cobertura do prédio sede do Lactec.



Fonte: Autoria própria. Gerada no software VE2DBE.

Figura 32 – Simulação da cobertura do gateway LoRaWAN instalado no telhado do prédio Leme do Lactec.



Fonte: Autoria própria. Gerada no software VE2DBE.

foi tomada por dois motivos principais, o primeiro é a diferença de altitude e o segundo é a linha de visada. A antena no Leme ficaria mais elevada, tanto pelo relevo da região (Figura 34), quanto pela altura do prédio (Anexo A). Além disso, o prédio do Cehpar fica atrás de outro bem mais alto, o prédio de ensaios de alta tensão, também do Lactec, mas a instalação neste não era viável.

A combinação da cobertura dos dois gateways, feita no VE2DBE e disponível na Figura 35, indicava que praticamente todo trajeto dos veículos estaria coberto.

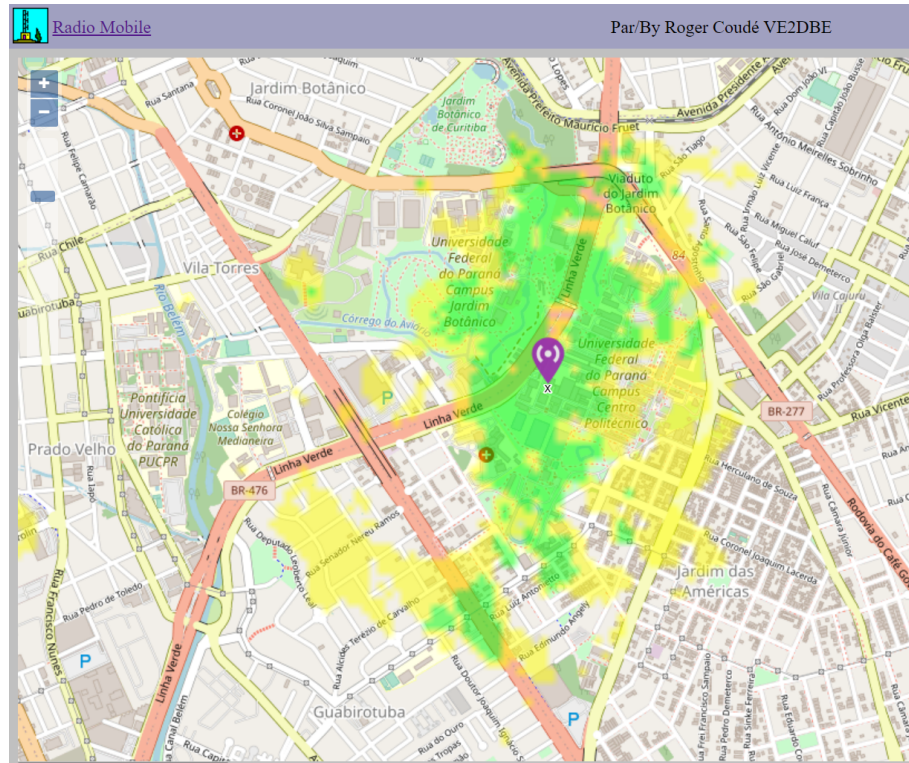
A região de interesse foi novamente percorrida e os dados de RSSI dos dois gateways foram coletados. O dispositivo instalado no Leme apresentou um resultado melhor que o esperado na simulação, como pode ser verificado na Figura 36. Isso pode ser explicado pela simulação considerar dados de relevo, cobertura terrestre e densidade populacional. Esta última pode estar mal dimensionada, dado que apesar de se tratar de uma região urbana, a área analisada apresenta um número reduzido de construções em relação às áreas adjacentes.

Os dados de RSSI da cobertura resultante dos gateways instalados na sede e no Leme podem ser visualizados no mapa da Figura 37.

Algumas ruas nas proximidades foram percorridas de carro para averiguar a cobertura. O desempenho de *uplink* se encontra na Figura 38. Cabe ressaltar que este resultado é melhor

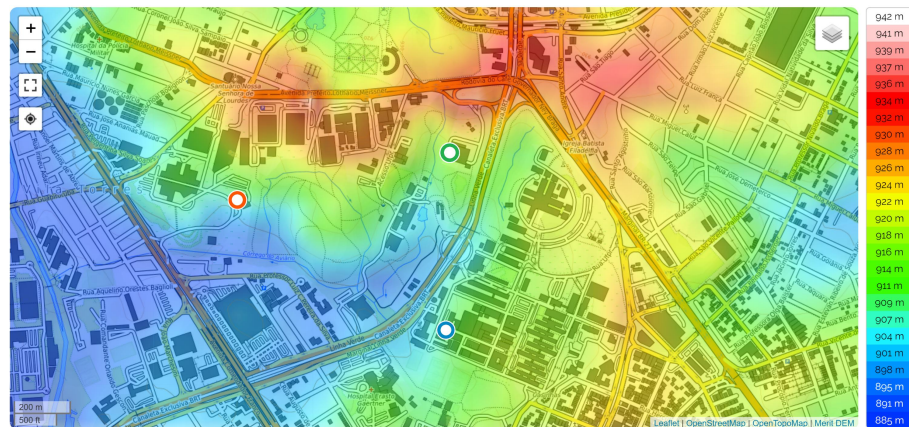
⁸ Disponível em <https://pt-br.topographic-map.com/maps/fwub/Curitiba/>. Acesso em 17 de julho de 2021.

Figura 33 – Simulação da cobertura do gateway LoRaWAN instalado no telhado do prédio Cehpar do Lactec.



Fonte: Autoria própria. Gerada no software VE2DBE.

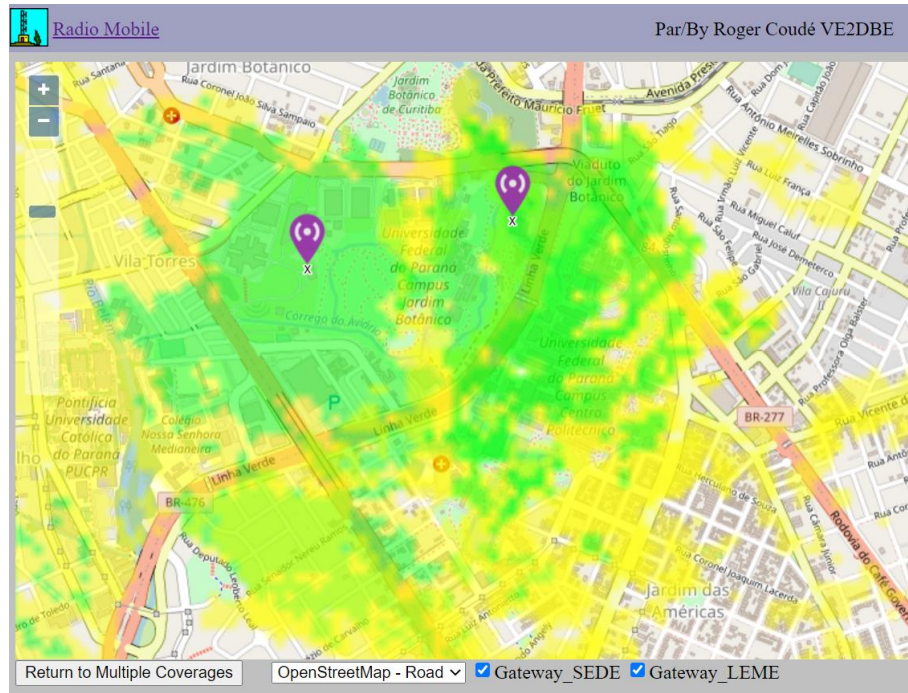
Figura 34 – Mapa de relevo da região de operação do sistema com destaque da localização para possível instalação dos gateways LoRaWAN.



Fonte: Adaptado de Topographic-map⁸.

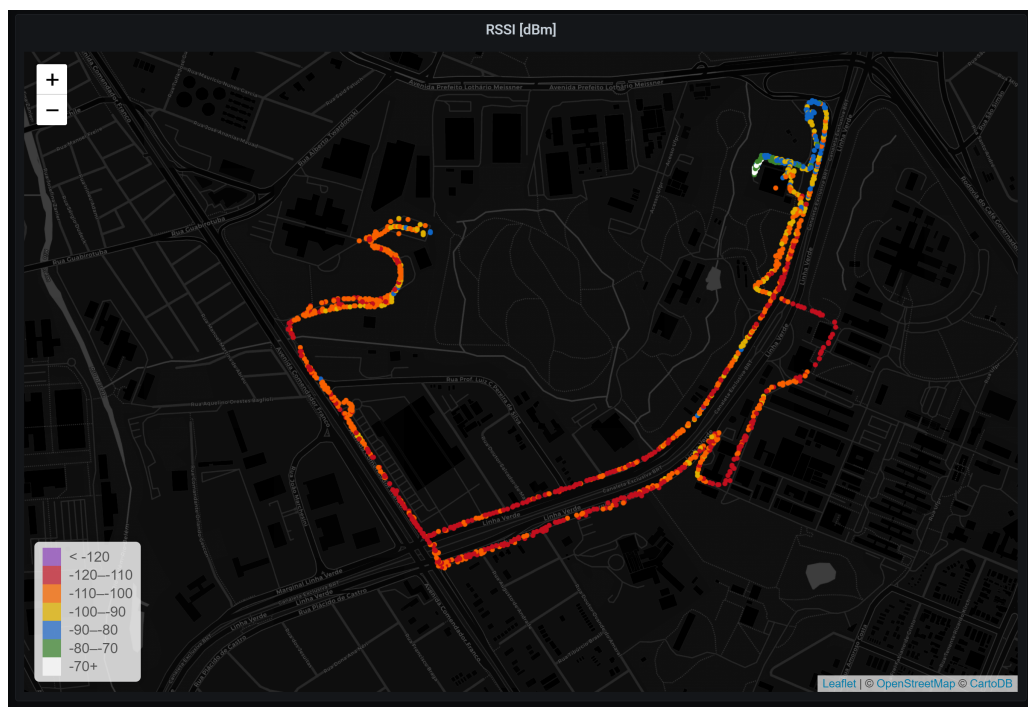
que o esperado com o sistema acoplado ao patinete, por conta do aumento da altura da antena.

Figura 35 – Simulação da cobertura dos gateways LoRaWAN instalado no prédios sede e Cehpar.



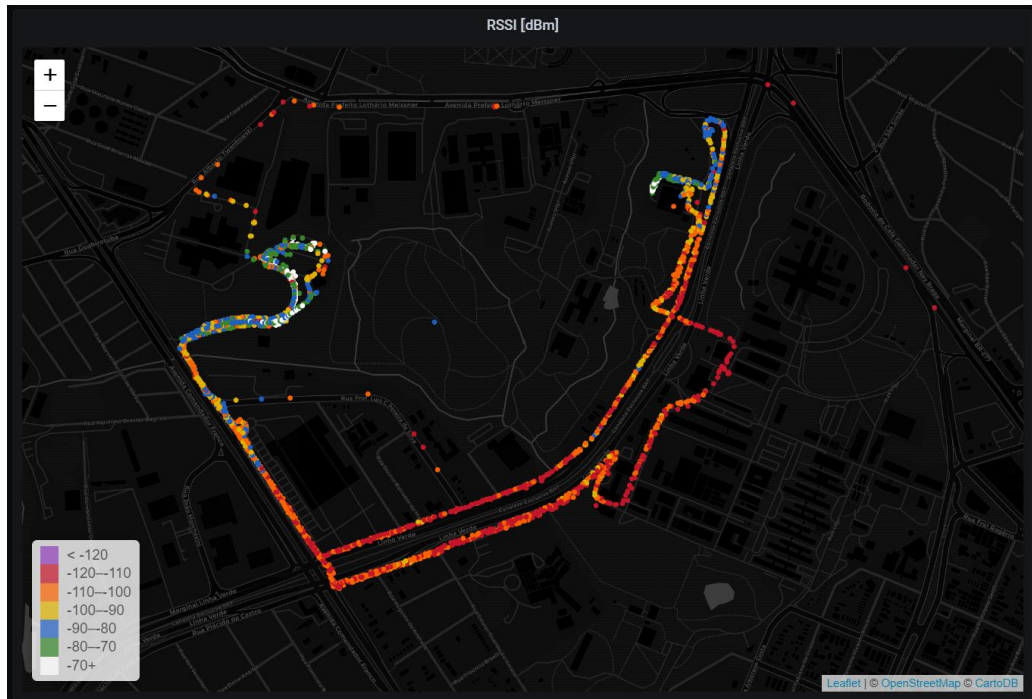
Fonte: Autoria própria. Gerada no software VE2DBE.

Figura 36 – Dados de RSSI coletados para avaliação da cobertura do gateway LoRaWAN instalado no telhado do prédio Leme do Lactec.



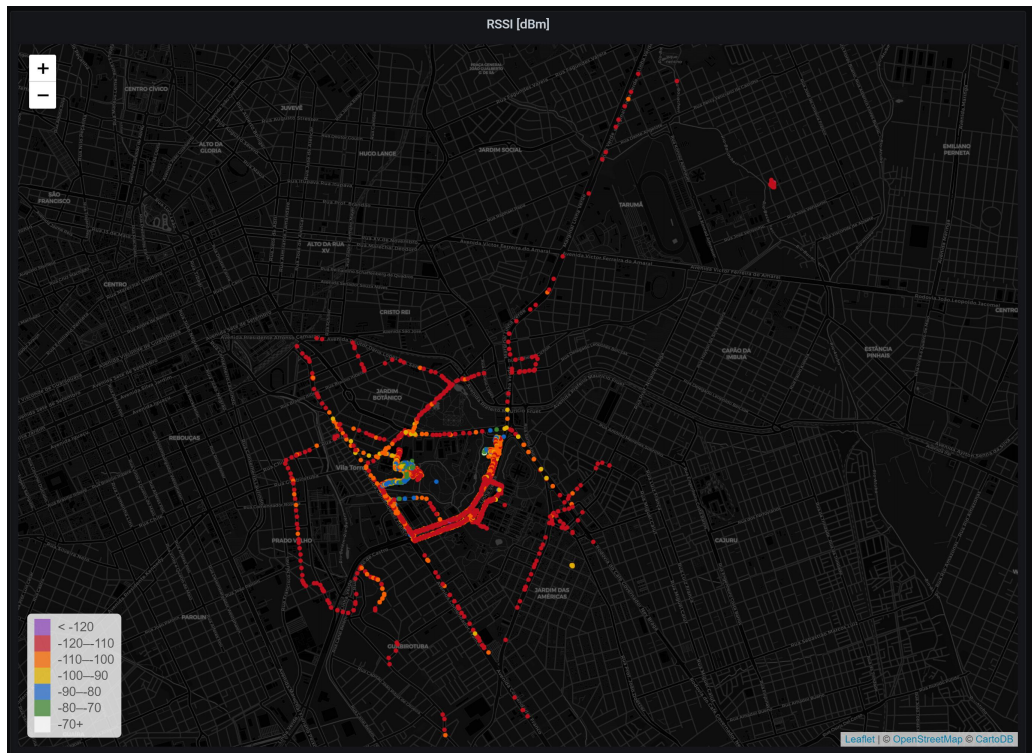
Fonte: Autoria própria. Gerada no software Grafana.

Figura 37 – Dados de RSSI coletados para avaliação da cobertura dos gateways LoRaWAN instalados nos prédios sede e Leme do Lactec – patinete.



Fonte: Autoria própria. Gerada no software Grafana.

Figura 38 – Dados de RSSI coletados para avaliação da cobertura dos gateways LoRaWAN instalados nos prédios sede e Leme do Lactec – carro.

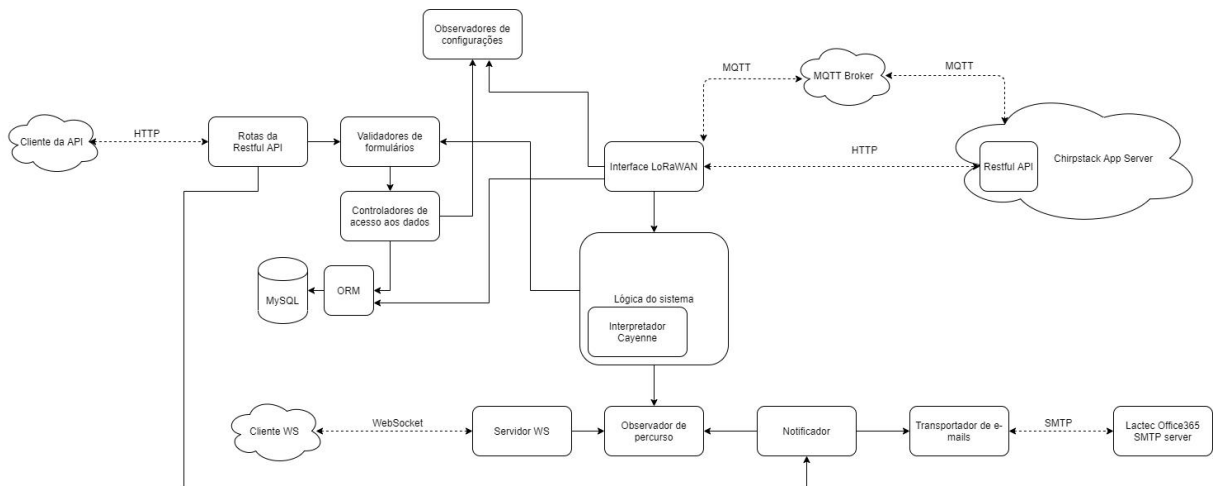


Fonte: Autoria própria. Gerada no software Grafana.

3.6 SERVIDOR DE APLICAÇÃO PIOTNET

Para implementação do bloco “Servidor de Aplicação PloTnet” apresentado na Figura 16, um projeto de *software* foi desenvolvido. O diagrama de blocos da Figura 39 representa a arquitetura do *software* projetado para satisfazer os requisitos de sistema (Apêndice A). Na figura, as setas indicam que o bloco a apontar conhece o bloco apontado, podendo utilizar seus métodos e atributos públicos. Em seguida, uma breve explicação da função de cada bloco é fornecida. A construção desse *software* foi feita em *Javascript* para execução no ambiente *Node.JS*. A ferramenta de versionamento *Git* foi utilizada para controle das alterações feitas no código fonte e um repositório remoto está hospedado no *GitHub* na url <<https://github.com/gabriel-mendes-dev/PIoTNetAppServer>>. É necessária autorização dos autores para acesso ao repositório.

Figura 39 – Diagrama de blocos do servidor de aplicação.



Fonte: Autoria própria.

- **Rotas da Restful API:** conjunto de instâncias que se comunicam com o *software* cliente a partir da disponibilização de uma API Restful, permitindo que ele tenha acesso para consulta aos dados do sistema e configuração do mesmo, além de alguns comandos direcionados aos dispositivos IoT, conforme documentação apresentada no Apêndice E. A implementação dos tratamentos dos métodos do protocolo HTTP para criação da API foi feita por meio do *framework Express*⁹ para *Node.JS*. Uma instância existe para cada rota da API e cada uma delas é implementada por um arquivo da pasta “src/api/routes” no repositório do código fonte.

⁹ Disponível em: <https://github.com/expressjs/express>. Acesso em 18 de julho 2021

- **Validadores de formulários:** conjunto de instâncias que comparam as informações a serem salvas no banco de dados com os formatos esperados para evitar corrupções de dado e falhas na execução do *software*. Todos os validadores são implementados no mesmo arquivo “src/services/forms.js” e são baseadas no módulo *ajv*¹⁰ que aponta ser o mais rápido validador de esquemas em formato *JavaScript Object Notation* (JSON).
- **Controladores de acesso aos dados:** conjunto de instâncias que controlam a interface do *software* com o banco de dados, encapsulando os comandos de criação, edição e remoção de dados. Novamente, as implementações são separadas por arquivos, conforme as tabelas do banco de dados que são administradas pela instância, e estão na pasta “src/services”. O controlador dos dados de percurso, por exemplo, que tem funções para lidar com as tabelas “Route” e “RouteData” do banco de dados, é construída no arquivo “src/services/routeService.js”. Os controladores de acesso aos dados podem conhecer outros controladores para que respeitem o encapsulamento ao lidar com outros dados do banco. Por exemplo, o controlador dos dados de veículo “src/services/vehicleService.js” utiliza o módulo de acesso aos dados de percurso anteriormente citado.
- **ORM (*Object Relational Mapping*):** converte os dados entre objetos da linguagem utilizada e a linguagem de banco de dados de forma encapsulada. Ou seja, no caso deste projeto, em que foi escolhido o banco de dados MySQL, permite que as consultas no banco de dados sejam executadas por simples funções em linguagem *Javascript* e que as tabelas sejam modeladas por objetos em arquivos do tipo “.js”. Neste trabalho, foi utilizada a biblioteca *Sequelize*¹¹ para este fim. Todos os modelos utilizados estão na pasta “src/models”.
- **Observadores de configurações:** instâncias que observam eventos emitidos pelos controladores de dados do banco para comunicação entre diferentes módulos de *software*. Os observadores, que são objetos da classe *eventEmitter*, são instanciados dentro dos próprios controladores de acesso aos dados do banco. Sendo objetos públicos, por meio deles os eventos emitidos pelos controladores podem ser usados para disparar funções de *callback* por outras classes. Como exemplo, é por meio de um observador que o módulo “Lógica do Sistema” sabe que um veículo foi cadastrado pelo administrador da rede e cadastra o

¹⁰ Disponível em: <https://github.com/ajv-validator/ajv>. Acesso em 18 de julho 2021

¹¹ Disponível em: <https://github.com/sequelize/sequelize>. Acesso em 18 de julho 2021

mesmo na infraestrutura *Chirpstack*, evitando que o veículo tenha que ser cadastrado em dois locais.

- **Interface LoRaWAN:** parte do *software* responsável por se comunicar com o *Chirpstack* Application server, tanto para configurações, via HTTP, quanto para comunicação de *uplink* e *downlink* com os dispositivos via MQTT. Para comunicação com a API da *Chirpstack*, a biblioteca *axios*¹² foi utilizada para implementar um cliente HTTP. Já para comunicação com o *broker* MQTT Mosquitto, a biblioteca *mqtt*¹³ foi utilizada.
- **Lógica do sistema:** bloco que recebe as informações de *uplink* dos *end devices* e executa as principais ações intrínsecas ao sistema desenvolvido, conforme os fluxogramas apresentados no Apêndice C e no Apêndice D. Cabe a este bloco indicar quando um processo de recarga ou percurso foi iniciado ou finalizado. Ele também identifica quando um veículo está fora da área permitida, emitindo um evento por meio do *Observador de Percurso* que aciona uma notificação por e-mail ao responsável pelo sistema e um *downlink* para ativação do *buzzer* no *end node*. Um evento também é emitido quando uma informação de possível acidente é recebida. Além disso, neste bloco são calculados os tempos de percurso total e com o veículo parado, além da quilometragem, que são ao final do percurso integrados às informações do banco de dados. Todos os dados de percurso recebidos são também emitidos por meio do *Observador de Percurso*, para que um outro objeto possa utilizá-los para apresentação em tempo real – neste caso, o *Servidor WS*. Os cálculos de distância e verificação de pertencimento à área permitida a partir de pontos de latitude e longitude são realizados a partir da biblioteca *turf*¹⁴.
- **Observador de percurso:** instância da classe *eventEmitter* que observa os eventos emitidos pelo bloco "Lógica do sistema" em relação aos veículos em percurso para disparar outras funcionalidades do *software*, como notificações por e-mail e comunicação em tempo real.
- **Servidor WS:** servidor *WebSocket* para envio dos dados de percurso em tempo real, implementado por meio do módulo *ws*¹⁵.

¹² Disponível em: <https://github.com/axios/axios>. Acesso em 18 de julho 2021

¹³ Disponível em: <https://github.com/mqttjs/MQTT.js>. Acesso em 18 de julho 2021

¹⁴ Disponível em: <https://github.com/Turfjs/turf>. Acesso em 18 de julho 2021

¹⁵ Disponível em: <https://github.com/websockets/ws>. Acesso em 18 de julho 2021

- **Notificador:** instância que controla as notificações enviadas ao responsável pelo sistema quando um evento indicando possível acidente ou que um veículo está fora da área permitida é acionado. Este bloco torna públicos os métodos que interrompem o envio periódico de notificações, permitindo a uma outra classe reconhecer que o supervisor visualizou a notificação e encerrar o envio – o que é feito por uma das rotas da API, cuja URL é disponibilizada no próprio e-mail da notificação.
- **Transportador de e-mails:** conecta-se ao servidor SMTP e envia os e-mails conforme requisitado pelo bloco anteriormente descrito. Neste trabalho, foi utilizada a biblioteca *Nodemailer*¹⁶ para este fim.

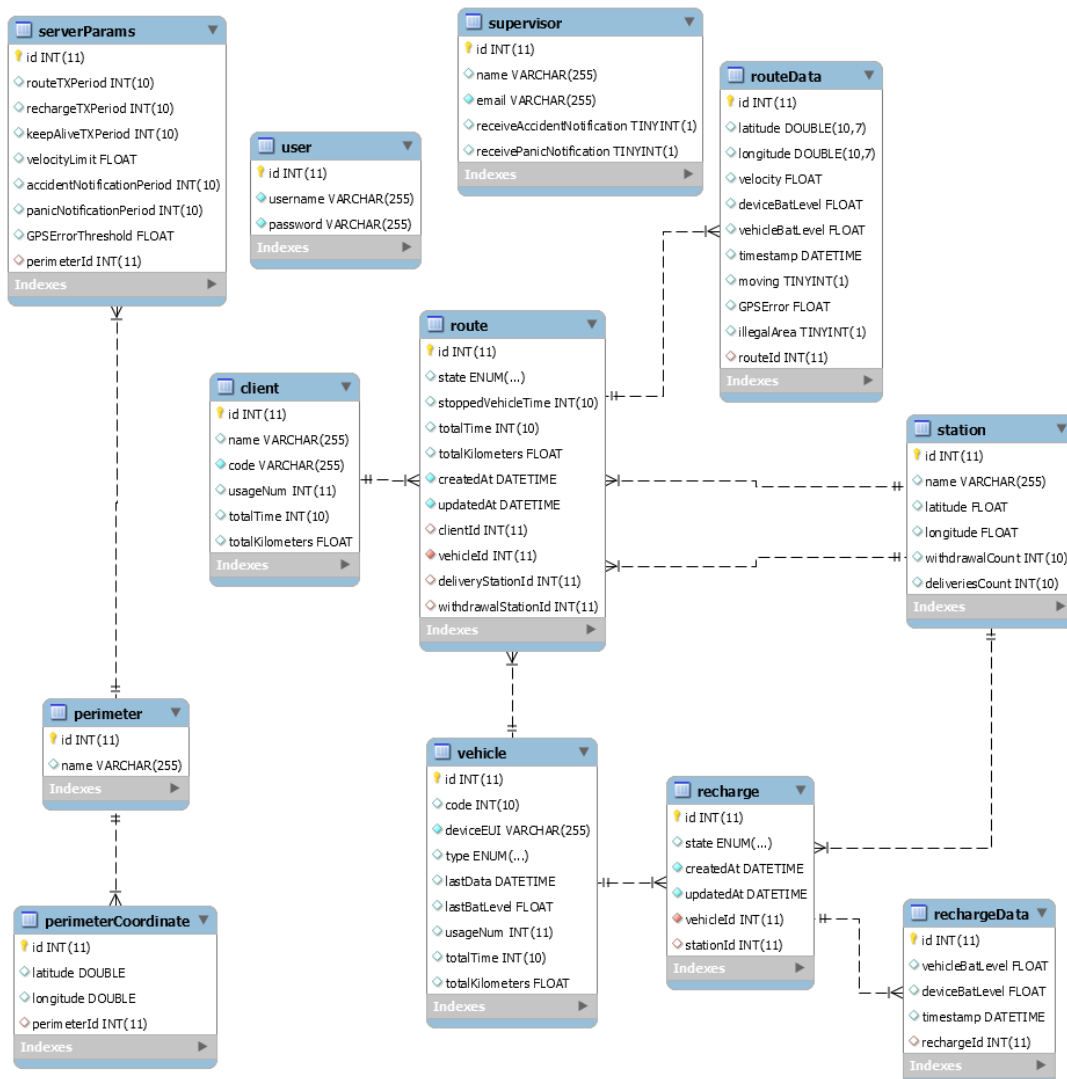
A Figura 40 apresenta o diagrama do banco de dados desenvolvido para o servidor de aplicação, cujos significados das tabelas são explicados a seguir:

- *serverParams*: as colunas desta tabela são os parâmetros de configuração da aplicação, como períodos de envio de mensagens pelos nós nos diferentes modos, incerteza considerada adequada para os dados de GPS, área de circulação considerada legal entre as cadastradas na tabela *perimeter*, etc. Portanto, só existe uma instância deste tipo.
- *perimeterCoordinate*: cada linha dessa tabela é um ponto de localização geográfica, a partir da latitude e longitude, pertencente a um ítem da tabela *perimeter*.
- *perimeter*: cada linha dessa tabela identifica um conjunto de pontos geográficos que define um perímetro, cuja área interna é utilizada como área legal de circulação.
- *user*: cada linha é um usuário para autenticação na API Restful.
- *supervisor*: cada linha é um responsável que pode receber notificações de possível acidente e/ou veículo fora da área legal de circulação.
- *route*: cada linha dessa tabela representa um percurso realizado.
- *routeData*: cada linha dessa tabela é um dado de percurso pertencente a um ítem da tabela *route*.
- *client*: cada linha representa um funcionário do Lactec cadastrado no sistema de veículos elétricos.

¹⁶ Disponível em: <https://github.com/nodemailer/nodemailer>. Acesso em 18 de julho 2021

- *station*: cada linha representa uma estação onde os patinetes podem ser acoplados para carregamento ou retirados para entrar em percurso.
- *vehicle*: cada linha representa um veículo do sistema.
- *recharge*: cada linha representa uma recarga de bateria pela qual um veículo passou.
- *rechargeData*: cada linha é um dado de recarga pertencente a um item da tabela *recharge*.

Figura 40 – Diagrama de banco de dados do servidor de aplicação.



Fonte: Autoria própria. Gerada no software *MySQL Workbench*.

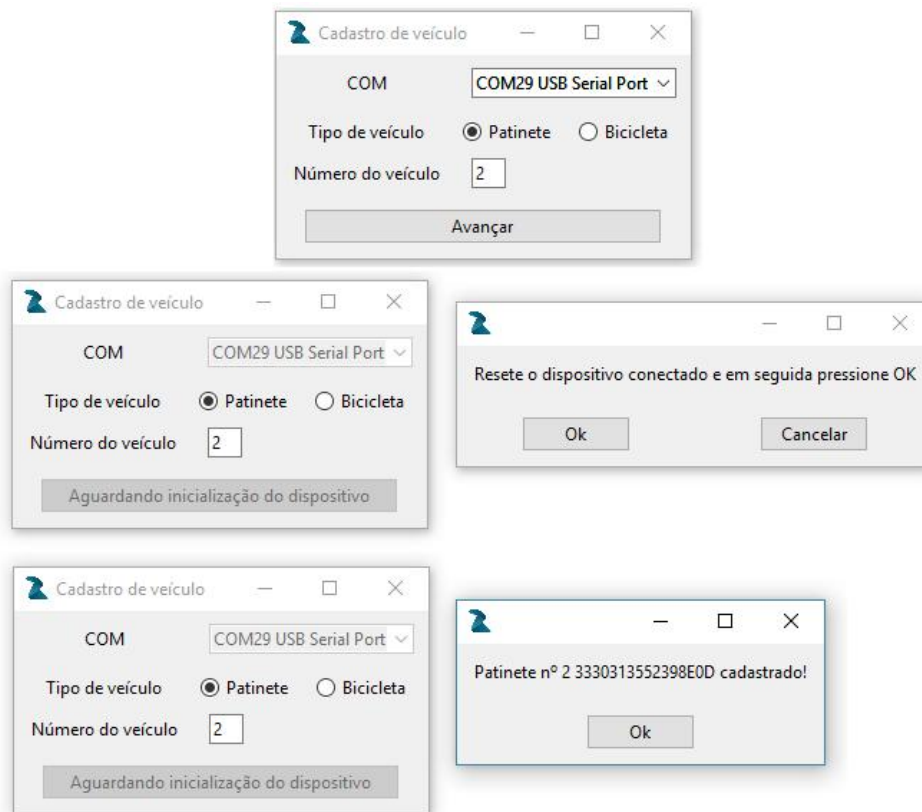
Cadastro de dispositivos

Para auxiliar no cadastramento de um novo dispositivo no sistema, foi desenvolvida uma aplicação gráfica. O aplicativo desenvolvido em *Python* permite a conexão via USB com o

dispositivo de monitoramento e por meio desta, o *deviceEUI* e o *AppKey* são lidos.

Na tela de configuração (Figura 41), pode ser indicado se o veículo é um patinete ou uma bicicleta e seu número de identificação na estação. O aplicativo, na realidade, automatiza o uso da API desenvolvida para facilitar o cadastro do dispositivo. Este cadastro é feito tanto no servidor *Chirpstack*, quanto na aplicação de monitoramento desenvolvida pelos autores.

Figura 41 – Sequência de telas do processo de cadastro de dispositivos com auxílio do aplicativo de cadastro.



Fonte: Autoria própria.

4 TESTES E RESULTADOS DO SISTEMA DE MONITORAMENTO DOS VEÍCULOS

Para avaliar os resultados obtidos com o desenvolvimento deste trabalho, foi verificado se os requisitos apresentados no Apêndice A são satisfeitos por meio de testes do sistema. Nesta seção, serão apresentados os principais resultados e a qualidade com a qual os requisitos foram satisfeitos, indicando possíveis melhorias por meio de alterações no sistema e no dispositivo IoT.

4.1 MONITORAMENTO DE VEÍCULOS EM PERCURSO

Inicialmente, para avaliar controladamente o monitoramento do veículo em rota, foram realizados percursos no estacionamento do Lactec Sede, próximo à estação de carregamento dos veículos. Para isso, a área legal de circulação foi configurada, por meio da API desenvolvida (Apêndice E), como a área em roxo mostrada na Figura 42. O polígono da área legal é definido a partir dos quatro vértices localizados pelos marcadores amarelos na imagem. Com base nos pontos apresentados na figura, o procedimento apresentado a seguir foi executado com o veículo. Os pontos marcados no mapa eram somente um guia aproximado para o condutor e não o lugar exato onde ele deveria parar ou passar. O controle da velocidade foi feito pelo condutor a partir da observação da mesma no visor do veículo.

1. Retirar o veículo da estação de carregamento.
2. Partir da estação ao ponto B, passando próximo ao ponto A, com uma velocidade de aproximadamente 10km/h.
3. Esperar 30 segundos.
4. Partir do ponto B ao ponto C com uma velocidade próxima a 5km/h.
5. Esperar 20 segundos.
6. Partir do ponto C ao ponto D com uma velocidade próxima a 15km/h.
7. Esperar 10 segundos.
8. Partir do ponto D para a estação de carregamento.
9. Acoplar o veículo à estação de carregamento.

Figura 42 – Área legal de circulação dos veículos para o teste no estacionamento do Lactec.

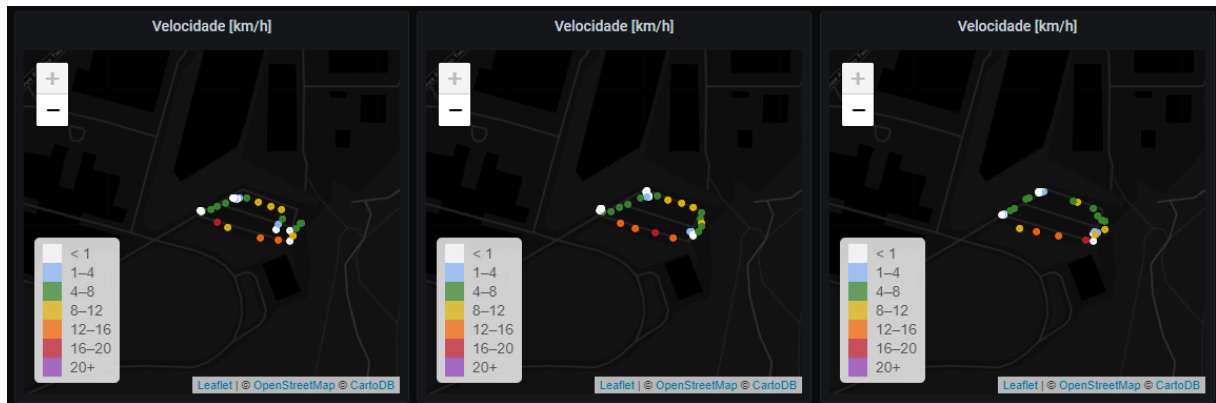


Fonte: Autoria própria. Gerada no software Google Earth.

A Figura 43 apresenta os resultados das medidas de velocidade e posição no mapa para diferentes execuções do procedimento. A distância total percorrida, medida por meio do *Google Earth*, deve ser próxima de 280 metros. A Figura 44 apresenta os resultados computados para o percurso lidos com o *Grafana* por meio da API desenvolvida. O identificador “1” nas colunas “Estação de retirada” e “Estação de devolução” identificam a estação de carregamento da sede do Lactec no banco de dados, enquanto “2” e “3” identificam Leme e Cehpar, respectivamente. A cada parada, o alinhamento dos pacotes de *uplink* no tempo pode adicionar um erro de até 10 segundos devido ao período de envio, o que é significativo para esse teste (percurso pequeno com várias paradas) mas não para os percursos reais (grandes e com poucas paradas).

A Figura 45 apresenta um gráfico com o reconhecimento de movimentação do veículo por meio da vibração e a velocidade definida por meio do GPS. Os momentos em que a velocidade é nula e o reconhecimento por vibração acusa movimentação ocorrem devido a falha na recepção de GPS porque, neste caso, a velocidade é armazenada como 0 – assim, é uma vantagem ter o reconhecimento de movimentação por acelerômetro. As medidas de velocidade, distância percorrida e o reconhecimento da estação de retirada e de acoplamento apresentaram resultados satisfatórios, assim como o reconhecimento de movimentação e contagem de tempo parado, considerando-se o trajeto executado.

Figura 43 – Resultados do teste de rastreamento no estacionamento – mapa.



Fonte: Autoria própria. Gerada no *software* Grafana.

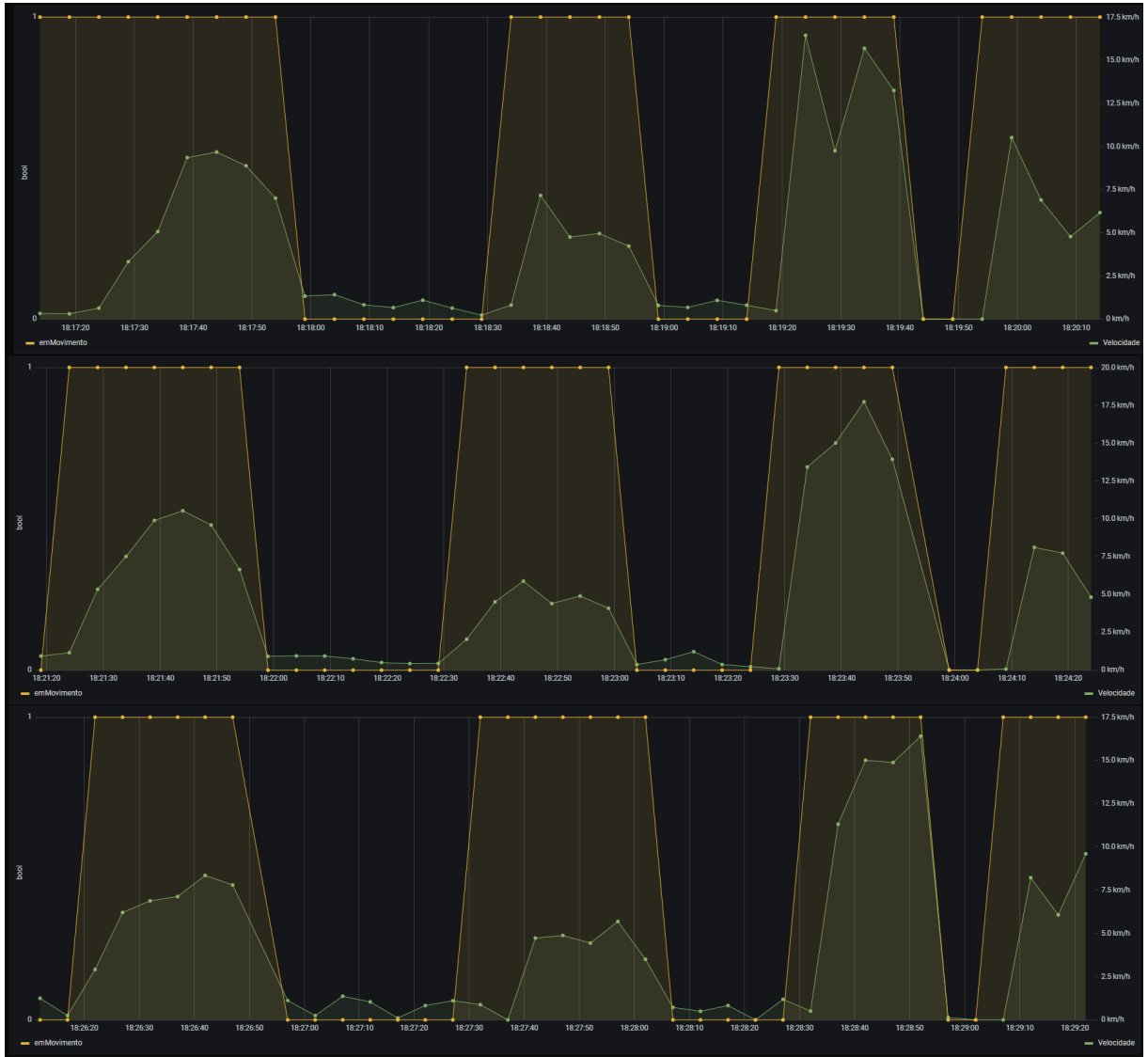
Figura 44 – Resultados do teste de rastreamento no estacionamento – tabela.

Rotas									
id	Cliente ID	Status	Tempo parado [s]	Tempo total [s]	Total [km]	Criado em	Atualizado em	Estação de retirada	Estação de devolução
90	1	Finished	65	185	0.260	2021-07-22 18:17:16	2021-07-23 22:52:14	1	1
91	1	Finished	75	189	0.279	2021-07-22 18:21:21	2021-07-23 22:52:14	1	1
92	1	Finished	80	195	0.276	2021-07-22 18:26:14	2021-07-23 22:52:14	1	1

Fonte: Autoria própria. Gerada no *software* Grafana.

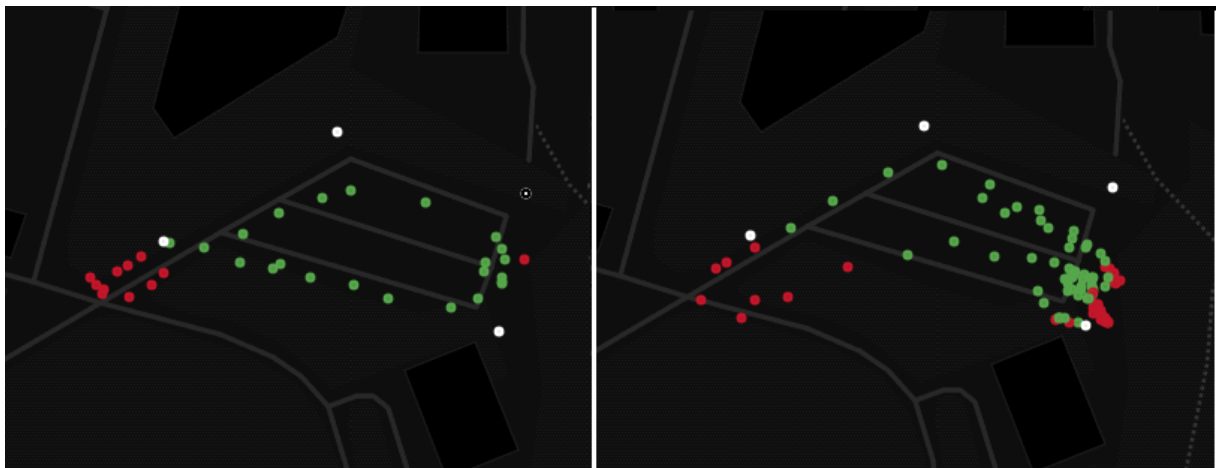
O reconhecimento de veículo fora da área legal de circulação foi testado por meio do transpasse com o veículo do perímetro definido e verificação dos dados salvos por meio da API. A Figura 46 apresenta resultados de dois testes distintos com este propósito. Os pontos brancos são os vértices da área legal de circulação, enquanto os pontos verdes e vermelhos são pontos em que o sistema reconheceu o veículo dentro e fora da área, respectivamente. Enquanto o veículo estava fora de área, o aviso sonoro de indicação foi emitido e, quando o veículo retornou, o aviso sonoro parou, como foi definido. A funcionalidade, portanto, apresentou o resultado esperado. O mesmo teste foi realizado, ainda dentro do Campus da FIEP, após a configuração da área de circulação dos patinetes do Lactec (Figura 47) e alguns resultados são apresentados na Figura 48. O reconhecimento de veículo fora de área ocorreu normalmente, no entanto não houve acionamento do aviso sonoro. Os registros do *Chirpstack Application Server* foram verificados e o *downlink* de “Área ilegal” foi enviado corretamente e repetido periodicamente uma vez que a confirmação não era recebida. Neste caso, percebeu-se que na área do teste, embora o enlace permita o envio dos *uplinks* sem nenhum problema, não é possível enviar *downlinks*. Isso ocorre, possivelmente, devido às diferentes largura de banda e taxa de transmissão utilizadas para *uplink* e *downlink*, que levam a diferentes sensibilidades no receptor *LoRa*.

Figura 45 – Resultados do teste de rastreamento no estacionamento – velocidade e detecção de movimento por acelerômetro.



Fonte: Autoria própria. Gerada no software Grafana.

Figura 46 – Resultados de verificação de veículo fora de área – teste no estacionamento.



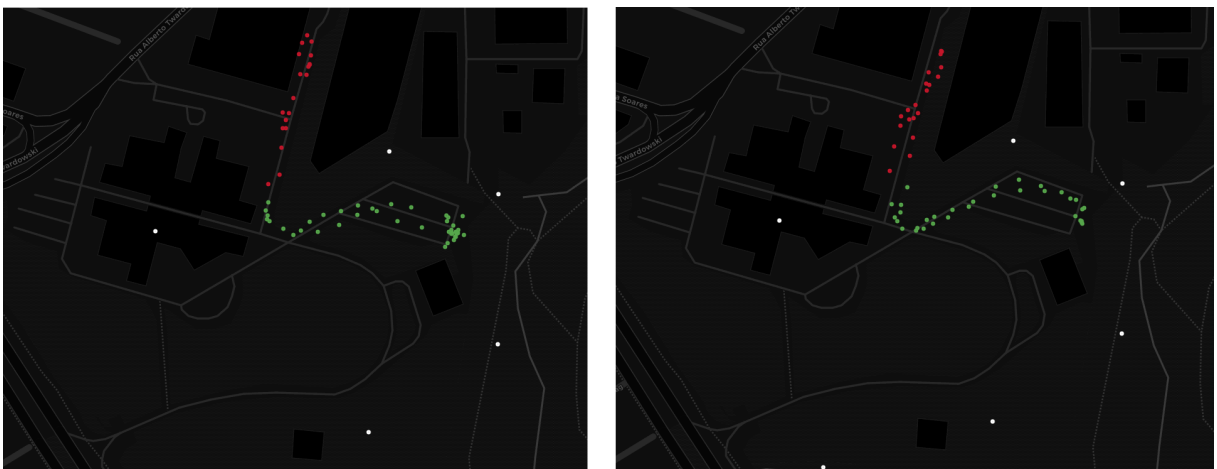
Fonte: Autoria própria. Gerada no software Grafana.

Figura 47 – Área legal de circulação dos veículos do Lactec.



Fonte: Autoria própria. Gerada no software Google Earth.

Figura 48 – Resultados de verificação de veículo fora de área – teste no estacionamento com área oficial.



Fonte: Autoria própria. Gerada no software Grafana.

Após os testes no estacionamento do Lactec, um teste foi feito com um percurso real, com o procedimento descrito a seguir. Não foi possível realizar o acoplamento do veículo na estação do Lactec Cehpar pois a mesma se encontrava em manutenção na data de execução dos testes. No entanto, o percurso seguido inclui este prédio para garantir o funcionamento do sistema na região.

1. Retirar o veículo da estação de carregamento no Lactec Sede.
2. Realizar a rota amarela da Figura 15 até a bifurcação com a rota vermelha e, então, realizar a rota vermelha até o Lactec Leme.
3. Acoplar o patinete à estação de carregamento do Lactec Leme.
4. Realizar a rota azul até o Lactec Cehpar. No caminho, dentro do Campus Politécnico da UFPR, onde o prédio do Cehpar é localizado, sair brevemente da área legal de circulação e então retornar. Verificar se o supervisor cadastrado no sistema recebe as notificações de saída e retorno à área.
5. Realizar a rota amarela até o Lactec Sede.
6. Acoplar o patinete à estação de carregamento do Lactec Sede.

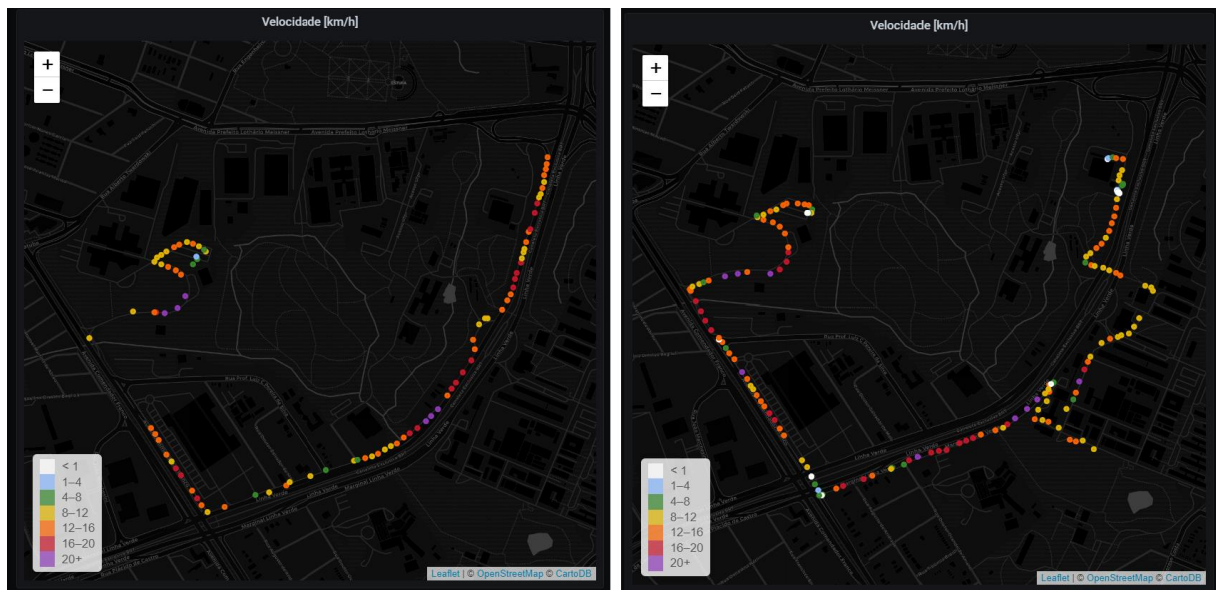
Com o procedimento descrito, dois históricos de percurso e dois históricos de recarga foram criados no sistema. A Figura 49 apresenta os dados de GPS para o percurso Sede-Leme e para o percurso Leme-Sede – as diferentes cores representam diferentes intervalos de velocidade, em km/h. Pode-se notar um intervalo de ausência de dados no início do percurso de ida, sendo que no mesmo trecho, na volta, os dados são apresentados normalmente. Ao final do percurso de ida, também houve um momento de falha ao adentrar o Lactec Leme. Esses pontos não estão ausentes devido a falhas na comunicação *LoRa*, mas sim devido a falhas ao tentar receber dados de GPS.

A Figura 53 apresenta o RSSI dos pacotes *LoRa* recebidos pelo *gateway* juntamente com a DOP do GPS – quando esta é apresentada como zero, é porque não houve recepção. A Figura 52 apresenta a detecção de veículo em movimento por meio do acelerômetro juntamente com a velocidade adquirida por GPS para os dois percursos – novamente, é importante notar que a velocidade é armazenada como zero quando ocorre erro na recepção de GPS, o que não prejudica a contagem de tempo parado já que esta é feita pela consideração do estado detectado

por acelerômetro. A Figura 54 apresenta os valores computados de tempo de percurso e distância percorrida, requisitados por meio da API, além das estações de partida e de chegada. A distância percorrida esperada, a partir do *Google Earth*, era de aproximadamente 2,7 km para o percurso Sede-Leme e 3,2 km para o percurso Leme-Sede. O tempo com o veículo parado durante o percurso foi cronometrado com um relógio e foi de 18 segundos e 121 segundos, respectivamente. Todos os resultados apresentados na tabela estão, portanto, adequados ao que se esperava.

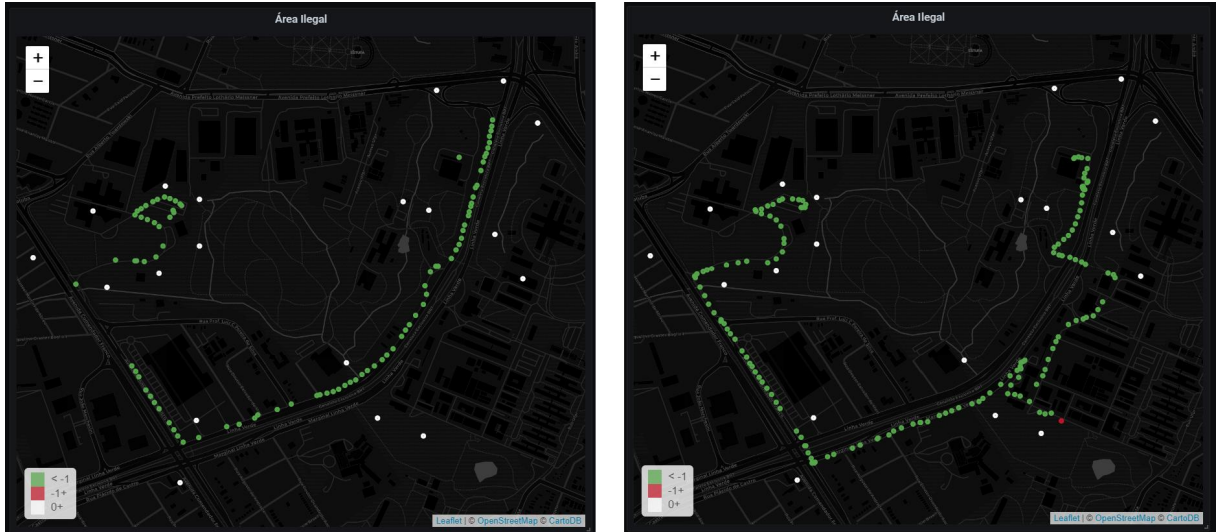
A Figura 50 apresenta a detecção de veículo fora da área legal de circulação pelo sistema. As notificações enviadas por e-mail ao supervisor são mostradas na Figura 51. Durante o percurso, o condutor foi até um ponto mais distante do que o único ponto vermelho indicado no mapa, o que indica que a comunicação também falhou nessa região. Nesse caso, houve falha na comunicação *LoRa*, pelo que foi verificado nos pacotes recebidos. Apesar da detecção, os *downlinks* também não foram recebidos devido a falha na comunicação, como no teste anterior.

Figura 49 – Resultados de rastreamento do veículo em percurso da Sede ao Leme, à esquerda, e do Leme à Sede, à direita.



Fonte: Autoria própria. Gerada no software Grafana.

Figura 50 – Resultados de identificação de veículo fora de área no percurso da Sede ao Leme, à esquerda, e do Leme à Sede, à direita.



Fonte: Autoria própria. Gerada no *software* Grafana.

Figura 51 – Notificações recebidas ao sair e retornar à área durante percurso com veículo nas rotas do Lactec.



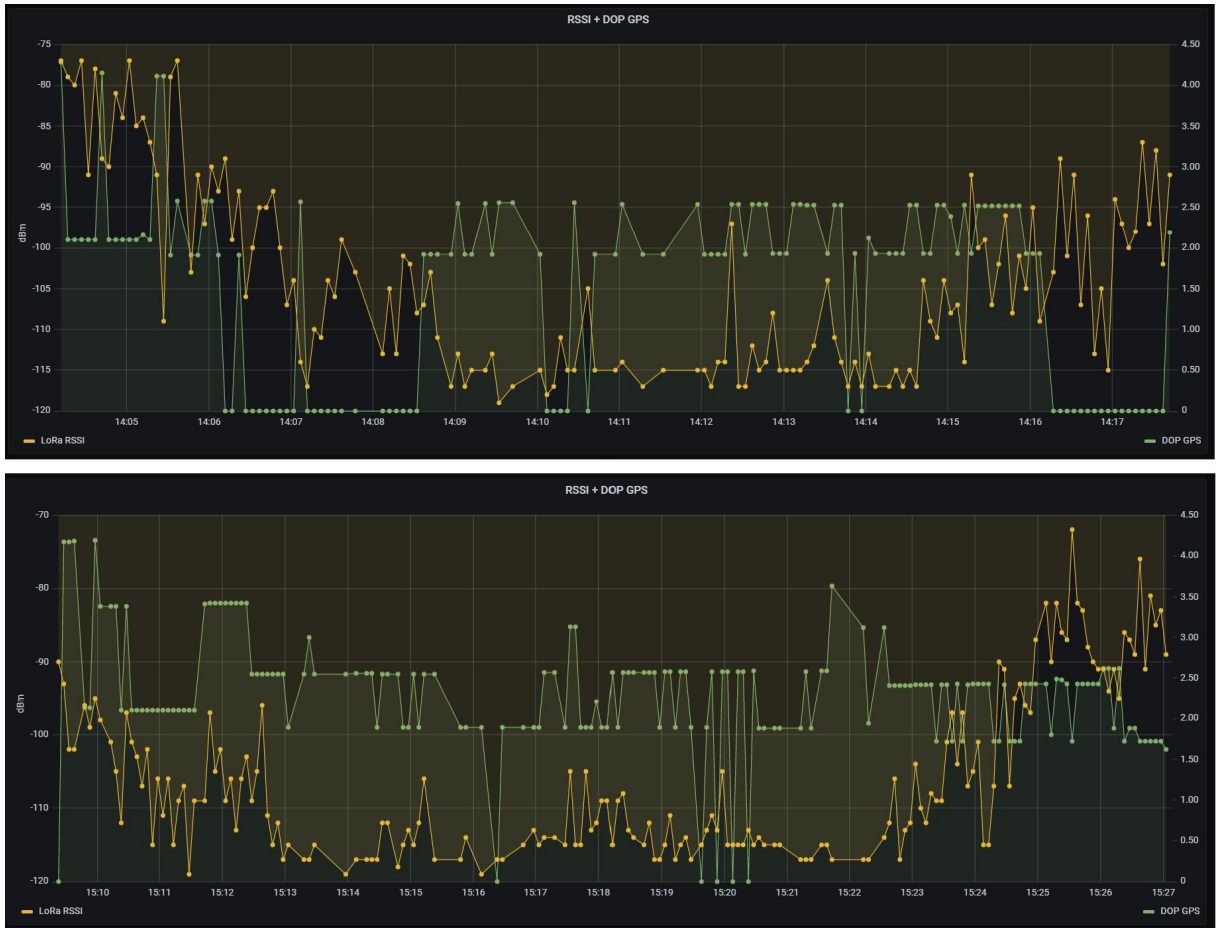
Fonte: Autoria própria. Capturas de tela do *gmail*.

Figura 52 – Resultados de detecção de movimento do veículo no percurso da Sede ao Leme, acima, e do Leme à sede, abaixo.



Fonte: Autoria própria. Gerada no software Grafana.

Figura 53 – Resultados de RSSI e DOP do veículo no percurso da Sede ao Leme, acima, e do Leme à sede, abaixo.



Fonte: Autoria própria. Gerada no software Grafana.

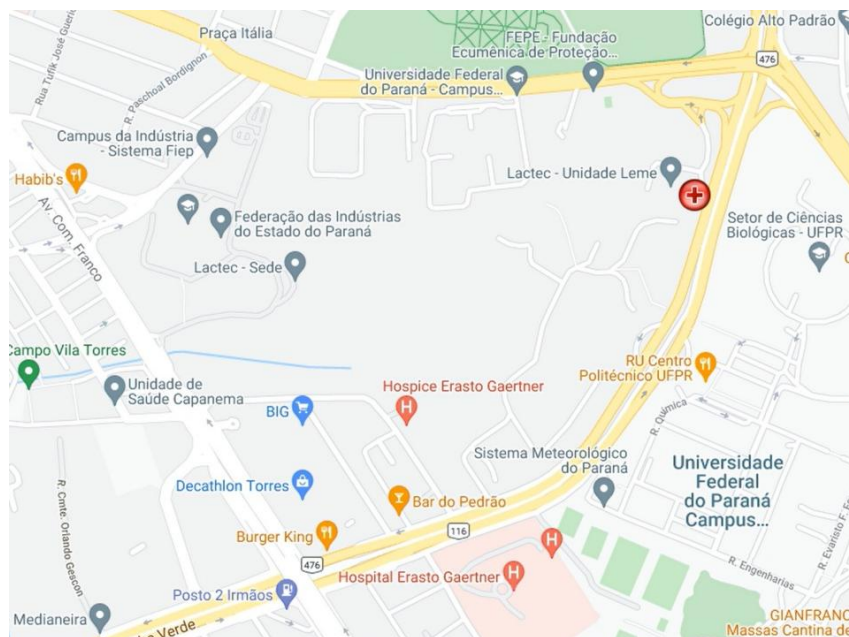
Figura 54 – Resultados dos testes de rastreamento do veículo em percurso entre as sedes do Lactec – tabela.

Rotas									
id	Cliente ID	Status	Tempo parado [s]	Tempo total [s]	Total [km]	Criado em	Atualizado em	Estação de retirada	Estação de devolução
108	1	Finished	20	814	2.64	2021-07-29 14:04:14	2021-07-29 14:17:48	2	1
111	1	Finished	125	1064	3.22	2021-07-29 15:09:25	2021-07-29 15:27:09	1	2

Fonte: Autoria própria. Gerada no software Grafana.

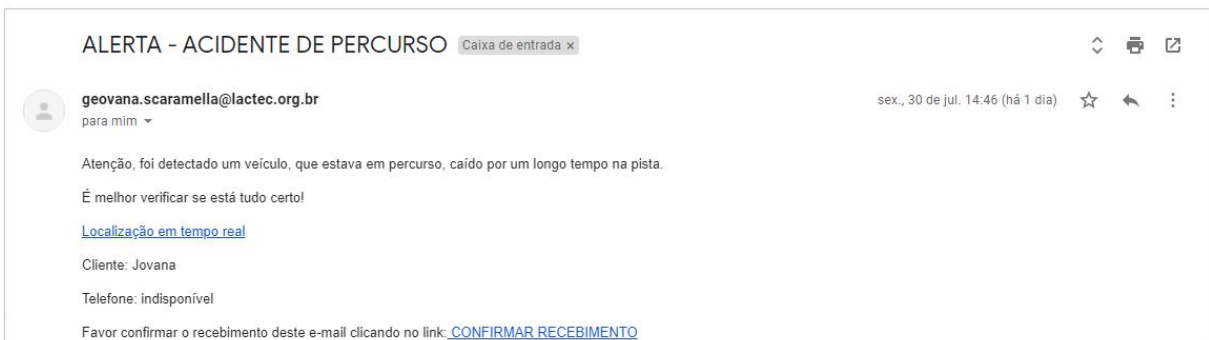
Por fim, foi feito mais um teste, realizando os mesmos dois percursos, com simulação de um possível acidente com o veículo no caminho, colocando-o na horizontal. Durante o percurso, os dados do veículo foram observados em tempo real por meio da interface *websocket* e de um mapa mostrado no navegador. Um marcador especial, de cruz vermelha, foi utilizado para representar o veículo em estado de “possível acidente”. No momento do acidente simulado, o mapa foi observado e o marcador estava sendo apresentado, como previsto. A Figura 55 apresenta uma captura da tela do *smartphone* no momento do acidente. Além disso, a notificação de possível acidente, apresentada na Figura 56, foi enviada ao supervisor no momento do ocorrido, conforme esperado. Como definido, após a confirmação do recebimento, a notificação não foi enviada periodicamente.

Figura 55 – Tela de acompanhamento do veículo em tempo real com possível acidente no percurso Leme-Sede.



Fonte: Autoria própria. Captura do navegador do *smartphone*.

Figura 56 – Notificação de possível acidente enviada durante teste no percurso Leme-Sede.



Fonte: Autoria própria. Captura do *gmail*.

4.2 MONITORAMENTO DE CARGA E DESCARGA DA BATERIA DOS VEÍCULOS

A Figura 57 apresenta as curvas de tensão da bateria do veículo durante os percursos, enquanto a Figura 58 apresenta o carregamento da bateria do veículo acoplado à estação Leme.

Figura 57 – Resultados de monitoramento da bateria do veículo no percurso da Sede ao Leme, acima, e do Leme à Sede, abaixo.



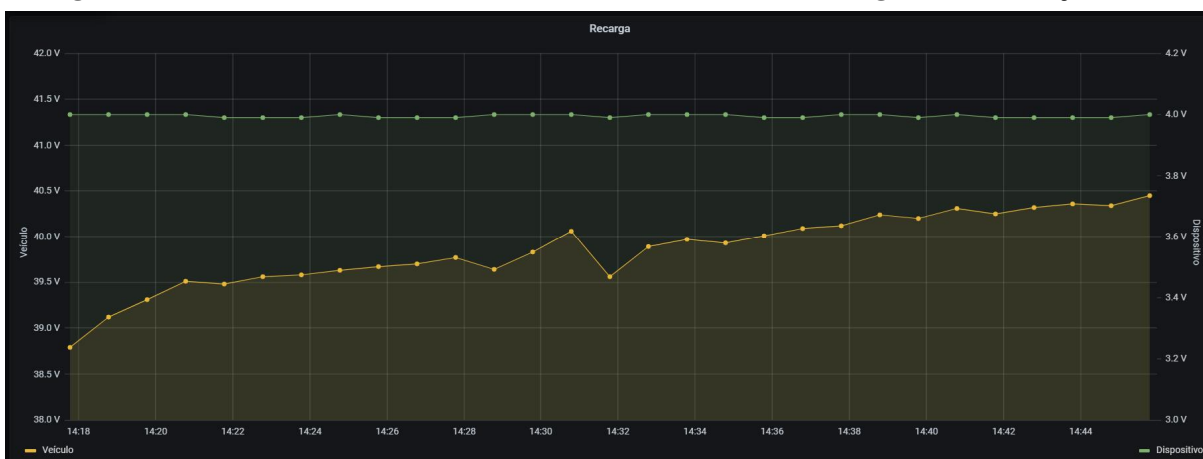
Fonte: Autoria própria. Gerada no software Grafana.

Para analisar um ciclo de carregamento completo, a bateria do veículo foi descarregada até atingir a tensão de 29V e então o veículo foi acoplado à estação de carregamento. O resultado do carregamento é apresentado na Figura 59.

4.3 ENVIO DE *DOWNLINKS* DE CONFIGURAÇÃO

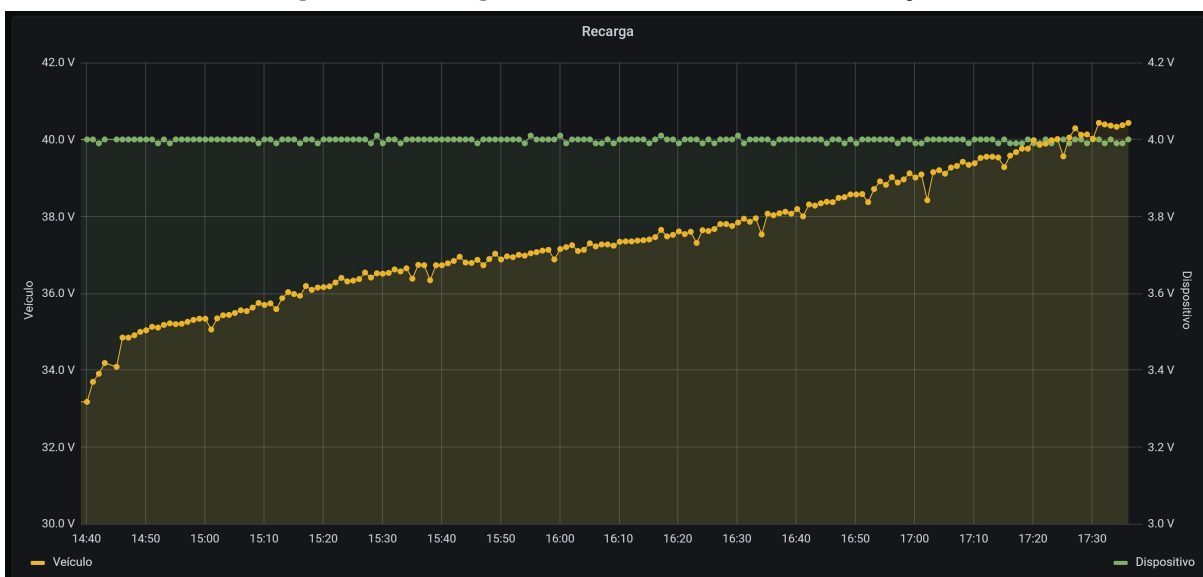
A validação da configuração dos parâmetros do dispositivo de monitoramento, por meio de mensagens de *downlink LoRaWAN*, foi feita utilizando a API desenvolvida pelos autores (Apêndice E). Com o método “PATCH” na rota “10.108.70.165/api/config”, os valores dos períodos de envio de mensagens em cada estado e os limiares de velocidade e erro de GPS, no

Figura 58 – Resultados de monitoramento da bateria do veículo em carregamento na estação Leme.



Fonte: Autoria própria. Gerada no *software* Grafana.

Figura 59 – Carregamento da bateria do veículo na estação.



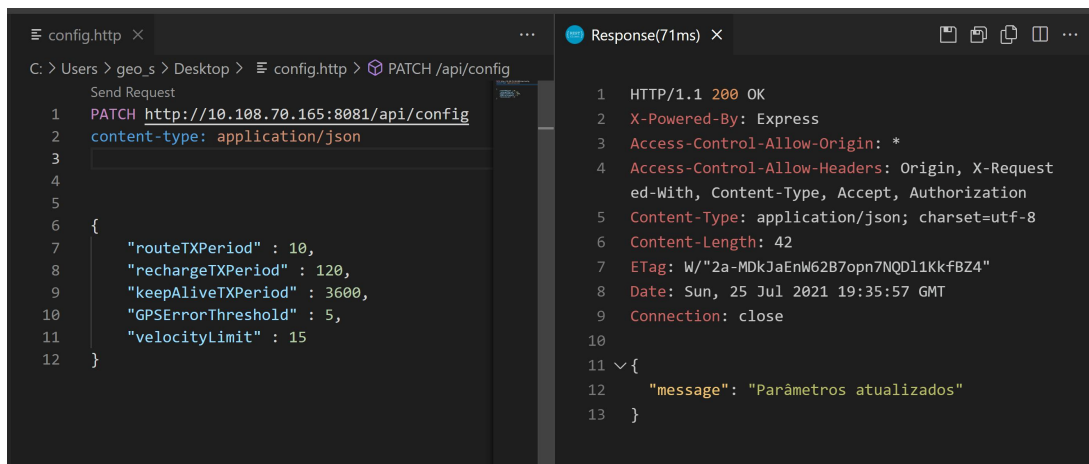
Fonte: Autoria própria. Gerada no *software* Grafana.

modo percurso, foram alterados. Na Figura 60 encontram-se os valores enviados ao dispositivo. No log do dispositivo, disponível na Figura 61, é possível verificar a alteração dos valores padrão, destacados em laranja, para os novos valores, em azul, após os *downlinks*, em verde.

Após a alteração dos valores, o dispositivo foi reiniciado para verificar a persistência dos mesmos. Este procedimento foi realizado tanto por meio do comando de *reset* da API (Figura 62), quanto por meio da desenergização do circuito. Em ambos casos, os valores permaneceram os configurados no procedimento anterior (Figura 63).

Além de configurações de valores válidos, tentativas de configuração utilizando valores fora da faixa permitida foram feitas para assegurar que isto não é possível. Uma mensagem de erro é gerada para informar o usuário (Figura 64).

Figura 60 – Configuração dos parâmetros do dispositivo de monitoramento por meio da API.



```
config.http x
C: > Users > geo_s > Desktop > config.http > PATCH /api/config
Send Request
1 PATCH http://10.108.70.165:8081/api/config
2 content-type: application/json
3
4
5
6 {
7   "routeTXPeriod" : 10,
8   "rechargeTXPeriod" : 120,
9   "keepAliveTXPeriod" : 3600,
10  "GPSErrorThreshold" : 5,
11  "velocityLimit" : 15
12 }

Response(71ms) x
1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Access-Control-Allow-Origin: *
4 Access-Control-Allow-Headers: Origin, X-Requested-With, Content-Type, Accept, Authorization
5 Content-Type: application/json; charset=utf-8
6 Content-Length: 42
7 ETag: W/"2a-MDkJaEnW62B7opn7NQD11KkfBZ4"
8 Date: Sun, 25 Jul 2021 19:35:57 GMT
9 Connection: close
10
11 {
12   "message": "Parâmetros atualizados"
13 }
```

Fonte: Autoria própria. Utilizando o software *Visual Studio Code*.

Figura 61 – Log do dispositivo durante a configuração remota utilizando a API.

```

7/25/2021 16:35:44.623 [RX] - [MPU] INICIALIZADO<LF>
APP_VERSION= 01.03.00.00<CR><LF>
MAC_VERSION= 04.04.02.00<CR><LF>
OTAA<LF><CR>
DevEui= <LF><CR>
AppEui= <LF><CR>
AppKey= <LF><CR>
[EEPROM]<LF>
Estado: ROTA<LF>
Buzzer:PERIODO_KEEP_ALIVE: 300000 [s]<LF>
PERIODO_ROTA: 5000 [s]<LF>
PERIODO_RECARGA: 60000 [s]<LF>
LIMITE_VELOCIDADE: 200/10 [km/h]<LF>
LIMIAR_GPS_DOP: 30/10<LF>
[EEPROM] PANIC FLAG: RESET<LF>
[LORA] APP_STATE_ROUTE<LF>
[GPS] Inicializado<LF>
[GPS] WAKEUP<LF>
  2s914: PHY txDone<LF><CR>
7/25/2021 16:35:50.443 [RX] - 8s029: PHY rxDone<LF><CR>
[LORA] JOINED<LF><CR>
[UPLINK] Rota<LF>
7/25/2021 16:35:57.031 [RX] - 14s610: PHY txDone<LF><CR>
7/25/2021 16:35:58.368 [RX] - 15s948: PHY rxDone<LF><CR>
[UPLINK] Rota<LF>
7/25/2021 16:36:02.188 [RX] - 19s775: PHY txDone<LF><CR>
7/25/2021 16:36:03.524 [RX] - 21s112: PHY rxDone<LF><CR>
[DOWNLINK] Configuracao de periodo de transmissao<LF>
[EEPROM]<LF>
Estado: ROTA<LF>
Buzzer:PERIODO_KEEP_ALIVE: 3600000 [s]<LF>
PERIODO_ROTA: 10000 [s]<LF>
PERIODO_RECARGA: 120000 [s]<LF>
LIMITE_VELOCIDADE: 200/10 [km/h]<LF>
LIMIAR_GPS_DOP: 30/10<LF>
[EEPROM] PANIC FLAG: RESET<LF>
[UPLINK] Rota<LF>
  22s290: PHY txDone<LF><CR>
7/25/2021 16:36:05.998 [RX] - 23s585: PHY rxDone<LF><CR>
[DOWNLINK] Configuracao de limite de velocidade<LF>
[EEPROM]<LF>
Estado: ROTA<LF>
Buzzer:PERIODO_KEEP_ALIVE: 3600000 [s]<LF>
PERIODO_ROTA: 10000 [s]<LF>
PERIODO_RECARGA: 120000 [s]<LF>
LIMITE_VELOCIDADE: 150/10 [km/h]<LF>
LIMIAR_GPS_DOP: 30/10<LF>
[EEPROM] PANIC FLAG: RESET<LF>
7/25/2021 16:36:07.183 [RX] - 24s763: PHY txDone<LF><CR>
7/25/2021 16:36:08.472 [RX] - 26s059: PHY rxDone<LF><CR>
[DOWNLINK] Configuracao de limiar de DOP do GPS<LF>
[EEPROM]<LF>
Estado: ROTA<LF>
Buzzer:PERIODO_KEEP_ALIVE: 3600000 [s]<LF>
PERIODO_ROTA: 10000 [s]<LF>
PERIODO_RECARGA: 120000 [s]<LF>
LIMITE_VELOCIDADE: 150/10 [km/h]<LF>
LIMIAR_GPS_DOP: 50/10<LF>
[EEPROM] PANIC FLAG: RESET<LF>
[UPLINK] Rota<LF>

```

Fonte: Autoria própria. Utilizando o software Docklight.

Figura 62 – Reinicialização do dispositivo por meio de comando da API.

The screenshot shows a REST client interface in Visual Studio Code. On the left, a request is defined with the method GET and the URL `http://10.108.70.165:8081/api/vehicle/5/reset`. On the right, the response is displayed, showing a 200 OK status and a JSON body: `{ "message": "Reset agendado" }`.

```

1 GET http://10.108.70.165:8081/api/vehicle/5/reset
2
3
4
5
6
7
8
9
10
11 {
12   "message": "Reset agendado"
13 }

```

Fonte: Autoria própria. Utilizando o software *Visual Studio Code*.

Figura 63 – Log do dispositivo ao reiniciar após o recebimento de um comando de *reset*.

```

[UPLINK] Rota<LF>

7/25/2021 16:47:27.014 [RX] - 704s611: PHY txDone<LF><CR>

7/25/2021 16:47:28.319 [RX] - 705s908: PHY rxDone<LF><CR>

7/25/2021 16:47:30.578 [RX] - [MPU] INICIALIZADO<LF>
APP_VERSION= 01.03.00.00<CR><LF>
MAC_VERSION= 04.04.02.00<CR><LF>
OTAA<LF><CR>
DevEui= <LF><CR>
AppEui= <LF><CR>
AppKey= <LF><CR>
[EEPROM]<LF>
Estado: ROTA<LF>
Buzzer:PERIODO_KEEP_ALIVE: 3600000 [s]<LF>
PERIODO_ROTA: 10000 [s]<LF>
PERIODO_RECARGA: 120000 [s]<LF>
LIMITE_VELOCIDADE: 150/10 [km/h]<LF>
LIMIAR_GPS_DOP: 50/10<LF>
[EEPROM] PANIC FLAG: RESET<LF>
[LORA] APP_STATE_ROUTE<LF>
[GPS] Inicializado<LF>
[GPS] WAKEUP<LF>
  2s912: PHY txDone<LF><CR>

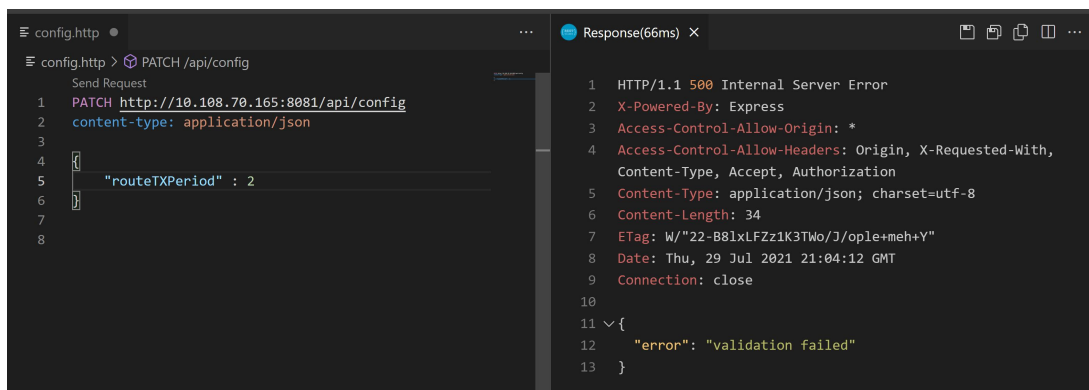
7/25/2021 16:47:36.396 [RX] - 8s027: PHY rxDone<LF><CR>
[LORA] JOINED<LF><CR>
[UPLINK] Rota<LF>

7/25/2021 16:47:42.969 [RX] - 14s607: PHY txDone<LF><CR>

```

Fonte: Autoria própria. Utilizando o software *Docklight*.

Figura 64 – Erro ao tentar configurar um parâmetro com um valor fora da faixa aceita.



```
config.http
PATCH /api/config
Send Request
1 PATCH http://10.108.70.165:8081/api/config
2 content-type: application/json
3
4 {
5   "routeTXPeriod" : 2
6 }
7
8

Response(66ms) X
1 HTTP/1.1 500 Internal Server Error
2 X-Powered-By: Express
3 Access-Control-Allow-Origin: *
4 Access-Control-Allow-Headers: Origin, X-Requested-With,
Content-Type, Accept, Authorization
5 Content-Type: application/json; charset=utf-8
6 Content-Length: 34
7 ETag: W/"22-B8lxLFZz1K3Two/J/ople+meh+Y"
8 Date: Thu, 29 Jul 2021 21:04:12 GMT
9 Connection: close
10
11 {
12   "error": "validation failed"
13 }
```

Fonte: Autoria própria. Utilizando o software *Visual Studio Code*.

5 CONCLUSÕES E PERSPECTIVAS

O projeto desenvolvido mostra que é possível implantar um sistema de monitoramento de veículos alternativos, com percursos pré-estabelecidos, por meio de uma rede *LoRaWAN*. Apesar de apresentar alguns pontos a melhorar na cobertura da rede *LoRaWAN* e na robustez da comunicação GPS, o sistema cumpriu a maioria dos requisitos técnicos estabelecidos, total ou parcialmente e, neste último caso, mostra indícios de que pequenas alterações podem resolver os aspectos em débito.

As funcionalidades referentes ao monitoramento da bateria, tanto em recarga quanto em descarga (em percurso) cumpriram os requisitos estabelecidos, sendo possível ao usuário consultar todas as curvas de tensão da bateria no tempo para os dois casos. Quanto ao monitoramento do percurso, quando não há falha na cobertura *LoRaWAN* e sinal de GPS, todos os requisitos também são satisfeitos, sendo eles: contagem de tempo de percurso e tempo parado, medição da distância percorrida, reconhecimento de estação de retirada e devolução, rastreamento da posição geográfica, monitoramento da velocidade e do respeito à área legal de circulação e reconhecimento de possível acidente com o veículo. O único requisito que não foi cumprido, em nenhuma circunstância, é a identificação do funcionário em posse do veículo. Esta funcionalidade depende da implementação de um meio de obtenção deste dado por parte da empresa fornecedora dos veículos, a Emove.

Com a implantação do sistema, será possível aos responsáveis pela manutenção do mesmo (o departamento de *Facilities* do Lactec) averiguar o histórico dos veículos para extrair qualquer tipo de informação, como investigar o uso feito por diferentes clientes, as rotas tomadas por eles, se houve quedas, o tempo de percurso entre as estações, entre outras. Com o registro das curvas de carga e descarga da bateria, será possível determinar se uma bateria ou veículo apresenta defeito e, com o tempo e após a geração de uma base de dados pelo uso do sistema, até prever defeitos e agendar manutenções preditivas. As notificações geradas quando o veículo sai da área estabelecida de circulação ou quando fica caído por alguns segundos podem auxiliar os supervisores a atenderem mais rapidamente essas situações, por exemplo, acionando a polícia em caso de um veículo que sai completamente da área e não retorna ou, em caso de acidente, prestar socorro ao condutor.

Para que o sistema esteja completamente adequado às necessidades do cliente, cumprindo todos os requisitos técnicos com excelência, algumas melhorias deverão ser realizadas

após a conclusão deste trabalho, conforme se segue:

1. Como se tratava de um protótipo, a adequação mecânica do *hardware* ao veículo foi apenas providencial e inadequada para um produto final. Um novo desenho de placa de circuito impresso deverá ser desenvolvido, com os principais componentes utilizados no projeto, integrando os circuitos de alimentação e o dispositivo IoT em um *design* mais compacto para ser instalado no interior do veículo de forma ótima.
2. Como foi observado, com o *hardware* atual, o GPS sofre com falhas na comunicação em alguns momentos. A nova placa desenvolvida deverá apresentar um funcionamento mais adequado do GPS, com menor incidência de falhas na comunicação, seja pela utilização de outra antena, outro módulo de comunicação GPS ou pelo melhor roteamento das ligações elétricas.
3. A possibilidade da instalação de um novo *gateway LoRaWAN*, no prédio Cehpar, onde fica uma das estações, deverá ser investigada, para melhorar a qualidade do sinal naquela região. Atualmente, a rota pré-estabelecida está totalmente coberta pela rede, porém, como foi apresentado, quando se sai da rota o dispositivo logo para de se comunicar apropriadamente.
4. Além da instalação de um novo *gateway*, a possibilidade do uso de uma nova antena no *end device* deverá ser investigada, já que a potência emitida pelo *gateway* já está configurada para o máximo permitido pela legislação e, como se pode notar nos testes de transpasse da área permitida, os *downlinks* não apresentam a mesma robustez de comunicação que os *uplinks*, mesmo com o maior fator de espalhamento. Uma antena especialmente adequada ao chassi do veículo deverá ser encontrada.
5. Uma integração *Web* com os serviços da empresa que mantém a estação deverá ser desenvolvida para permitir que mais informações sejam extraídas do sistema. O cliente a retirar o veículo, por exemplo, é uma informação que só estará disponível após essa integração, cuja possibilidade já foi confirmada pela empresa Emove.

Há ainda a possibilidade de revisão de alguns requisitos de projeto para que possam contornar problemas encontrados no decorrer do desenvolvimento do sistema. Este é o caso, por exemplo, do requisito de aviso sonoro ao usuário trafegando em área não permitida. O cálculo da detecção de transpasse de área é facilmente realizado no servidor, mas como o *downlink* não

apresentou resultados confiáveis, longe dos *gateways*, esta funcionalidade não foi atendida de maneira satisfatória. Uma alternativa seria realizar este monitoramento no próprio dispositivo, como é feito no caso de transpasse do limite de velocidade. Porém, o critério de delimitação de área precisa ser analisado, visto que o dispositivo *iot* não apresenta as mesmas capacidades de processamento que o servidor. Para resolver isso, poderia ser feita a redefinição do requisito, aceitando uma área de circulação menos complexa, como circunferência, em vez de um polígono, caso exista essa flexibilidade. Outra forma seria verificar se o bipe realmente traz algum benefício para o usuário do sistema ou se somente o aviso ao supervisor já é suficiente. Em todo caso, o aperfeiçoamento do enlace para melhorar o funcionamento dos *downlinks* é a melhor opção.

Por fim, feitas as melhorias apontadas e novos testes, oito dispositivos deverão ser confeccionados para a instalação em todos os veículos do Lactec, possibilitando assim o monitoramento em tempo real da frota em sua totalidade e o acompanhamento do projeto de mobilidade elétrica.

REFERÊNCIAS

ADELANTADO, Ferran; VILAJOSANA, Xavier; TUSET-PEIRO, Pere; MARTINEZ, Borja; MELIÀ-SEGUÍ, Joan; WATTEYNE, Thomas. Understanding the limits of lorawan. **IEEE Communications Magazine**, 2017.

AFISIADIS, Orion; BURG, Andreas; BALATSOUKAS-STIMMING, Alexios. **Coded LoRa Frame Error Rate Analysis**. 2019.

AFTAB, Noman; ZAIDI, Syed Ali Raza; MCLERNON, Des. Scalability analysis of multiple loragateways using stochastic geometry. 2019.

ALBUQUERQUE, Daniel Thomazini e Pedro Urbano Braga de. **Sensores Industriais - Fundamentos e Aplicações**. [S.l.: s.n.], 2009.

ALLIANCE, Inc LoRa. Lorawan 1.0.3 regional parameters. 2018.

AMERICAN TOWERS. **REDE ATC LoRaWAN™**. 2021. Disponível em: <https://americantower.com.br/pt/solu%C3%A7%C3%B5es/rede-neutra-loRaWAN.html>.

AMIN, Ershad Junus; SUKSMONO, Andriyan Bayu; MUNIR, Achmad. Accuracy analysis of fm chirp in gnu radio-based fmcw radar for multiple target detection. *In: 2014 International Conference on Computer, Control, Informatics and Its Applications (IC3INA)*. [S.l.: s.n.], 2014. p. 115–119.

BAHARUDIN, Ahmad Muzaffar bin; YAN, Wanglin. Long-range wireless sensor networks for geo-location tracking: Design and evaluation. 2016.

BEEBY, Stephen; ENSELL, Graham; KRAFT, Michael; WHITE, Neil. **MEMS Mechanical Sensors**. [S.l.: s.n.], 2004.

BIEHL, Matthias. **RESTful API Design**. [S.l.: s.n.], 2016.

CARVALHO, D. Fernandes; DEPARI, A.; PFERRARI; FLAMMINI, A.; RINALDI, S.; E.SISINNI. On the feasibility of mobile sensing and tracking applications based on lpwan. 2018.

CHIRPSTACK. **ChirpStack open-source LoRaWAN® Network Server**. 2021. Chirpstack. Disponível em: <https://www.chirpstack.io>.

CROCE, Daniele; GARLISI, Domenico; GIULIANO, Fabrizio; VALVO, Alice Lo; MANGIONE, Stefano; TINNIRELLO, Ilenia. Performance of lora for bike-sharing systems. *In: 2019 AEIT International Conference of Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE)*. [S.l.: s.n.], 2019. p. 1–6.

DANEBJER, Jonas; HALLDORSSON, Valthor. Gps-free geolocation for low cost iot devices. 2018.

DING, Jie; NEMATI, Mahyar; RANAWEERA, Chathurika; CHOI, Jinho. Iot connectivity technologies and applications:a survey. 2020.

DMITRIEVICH, Ilin Alexander; DMITRIEVICH, Belogurov Bogdan. Assessing the possibility of integration lpwantechology into tracking and monitoring systemsfor special vehicles. **International Siberian Conference on Control and Communications (SIBCON)**, 2019.

DRAGINO. **LGT-92 LoRaWAN GPS Tracker User Manual. v.1.4.3**. [S.l.], 2019.

DRAGINO. **LGT-92 v1.4 Schematics**. 2019.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de banco de dados**. [S.l.: s.n.], 2005.

ELSHABRAWY, Tallal; ROBERT, Joerg. Closed-form approximation of lora modulation ber performance. **IEEE**, set. 2018.

FARGAS, Bernat Carbonés; PETERSEN, Martin Nordal. Gps-free geolocation using lora in low-powerwans. 2017.

FIELDING, Roy Thomas. Architectural styles and the design of network-based software architectures. 2000.

FOUNDATION, OpenJS. **Introduction to NodeJS**. 2021. OpenJS Foundation. Disponível em: <https://nodejs.dev/learn>.

GARTNER. **Gartner Identifies Top 10 Strategic IoT Technologies and Trends**. 2018. Disponível em: <https://www.gartner.com/en/newsroom/press-releases/2018-11-07-gartner-identifies-top-10-strategic-iot-technologies-and-trends>.

GHADIRZADEH, Mohammadreza. Gps free geolocation in lora networks. 2017.

HALEEMUNNISA, Fatima; KUMUD, Wasnik. Comparison of sql, nosql and newsql databases for internet of things. 2016.

INDUSTRIES, The Things. **Enterprise LoRaWAN® connectivit.** 2021. The Things Industries. Disponível em: <https://www.thethingsindustries.com/>.

INVENSENSE. **MPU-9250 Product Specification Revision 1.1.** [S.l.], 2016.

IOTRACKER. **Track and Trace.** 2021. IoTracker. Disponível em: <https://www.iotracker.eu/track-trace>.

KAPLAN, Elliott; HEGARTY, Christopher. **Understanding GPS: principles and applications.** [S.l.]: ARTECH HOUSE INC, 2006. ISBN 1580538940.

KIM, Dong Hyun; PARK, Jung Bin; SHIN, Jae Ho; KIM, Jong Deok. Design and implementation of object trackign system based on lora. 2017.

LAIRD. **User Guide Sentrius RG1xx.** [S.l.], 2018.

LORA ALLIANCE. **RP002-1.0.2 LoRaWAN® Regional Parameters.** [S.l.], 2020.

MACKEY, Andrew; SPACHOS, Petros. Lora-based localization system for emergencyservices in gps-less environments. 2019.

MALUF, Nadim; WILLIAMS, Kirt. **An Introduction to Microelectromechanical Systems Engineering.** [S.l.: s.n.], 2004.

MEKKI, Kais; BAJIC, Eddy; CHAXEL, Frederic; MEYER, Fernand. A comparative study of LPWAN technologies for large-scale IoT deployment. **ICT Express**, Elsevier BV, v. 5, n. 1, p. 1–7, mar 2019.

MENGUE, Priscila. **Transporte compartilhado e bicicleta ganharão força no pós-pandemia, diz professor do MIT.** 2021. Terra. Disponível em: <https://www.terra.com.br/noticias/brasil/cidades/transporte-compartilhado-e-bicicleta-ganharao-forca-no-pos-pandemia-diz-professor-do-mit,ceeecea42c49353357bccbc0e40648eea7uyylha.html>.

MOUSER. **Laird Connectivity Gateways.** 2021. Mouser. Disponível em: https://br.mouser.com/Laird-Connectivity/Embedded-Solutions/Ethernet-Communication-Modules/Gateways/_/N-fb8zl?P=1y8g6p5.

MYDEVICES. **Cayenne Docs.** 2020. Disponível em: <https://developers.mydevices.com/cayenne/docs/intro/>.

NETWORK, The Things. **Building a global open LoRaWAN® network**. 2021. The Things Network. Disponível em: <https://www.thethingsnetwork.org/>.

NEWMAN, Peter; MATAN, Anne. Human mobility and human health. **Current Opinion in Environmental Sustainability**, v. 4, p. 420–426, 10 2012.

NMEA, National Marine Electronics Association. **Standard For Interfacing Electronic Devices - NMEA 0183**. [S.l.], 2002.

OLIVEIRA, Franklin; NERY, Dilan; COSTA, Daniel G.; SILVA, Ivanovitch; LIMA, Luciana. A survey of technologies and recent developments for sustainable smart cycling. **Sustainability**, MDPI AG, v. 13, n. 6, p. 3422, mar 2021.

PBSC. **The Meddin Bike-sharing World Map**. 2021. PBSC: Urban Solutions. Disponível em: <https://bikesharingworldmap.com/#/all/2.7/-336.67/13.83/>.

PETÄJÄRVI, Juha; MIKHAYLOV, Konstantin; ROIVAINEN, Antti; HÄNNINEN, Tuomo; PETTISSALO, Marko. On the coverage of lpwans: Range evaluation and channel attenuation model for lora technology. 2015.

PODEVIJN, Nico; PLETS, David; TROGH, Jens; MARTENS, Luc; SUANET, Pieter; HENDRIKSE, Kim; ; JOSEPH, Wout. Tdoa-based outdoor positioning with tracking algorithm in a public lora network. 2017.

PROAKIS, John. Digital communications. *In: _____*. 5. ed. Boston: McGraw-Hill, 2008. cap. Chapter 12: Spread Spectrum Signals for Digital Communications. ISBN 9780072957167.

QUECTEL. **L70-R Series GPS: Protocol Specification**. [S.l.], 2016. 39 p. V3.0.

RADIOENGE. **Gateway LoRaWAN**. 2021. Radioenge. Disponível em: <https://www.radioenge.com.br/solucoes/iot/18-gateway-lorawan.html>.

RAK. **RAKwireless – IoT Made Easy**. 2021. RAK. Disponível em: <https://www.rakwireless.com/en-us>.

RAUTMARE, Sharvari; BHALERAO, D. M. Mysql and nosql database comparison for iot application. 2016.

RINALDI, Stefano; BONAFINI, Federico; FERRARI, Paolo; FLAMMINI, Alessandra; SISINNI, Emiliano; BIANCHINI, Devis. Impact of data model on performance of time series database for internet of things applications. 2019.

RIVEIRA, Carolina; BRANCO, Leo; INGIZZA, Carolina. **Fim da moda, prejuízo e pandemia: como Grow, ex-Yellow, foi do boom à lona**. 2020. Exame. Disponível em: <https://exame.com/pme/fim-da-moda-prejuizo-e-pandemia-como-grow-ex-yellow-foi-do-boom-a-lona/>.

ROBYNS, Pieter; QUAX, Peter; LAMOTTE, Wim; THENAERS, William. A multi-channel software decoder for the LoRa modulation scheme. *In: Proceedings of the 3rd International Conference on Internet of Things, Big Data and Security*. [S.l.]: SCITEPRESS - Science and Technology Publications, 2018.

ROHS. **CEM-1203(42): Magnetic buzzer**. [S.l.], 2006.

SEMTECH. **LoRa® and LoRaWAN®: A Technical Overview**. [S.l.], 2019.

SEMTECH. **SX1272/73 - 860 MHz to 1020 MHz Low Power Long Range Transceiver**. [S.l.], 2019. Rev. 4.

SEMTECH CORPORATION. **AN1200.22: LoRa Modulation Basics**. [S.l.], 2015. Revision 2.

SIGFOX. **Sigfox: Technical Overview**. [S.l.], 2017.

SINHA, Rashmi Sharan; WEI, Yiqiao; HWANG, Seung-Hoon. A survey on lpwa technology: Lora and nb-iot. **ICT Express**, 2017.

STATISTA. **Global bike-sharing service market size between 2020 and 2026**. 2021. Statista. Disponível em: <https://www.statista.com/statistics/868126/global-bikesharing-market-size/>.

STMICROELECTRONICS. **Ultra-low-power 32-bit MCU Arm®-based Cortex®-M0+, up to 192KB Flash, 20KB SRAM, 6KB EEPROM, USB, ADC, DACs**. 2017.

SUBRAMANIAN, Harihara; RAJ, Pethuru. **Hands-On RESTful API Design Patterns and Best Practices**. [S.l.: s.n.], 2019.

TEKTELIC. **Tektelic Communication**. 2021. Tektelic. Disponível em: <https://tektelic.com/>.

TEXAS INSTRUMENTS. **LM2596 SIMPLE SWITCHER Power Converter 150-kHz 3-A Step-Down Voltage Regulator**. [S.l.], 2016.

TILKOV, Stefan; VINOSKI, Steve. Node.js: Using javascript to build high-performance network programs. 2010.

VOIGT, Thiemo; BOR, Martin; ROEDIG, Utz; ALONSO, Juan. Mitigating inter-network interference in lora networks. 2016.

ZEE, Renate van der. **Story of cities 30: how this Amsterdam inventor gave bike-sharing to the world.** 2016. The Guardian. Disponível em: <https://www.theguardian.com/cities/2016/apr/26/story-cities-amsterdam-bike-share-scheme>.

ZTRACK. **We can Track everything.** 2021. ZTrack. Disponível em: <https://ztrackmap.com/>.

APÊNDICES

APÊNDICE A – ESPECIFICAÇÃO DE PROBLEMAS, NECESSIDADES E REQUISITOS

PROBLEMAS

- P1 O cliente deve saber quando a bateria de um veículo de sua frota está apresentando um perfil de carregamento ou descarregamento fora do padrão, sob pena de: ter que arcar com os custos de uma maior frequência na troca das baterias, por não saber quando as baterias precisam de fato ser trocadas (manutenção preventiva); ter que arcar com os custos da troca de baterias quando estas são danificadas por um veículo que apresenta defeito no consumo de energia elétrica por um período prolongado (manutenção corretiva).
- P2 O cliente deve saber quando um veículo de sua frota excede os limites da área permitida de circulação, sob pena de: não saber se as pessoas estão fazendo o uso correto dos patinetes, andando somente dentro da área permitida; não saber quando um patinete é roubado, atrasando o acionamento da polícia e diminuindo a chance de recuperação do patrimônio.
- P3 O cliente deve saber o perfil de uso dos veículos de sua frota para avaliação da aceitabilidade pelos funcionários do Lactec e da viabilidade de forma contínua, sob pena de arcar com custos de manutenção de um sistema que não traga ganhos para a empresa e para o meio ambiente.
- P4 O Lactec deve desenvolver um sistema de monitoramento de veículos alternativos que incentive a produção de novas soluções em sistemas de *Internet of Things* (IoT), sob pena de aumento de custos de oportunidade, devido, por exemplo, ao não atendimento das demandas do mercado.
- P5 O cliente deve garantir que os usuários de seus veículos respeitem as limitações de velocidade e de área de circulação, sob pena de ter usuários acidentados pelo uso do veículo acima do limite de velocidade permitido ou em locais não adequados, causando abandono do uso do sistema.

NECESSIDADES

- N1 O cliente precisa de um sistema para consultar o histórico das informações da bateria dos veículos de sua frota, tanto relativas ao carregamento nas estações quanto ao descarrega-

mento durante o percurso.

N2 O cliente precisa de um sistema que o notifique quando um veículo sai da área permitida de circulação.

N3 O cliente precisa de um sistema que disponibilize em tempo real a localização e a velocidade dos veículos em percurso.

N4 O cliente precisa de um sistema para consultar o histórico das informações (localização e velocidade) dos percursos realizados com os veículos de sua frota.

N5 O cliente precisa de um sistema que avise o usuário quando ele está acima do limite de velocidade ou fora da área permitida de circulação.

N6 O cliente precisa de um sistema que notifique na ocorrência de queda com o veículo.

REQUISITOS TÉCNICOS

Requisitos técnicos do sistema

RS1 O sistema deverá permitir ao usuário cadastrar um novo veículo.

RS1.1 (NF) As informações de cadastro de um veículo são: identificação do veículo; identificação do dispositivo IoT; tipo do veículo (patinete ou bicicleta).

RS2 O sistema deverá permitir ao usuário excluir um veículo escolhido.

RS3 O sistema deverá permitir ao usuário alterar as informações de cadastro de um veículo escolhido.

RS4 O sistema deverá permitir ao usuário consultar os veículos cadastrados.

RS4.1 (NF) A resposta da consulta deverá ser uma lista com todos os veículos cadastrados.

RS4.2 (NF) Cada item da lista deverá conter as seguintes informações: identificação do veículo; identificação do dispositivo IoT; tipo do veículo; instante de tempo da última vez que um dado foi recebido; tensão da bateria do dispositivo IoT na última mensagem; quantidade de vezes que o veículo foi utilizado em percurso; quantidade de tempo pelo qual o veículo foi utilizado em percurso; distância total percorrida pelo veículo.

RS5 O sistema deverá armazenar em um banco de dados a estação e o momento nos quais um veículo é retirado de uma estação de carregamento.

RS5.1 O sistema deverá ser capaz de identificar quando um veículo é retirado da estação de carregamento e qual foi o veículo retirado.

RS6 O sistema deverá armazenar em um banco de dados as informações do veículo em percurso entre uma estação e outra.

RS7 O sistema deverá permitir ao usuário receber as informações de percurso de um veículo em tempo real.

RS8 (NF) As informações de percurso são: posicionamento GPS (latitude, longitude, velocidade, DOP- *Dilution Of Precision*); indicador de estado parado ou em movimento; indicador de detecção de possível acidente; bateria do veículo; bateria do dispositivo IoT; instante de tempo.

RS9 O sistema deverá manter atualizado em banco de dados a distância total percorrida por um veículo.

RS9.1 O sistema deverá ser capaz de identificar a distância percorrida pelo veículo entre cada dado de percurso e incrementar a informação de distância total percorrida pelo veículo salva no banco de dados.

RS9.1.1 O sistema deverá desconsiderar, no cálculo da distância percorrida, dados com erro do GPS (DOP) acima do limite especificado.

RS10 O sistema deverá manter atualizado em banco de dados o total de vezes que um veículo foi utilizado em percurso.

RS11 O sistema deverá manter atualizado em banco de dados o total de tempo pelo qual um veículo foi utilizado em percurso.

RS12 O sistema deverá armazenar em um banco de dados a estação e o momento nos quais um veículo é recolocado em uma estação de carregamento.

RS12.1 O sistema deverá ser capaz de identificar quando um veículo é recolocado em uma estação de carregamento e qual foi o veículo recolocado.

- RS12.2 O sistema deverá finalizar o armazenamento dos dados de percurso quando reconhecer a recolocação do veículo em uma estação.
- RS13 O sistema deverá armazenar em um banco de dados as informações do veículo durante a recarga em uma estação.
- RS13.1 (NF) As informações de recarga são: bateria do veículo; bateria do dispositivo IoT; instante de tempo.
- RS14 O sistema deverá finalizar o armazenamento dos dados de recarga quando o carregamento for completado.
- RS15 O sistema deverá finalizar o armazenamento dos dados de recarga quando o carregamento for interrompido pela retirada do veículo da estação.
- RS16 O sistema deverá enviar uma notificação a um responsável quando reconhecer um possível acidente com um veículo.
- RS16.1 O sistema deverá ser capaz de reconhecer uma possibilidade de acidente com um veículo.
- RS16.1.1 (NF) Uma situação de possível acidente é caracterizada pela posição anormal do veículo em relação ao chão.
- RS16.2 (NF) A notificação deverá conter as seguintes informações: identificação do veículo; localização.
- RS16.3 (NF) As notificações de possível acidente deverão ser enviadas por e-mail.
- RS16.4 O sistema deverá permitir ao usuário confirmar o recebimento da notificação de possível acidente.
- RS16.5 O sistema deverá reenviar as notificações de possível acidente periodicamente até que seja confirmado o recebimento pelo responsável.
- RS16.5.1 O sistema deverá permitir ao usuário configurar o período de reenvio de notificações de possível acidente.
- RS16.5.2 (NF) O período de reenvio mínimo é de 30 segundos.
- RS16.5.3 (NF) O período de reenvio máximo é de 3 minutos.
- RS17 O sistema deverá permitir ao usuário cadastrar um novo responsável para receber notificações de possível acidente.

- RS18 O sistema deverá permitir ao usuário alterar as informações cadastrais do responsável por receber notificações de possível acidente.
- RS19 O sistema deverá permitir ao usuário excluir um responsável por receber notificações de possível acidente.
- RS20 O sistema deverá enviar uma notificação a um responsável quando reconhecer um veículo fora da área permitida de circulação, a não ser que o erro do GPS (DOP) esteja acima do limite configurado.
- RS20.1 O sistema deverá ser capaz de reconhecer que um veículo saiu da área permitida de circulação.
- RS20.2 O sistema deverá acionar um aviso sonoro no veículo quando este estiver fora da área permitida de circulação, a não ser que o erro do GPS (DOP) esteja acima do limite configurado.
- RS20.3 O sistema deverá reconhecer que um veículo voltou para a área permitida de circulação.
- RS20.4 O sistema deverá interromper o aviso sonoro no veículo quando este retornar à área de circulação.
- RS20.5 O sistema deverá enviar uma notificação ao responsável quando reconhecer que o veículo voltou à área permitida de circulação.
- RS20.6 (NF) A notificação deverá conter as seguintes informações: identificação do veículo; localização.
- RS21 O sistema deverá permitir ao usuário cadastrar uma nova área de circulação.
- RS21.1 (NF) Uma área de circulação é definida por um conjunto de pontos de latitude e longitude.
- RS22 O sistema deverá permitir ao usuário consultar uma lista com as áreas de circulação cadastradas.
- RS23 O sistema deverá permitir ao usuário excluir uma área de circulação cadastrada.
- RS24 O sistema deverá permitir ao usuário selecionar qual área de circulação cadastrada deve ser usada.

RS25 O sistema deverá reenviar as notificações de veículo fora de área periodicamente até que seja confirmado o recebimento pelo responsável.

RS25.1 O sistema deverá permitir ao usuário configurar o período de reenvio de notificações de veículo fora de área.

RS25.1.1 (NF) O período de reenvio mínimo é de 30 segundos.

RS25.1.2 (NF) O período de reenvio máximo é de 3 minutos.

RS26 O sistema deverá acionar um aviso sonoro no veículo quando este estiver acima do limite de velocidade.

RS26.1 O sistema deverá ser capaz de reconhecer quando um veículo está acima da velocidade máxima permitida.

RS26.2 O sistema deverá permitir ao usuário configurar qual o limite de velocidade permitido.

RS26.3 O sistema deverá ser capaz de reconhecer quando um veículo retorna para abaixo da velocidade máxima permitida.

RS26.4 O sistema deverá desativar o aviso sonoro no veículo quando este voltar para abaixo da velocidade máxima permitida.

RS26.5 (NF) O valor de limite de velocidade mínimo é de 15km/h.

RS26.6 (NF) O valor de limite de velocidade máximo é de 25km/h.

RS27 O sistema deverá permitir ao usuário configurar o período de envio de dados de percurso do veículo.

RS27.1 (NF) O mínimo valor configurável é de 5 segundos.

RS27.2 (NF) O máximo valor configurável é de 30 segundos.

RS28 O sistema deverá permitir ao usuário configurar o período de envio de dados de recarga do veículo.

RS28.1 (NF) O mínimo valor configurável é de 5 segundos.

RS28.2 (NF) O máximo valor configurável é de 300 segundos.

RS29 O sistema deverá ser capaz de manter a comunicação entre os dispositivos IoT dos veículos e o *software* remoto por meio mensagens de *keep-alive*.

- RS29.1 O sistema deverá permitir ao usuário configurar o período de envio de mensagem de *keep alive* de rede do dispositivo IoT do veículo.
- RS29.2 (NF) O mínimo valor configurável é de 5 segundos.
- RS29.3 (NF) O máximo valor configurável é de 1 hora.
- RS30 O sistema deverá permitir ao usuário consultar o histórico de percursos realizados com os veículos monitorados.
- RS30.1 O sistema deverá permitir ao usuário filtrar a busca por percursos.
- RS30.1.1 (NF) A filtragem da busca por percursos poderá ser feita por: identificação do veículo; identificação do cliente; data inicial (só são considerados percursos que acontecem a partir dessa data); data final (só são considerados percursos que acontecem até esta data).
- RS30.2 (NF) A resposta da consulta deverá ser uma lista com todos os percursos em conformidade com os filtros selecionados.
- RS30.3 (NF) Cada percurso da lista deverá conter as seguintes informações: identificação do percurso; identificação do veículo; identificação da estação de origem; identificação da estação de destino; estado do percurso ("finalizado" ou "não finalizado"); tempo de percurso com o veículo parado; tempo total do percurso; distância percorrida; instante de início do percurso; instante de finalização do percurso; identificação do cliente que fez o percurso.
- RS31 O sistema deverá permitir ao usuário consultar os dados de um percurso do histórico.
- RS31.1 (NF) A resposta da consulta deverá conter as informações referentes ao percurso: identificação do veículo; identificação da estação de origem; identificação da estação de destino; estado do percurso ("finalizado" ou "não finalizado"); tempo de percurso com o veículo parado; tempo total do percurso; distância percorrida; instante de início do percurso; instante de finalização do percurso; identificação do cliente que fez o percurso.
- RS31.2 (NF) A resposta da consulta deverá conter uma lista com todos os dados referentes às informações do veículo no percurso.
- RS32 O sistema deverá permitir ao usuário consultar o histórico de recargas de bateria dos veículos monitorados.

- RS32.1 O sistema deverá permitir ao usuário filtrar a busca por recargas.
- RS32.1.1 (NF) A filtragem poderá ser feita por: identificação do veículo; data inicial (só são consideradas recargas que acontecem a partir dessa data); data final (só são consideradas recargas que acontecem até esta data).
- RS32.2 (NF) A resposta da consulta deverá ser uma lista com todas as recargas em conformidade com os filtros selecionados.
- RS32.3 (NF) Cada recarga da lista deverá conter as seguintes informações: identificação do veículo; identificação da estação de recarga; estado da recarga ("finalizada" ou "não finalizada"); instante de início da recarga; instante de finalização da recarga;.
- RS33 O sistema deverá permitir ao usuário consultar todos os dados de um processo de recarga do histórico.
- RS33.1 (NF) A resposta da consulta deverá conter as informações referentes à recarga: identificação do veículo; identificação da estação de recarga; estado da recarga ("finalizada" ou "não finalizada"); instante de início da recarga; instante de finalização da recarga;.
- RS33.2 (NF) A resposta da consulta deverá conter uma lista com todos os dados referentes às informações do veículo na recarga.
- RS34 O sistema deverá permitir ao usuário reiniciar o dispositivo IoT de um veículo remotamente.
- RS35 O sistema deverá permitir ao usuário configurar o erro do GPS (DOP) que deve ser considerado como limiar entre dados com boa e ruim exatidão.
- RS36 O sistema deverá permitir ao usuário ativar o alarme do veículo remotamente.
- RS37 O sistema deverá permitir ao usuário desativar o alarme do veículo remotamente.
- RS38 O sistema deverá permitir ao usuário consultar estatísticas gerais de uso dos veículos monitorados.
- RS38.1 (NF) A resposta da consulta deverá conter: a média da quantidade de percursos por dia útil com os veículos monitorados, considerando sábados; a média da quantidade de percursos por dia útil com os veículos monitorados, desconsiderando sábados; a média de tempo de uso por dia útil dos veículos monitorados.

RS39 O sistema deverá ser capaz de monitorar os veículos em todas as vias disponíveis para circulação pelos funcionários do Lactec.

RS40 O sistema deverá ser capaz de monitorar os veículos dentro de uma área definida por um raio de pelo menos 3km a partir de cada estação de carregamento.

RS41 O sistema deverá ter custo fixo nulo.

RS42 O sistema deverá apresentar baixo índice de manutenção, não aumentando a frequência de manutenção necessária existente hoje para os veículos.

Requisitos técnicos do sistema embarcado

RSE1 O sistema embarcado deverá ser capaz de se conectar à bateria do veículo para recebimento de energia elétrica.

RSE2 O sistema embarcado deverá ter uma bateria própria para funcionamento independente da bateria do veículo.

RSE2.1 O sistema embarcado deverá ser capaz de carregar sua própria bateria pela alimentação fornecida pela bateria do veículo.

RSE3 O sistema embarcado deverá ser capaz de ler o valor de tensão da bateria do veículo.

RSE4 O sistema embarcado deverá ser capaz de ler o valor de tensão de sua própria bateria.

RSE5 O sistema embarcado deverá ser capaz de interromper a alimentação recebida da bateria do veículo caso a tensão desta caia abaixo de um valor limite.

RSE5.1 (NF) O valor da tensão limite é de 30V.

RSE6 O sistema embarcado deverá possibilitar a instalação e o funcionamento no interior do chassi sem causar alterações no visual do veículo.

RSE7 O sistema embarcado deverá apresentar baixo consumo de energia elétrica.

RSE7.1 (NF) O consumo deverá ser baixo o suficiente para que, mesmo sem a alimentação pela bateria do veículo, o sistema embarcado tenha uma autonomia de pelo menos 3 dias enviando dados de percurso com o período mínimo entre mensagens.

RSE8 O sistema embarcado deverá ser capaz de se comunicar com o servidor de aplicação.

RSE8.1 (NF) A camada física da comunicação deverá ser do tipo *LoRa*.

RSE8.2 (NF) A camada de enlace da comunicação deverá ser do tipo *LoRaWAN*.

RSE8.3 (NF) O protocolo de comunicação deverá ser um protocolo para dispositivos de baixo consumo de energia.

RSE8.3.1 (NF) O protocolo de comunicação com o servidor de aplicação deverá ser o CayenneLPP.

RSE9 O sistema embarcado deverá informar ao servidor da aplicação quando o veículo é retirado da estação de carregamento.

RSE9.1 O sistema embarcado deverá ser capaz de detectar quando o veículo é desacoplado da estação de carregamento.

RSE10 O sistema embarcado deverá informar ao servidor da aplicação quando o veículo é recolocado na estação.

RSE10.1 O sistema embarcado deverá ser capaz de detectar quando o veículo é acoplado à estação de carregamento.

RSE11 O sistema embarcado deverá enviar periodicamente mensagens com as informações de percurso do veículo ao servidor de aplicação quando o veículo estiver em percurso.

RSE11.1 (NF) Um veículo é considerado em percurso desde a sua retirada de uma estação de carregamento até o acoplamento a uma outra (ou à mesma) estação de carregamento.

RSE11.2 (NF) As informações de percurso são: posicionamento GPS (latitude, longitude, velocidade, DOP- *Dilution Of Precision*); indicador de estado parado ou em movimento; indicador de detecção de possível acidente; bateria do veículo; bateria do dispositivo IoT; instante de tempo.

RSE11.2.1 O sistema embarcado deverá ser capaz de identificar se o veículo está em movimento ou está parado em relação ao solo.

RSE11.2.1.1 (NF) A identificação deverá ser feita pelo uso de um acelerômetro.

RSE11.2.2 O sistema embarcado deverá ser capaz de se comunicar com um Sistema Global de Navegação por Satélite (GNSS).

RSE11.2.2.1 (NF) O sistema será o Sistema de Posicionamento Global (GPS).

RSE11.2.3 O sistema embarcado deverá ser capaz de identificar se os eixos do veículo se encontram numa posição não usual, de possível acidente, em relação ao solo.

RSE11.2.3.1 (NF) A identificação deverá ser feita a partir de um acelerômetro.

RSE11.2.3.2 (NF) É considerada uma posição não usual, de possível acidente, quando o eixo normalmente vertical do veículo se encontra com uma inclinação de pelo menos 45° em relação à posição normal.

RSE12 O sistema embarcado deverá conter um relógio sincronizado com o Tempo Universal Coordenado (UTC).

RSE12.1 (NF) A sincronização inicial deverá ser feita por meio do Sistema de Posicionamento Global (GPS).

RSE12.2 (NF) A sincronização deverá ser mantida com um RTC (Real Time Clock).

RSE13 O sistema embarcado deverá acionar um aviso sonoro quando o veículo ultrapassar a velocidade máxima permitida.

RSE13.1 O sistema embarcado deverá ser capaz de receber mensagens de configuração da velocidade máxima permitida.

RSE13.2 O sistema embarcado deverá interromper o aviso sonoro quando o veículo retornar para abaixo da velocidade máxima permitida.

RSE14 O sistema embarcado deverá acionar um aviso sonoro quando o veículo estiver fora da área permitida de circulação.

RSE14.1 O sistema embarcado deverá ser capaz de receber uma mensagem do servidor de aplicação informando que o veículo se encontra fora da área permitida de circulação.

RSE14.2 O sistema embarcado deverá interromper o aviso sonoro quando o veículo retornar para dentro da área permitida de circulação.

RSE14.2.1 O sistema embarcado deverá ser capaz de receber uma mensagem do servidor de aplicação informando que o veículo retornou à área permitida de circulação.

RSE15 O sistema embarcado deverá acionar um aviso sonoro quando o veículo estiver com o nível de tensão da bateria abaixo de um valor limite.

RSE15.1 (NF) O valor limite é de 30V.

RSE15.2 O sistema embarcado deverá interromper o aviso sonoro quando o veículo for acoplado a uma estação de carregamento.

RSE15.3 (NF) O acionamento do aviso sonoro deverá ter baixo impacto no consumo de energia elétrica, utilizando um grande espaçamento no tempo entre cada emissão.

RSE16 O sistema embarcado deverá enviar periodicamente mensagens com as informações de recarga da bateria do veículo ao servidor de aplicação quando o veículo estiver em recarga de bateria.

RSE16.1 (NF) Um veículo é considerado em recarga desde o momento em que é acoplado a uma estação de carregamento até que seja retirado desta estação ou até que a tensão da bateria do veículo atinja o valor limite.

RSE16.1.1 (NF) O valor limite de carregamento da bateria é de 42V.

RSE16.2 (NF) As informações de recarga são: bateria do veículo; bateria do dispositivo IoT; instante de tempo.

RSE17 O sistema embarcado deverá enviar periodicamente mensagens de *keep-alive* ao servidor de aplicação quando o veículo não estiver em recarga e nem em percurso.

RSE18 O sistema embarcado deverá ser capaz de receber mensagens de configuração do período de envio de dados de percurso.

RSE18.1 (NF) O mínimo valor configurável é de 5 segundos.

RSE18.2 (NF) O máximo valor configurável é de 30 segundos.

RSE19 O sistema embarcado deverá ser capaz de receber mensagens de configuração do período de envio de dados de recarga.

RSE19.1 (NF) O mínimo valor configurável é de 5 segundos.

RSE19.2 (NF) O máximo valor configurável é de 300 segundos.

RSE20 O sistema embarcado deverá ser capaz de receber mensagens de configuração do período de envio de dados de *keep-alive*.

RSE20.1 (NF) O mínimo valor configurável é de 5 segundos.

RSE20.2 (NF) O máximo valor configurável é de 1 hora.

RSE21 O sistema embarcado deverá ser capaz de receber mensagens de configuração do limite de velocidade.

RSE21.1 (NF) O valor de limite de velocidade mínimo é de 15km/h.

RSE21.2 (NF) O valor de limite de velocidade máximo é de 25km/h.

RSE22 O sistema embarcado deverá ser capaz de receber um comando, do servidor de aplicação, para acionamento de aviso sonoro.

RSE23 O sistema embarcado deverá ser capaz de receber um comando para interrupção do aviso sonoro, do servidor de aplicação.

RSE24 O sistema embarcado deverá ser capaz de receber uma mensagem para configuração do erro do GPS (DOP) que deve ser considerado como limiar entre dados com boa e ruim exatidão.

RSE25 O sistema embarcado deverá ser capaz de receber um comando de *reset*, do servidor de aplicação.

Requisitos técnicos de *hardware* do sistema embarcado

RHW1 O *hardware* deverá ser capaz de suprir todos os níveis de tensão e corrente necessários para operação da eletrônica a partir da alimentação fornecida pela bateria do veículo.

RHW1.1 (NF) A conexão da bateria do veículo com o *hardware* deverá ser feita por meio de conectores do tipo *microfit*.

RHW1.2 (NF) O intervalo de tensão da bateria do veículo deverá ser considerado entre 30 e 40V.

RHW1.2.1 O *hardware* deverá limitar a tensão de alimentação recebida à tensão máxima considerada mesmo quando a bateria do veículo estiver acima desse valor.

RHW1.2.2 O *hardware* deverá cortar a alimentação recebida, anulando a corrente elétrica, quando a tensão da bateria do veículo estiver abaixo da tensão mínima considerada.

RHW2 O *hardware* deverá ter suporte a uma bateria própria.

RHW2.1 (NF) A bateria própria será de 4,2V.

RHW2.2 O *hardware* deverá ser capaz de suprir todos os níveis de tensão e corrente necessários para operação da eletrônica a partir da alimentação fornecida pela bateria própria.

RHW2.3 O *hardware* deverá ser capaz de carregar a própria bateria a partir da alimentação fornecida pela bateria do veículo.

RHW3 (NF) O *hardware* deverá conter um dispositivo programável em que seja possível embarcar o *software* do sistema.

(NF) O dispositivo programável será um microcontrolador

RHW4 (NF) O *hardware* deverá conter um acelerômetro.

RHW4.1 O *hardware* deverá permitir que os valores medidos pelo acelerômetro sejam lidos pelo *software* embarcado no dispositivo programável.

RHW4.1.1 A comunicação entre o dispositivo programável e o módulo do acelerômetro deverá ser feita por uma interface serial I2C ou SPI.

RHW5 (NF) O *hardware* deverá conter um módulo de comunicação com GPS.

RHW5.1 O *hardware* deverá permitir que os valores de georreferenciamento recebidos pelo módulo de GPS sejam lidos pelo *software* embarcado no dispositivo programável.

RHW5.1.1 A comunicação entre o dispositivo programável e o módulo de GPS deverá ser feita por uma interface serial.

RHW5.2 O *hardware* deverá permitir o desligamento do módulo GPS para diminuir o consumo de energia.

RHW6 (NF) O *hardware* deverá conter um módulo de rádio *LoRa*.

RHW6.1 O *hardware* deverá permitir a comunicação entre o módulo de rádio *LoRa* e o *software* embarcado no dispositivo programável.

RHW7 (NF) O *hardware* deverá conter um módulo de *Real Time Clock*.

RHW7.1 (NF) O módulo de RTC utilizado será um módulo interno do dispositivo programável.

RHW8 O *hardware* deverá permitir a aquisição digital do valor de tensão elétrica da bateria do veículo e da bateria do dispositivo IoT.

RHW8.1 O *hardware* deverá permitir que os sinais digitais da leitura das tensões das baterias sejam lidos por meio de *software* embarcado no dispositivo programável.

RHW8.1.1 (NF) O módulo de conversão analógico-digital utilizado será um módulo interno do dispositivo programável.

RHW8.2 O *hardware* deverá conter um circuito para atenuação do nível de tensão das baterias para que seja compatível com o módulo de AD utilizado.

RHW8.3 O *hardware* deverá conter um circuito de proteção para o ADC, limitando a tensão após a atenuação.

RHW8.4 (NF) A conexão da bateria do veículo com o *hardware* deverá ser feita por meio de conectores do tipo *microfit*.

RHW9 O *hardware* deverá permitir a aquisição digital do valor de tensão elétrica da estação de carregamento de veículos.

RHW9.1 O *hardware* deverá permitir que os sinais digitais da leitura da tensão da estação de carregamento seja lido por meio de *software* embarcado no dispositivo programável.

RHW9.1.1 (NF) O módulo de conversão analógico-digital utilizado será um módulo interno do dispositivo programável.

RHW9.2 O *hardware* deverá conter um circuito para atenuação do nível de tensão da estação de carregamento para que seja compatível com o módulo de AD utilizado.

RHW9.3 O *hardware* deverá conter um circuito de proteção para o ADC, limitando a tensão após a atenuação.

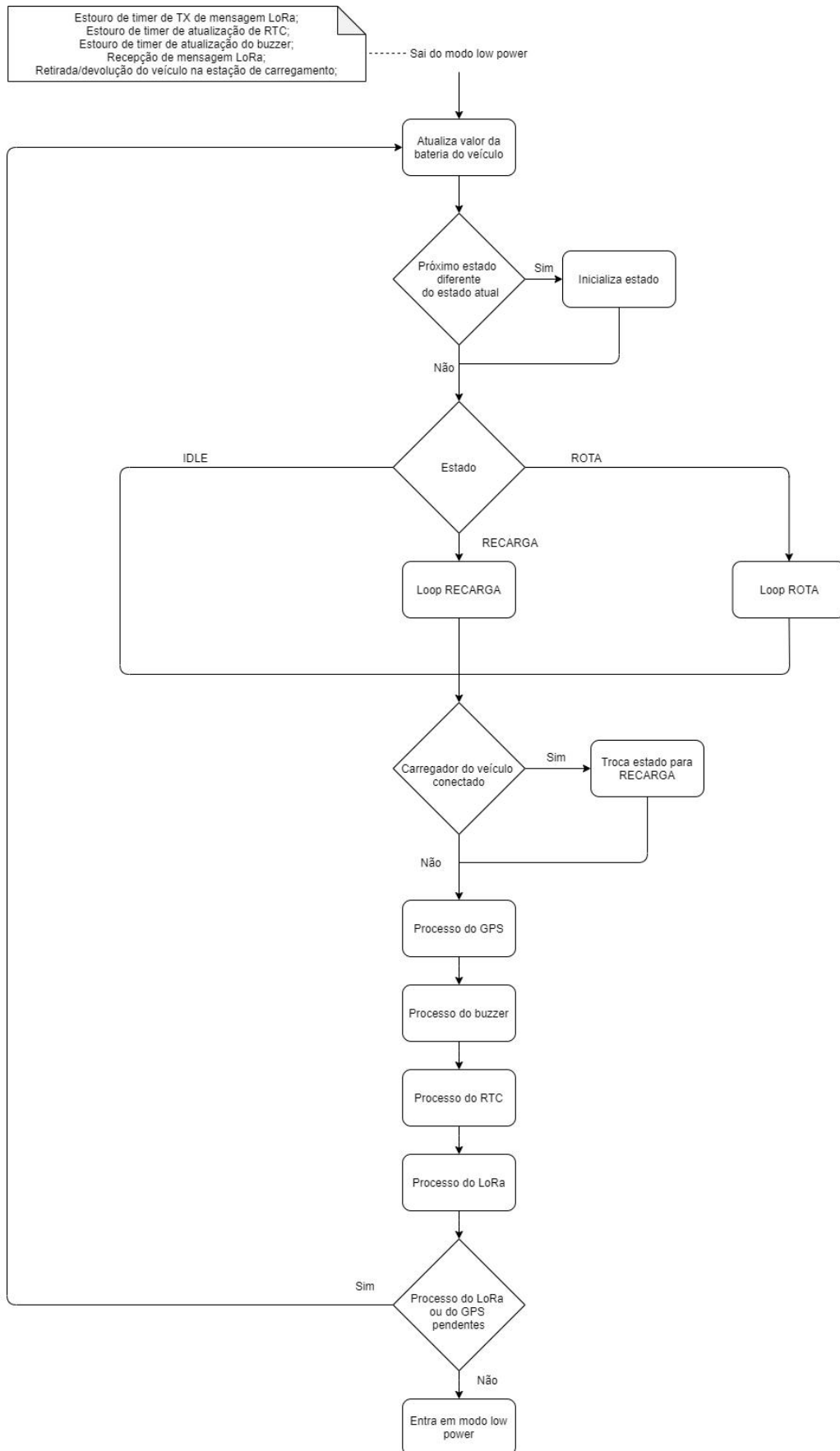
RHW9.4 (NF) A conexão da entrada da energia no veículo com o *hardware* deverá ser feita por meio de conectores do tipo *microfit*.

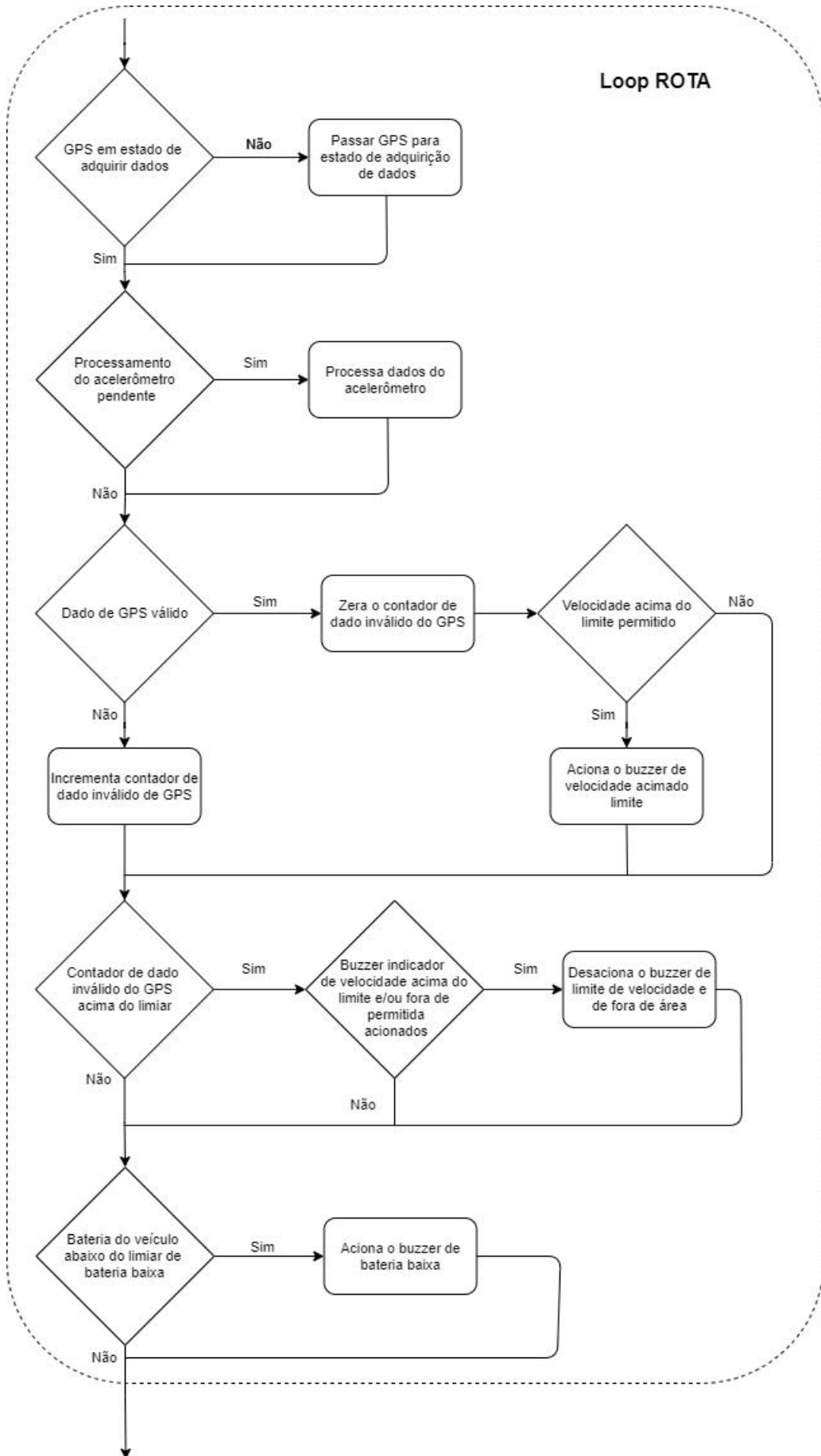
RHW10 (NF) O *hardware* deverá conter um *buzzer*.

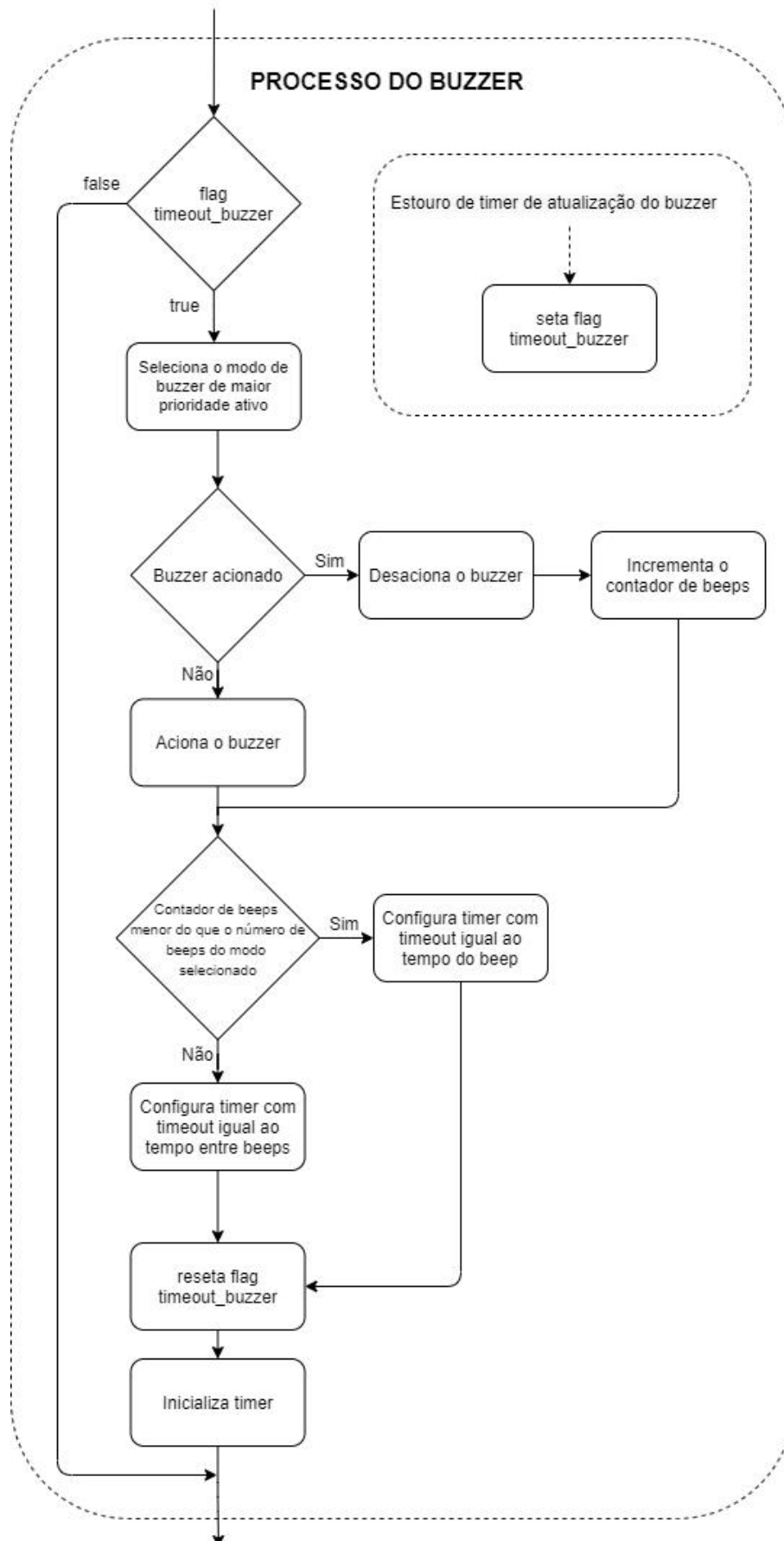
RHW10.1 O *hardware* deverá permitir que o *buzzer* seja acionado a partir de *software* embarcado no dispositivo programável.

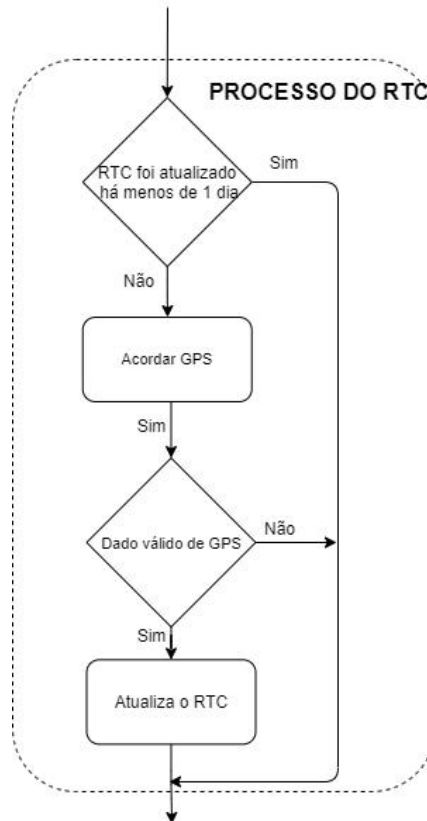
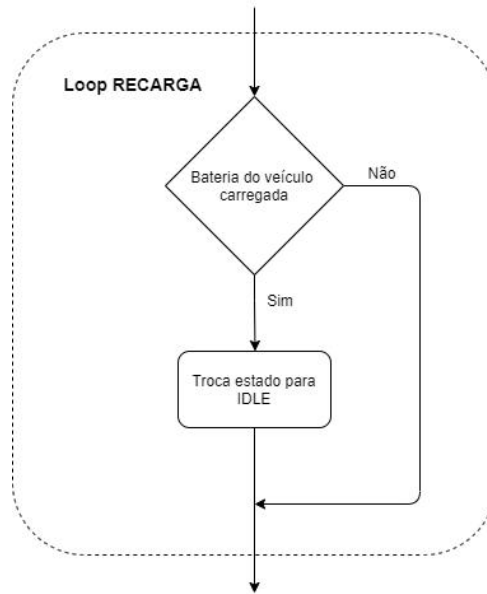
RHW10.2 (NF) O acionamento do *buzzer* deverá ser feito por meio de um transistor comandado pelo dispositivo programável.

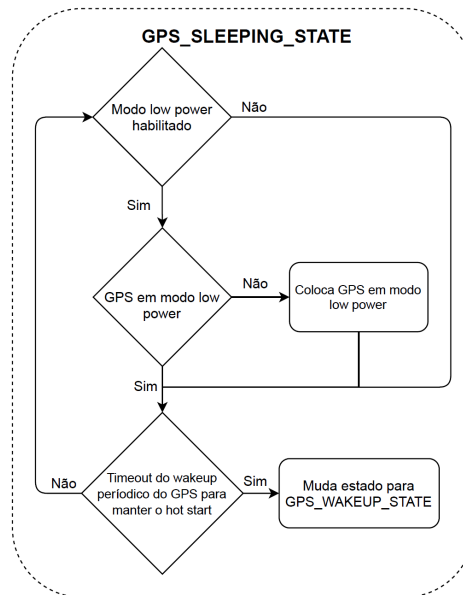
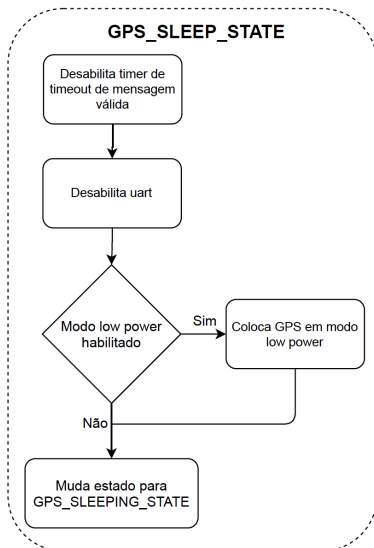
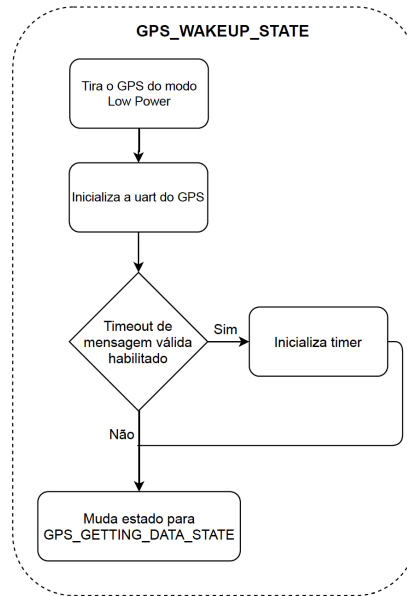
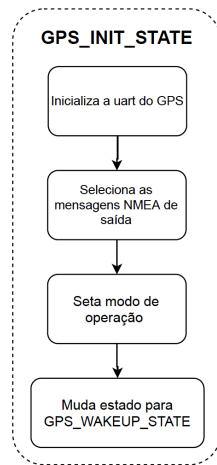
APÊNDICE B – DIAGRAMAS DE FUNCIONAMENTO DO *FIRMWARE* DO DISPOSITIVO DE MONITORAMENTO

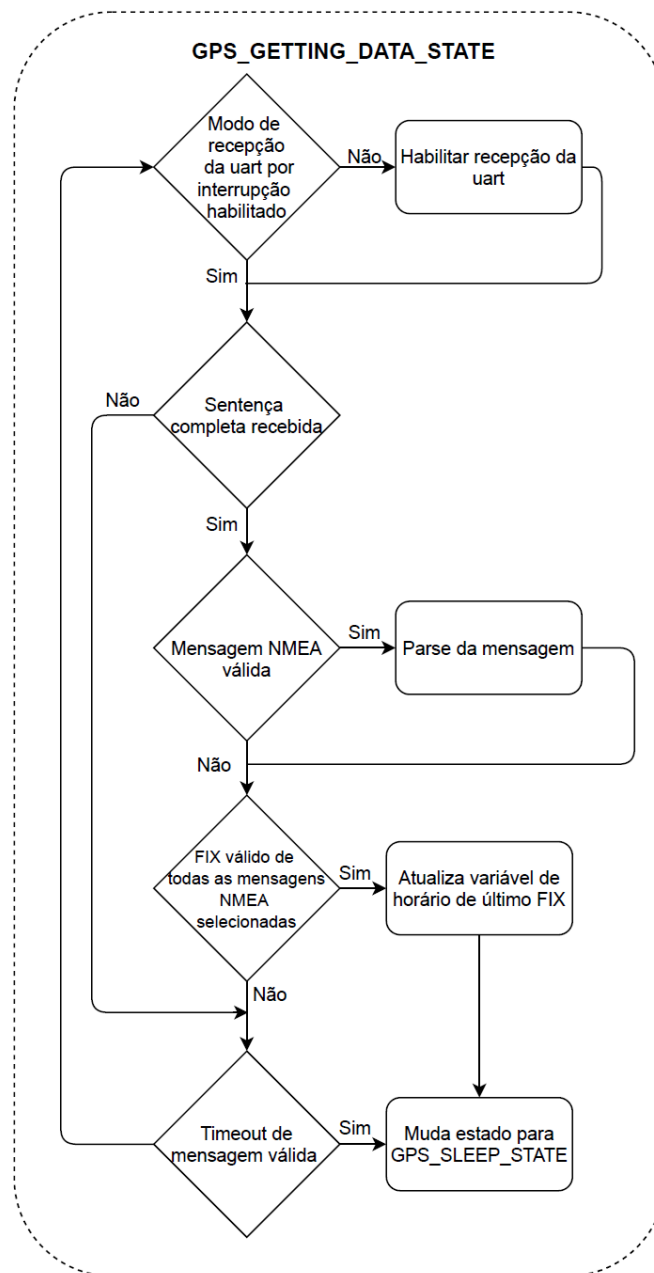




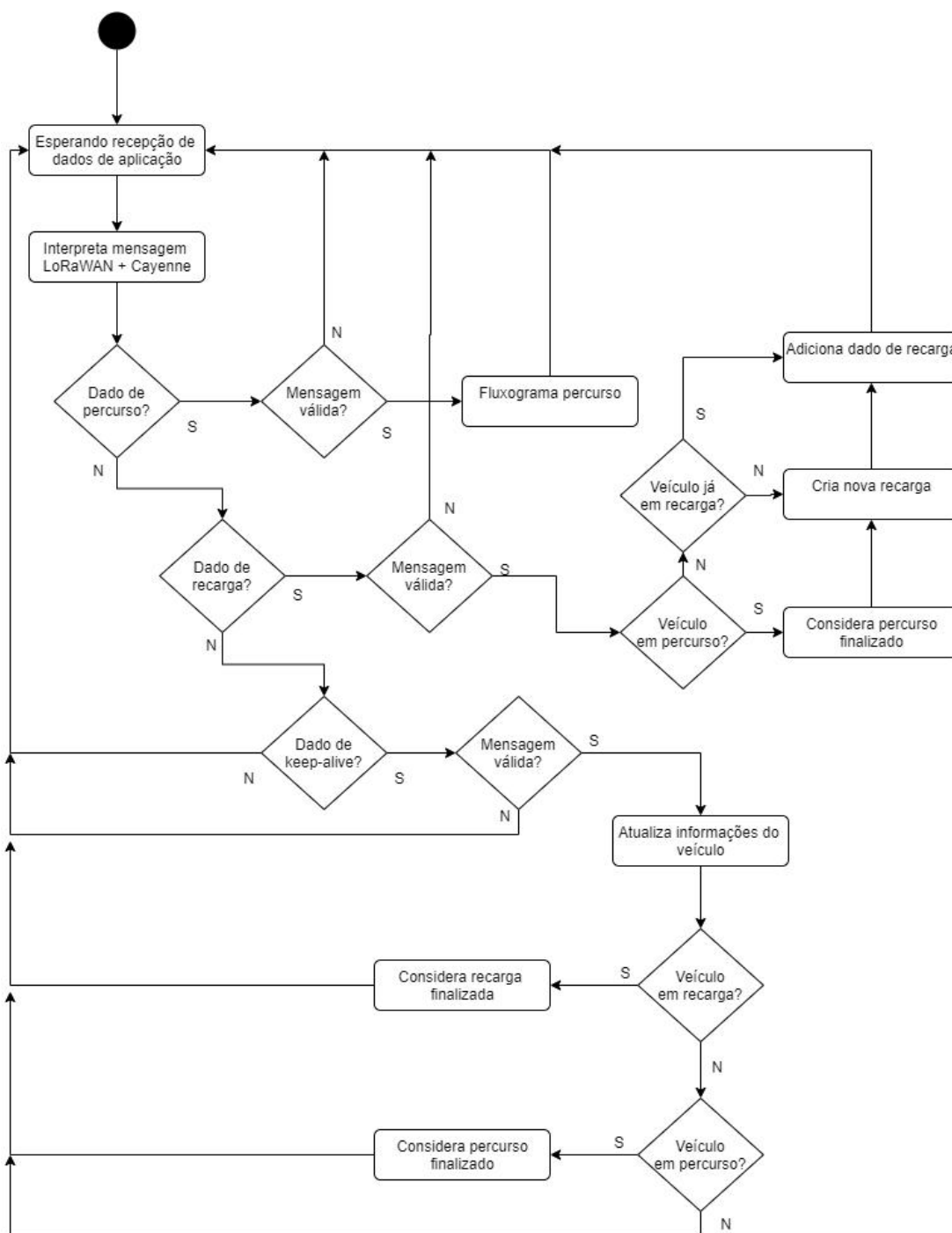






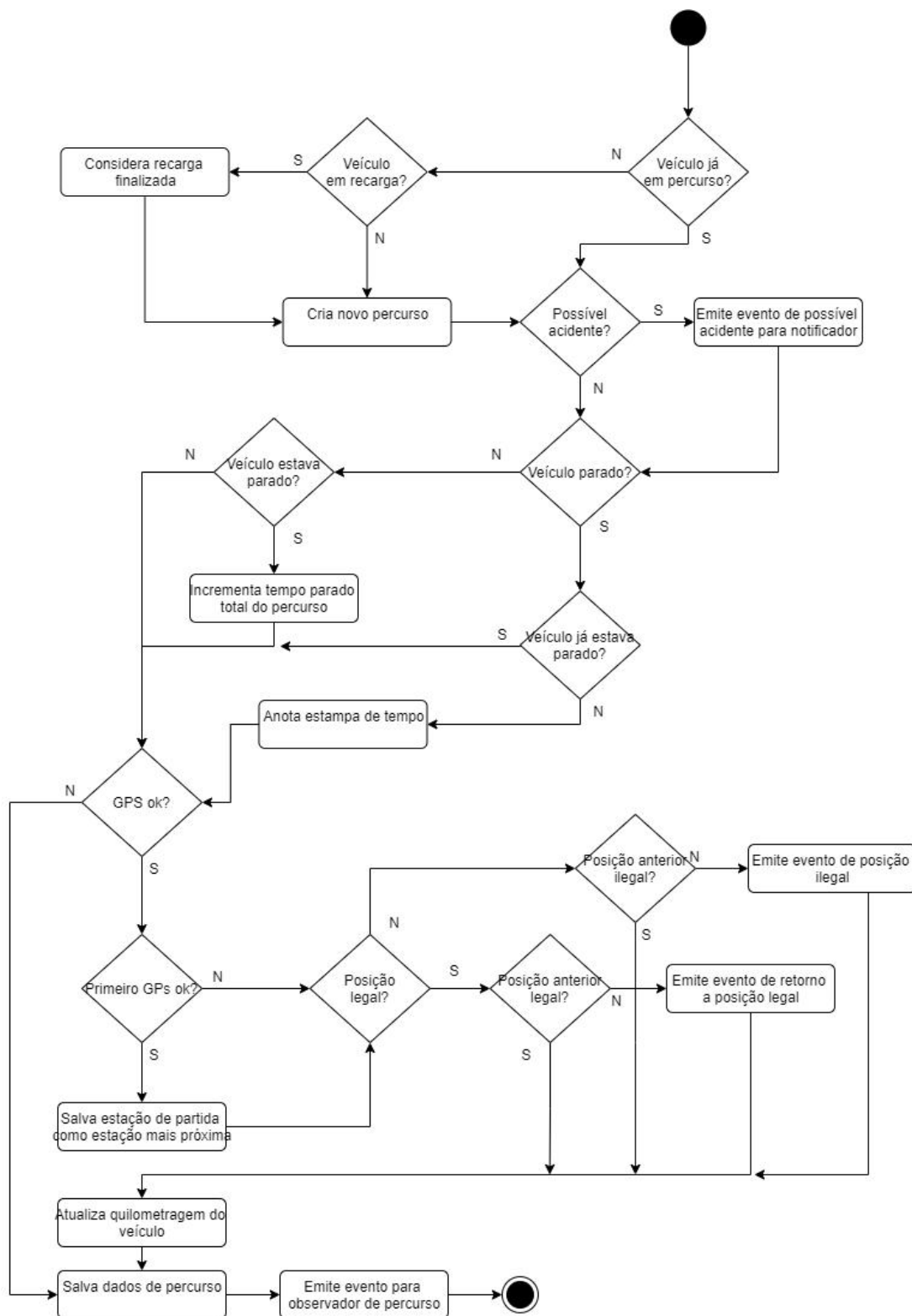


APÊNDICE C – DIAGRAMA DE ATIVIDADES DO BLOCO "LÓGICA DO SISTEMA" DO SERVIDOR DE APLICAÇÃO



Fonte: Autoria própria.

APÊNDICE D – DIAGRAMA DE ATIVIDADES PARA TRATAMENTO DE DADOS DE PERCURSO



Fonte: Autoria própria.

APÊNDICE E – DOCUMENTAÇÃO - API

Rotas

1. **Client**
Usuários dos patinetes. **Ainda não existe uma forma de buscar os clientes cadastrados no sistema da Emov e está sendo utilizado um cliente para testes.**
2. **Vehicle**
Patinetes e bicicletas com dispositivo de rastreamento.
3. **Route**
Percurso - conjunto de dados dos patinetes entre a retirada e a devolução na estação
4. **Recharge**
Recargas - conjunto de dados dos patinetes entre a colocação na estação para carregamento e a carga total ou retirada do patinete.
5. **General**
Informações gerais e estatísticas de uso dos patinetes.
6. **Config**
Configurações de parâmetros do sistema - tempo entre dados, área legal, etc
7. **Perimeter**
Conjuntos de pontos de GPS que definem um polígono dentro do qual é considerada a região legal de utilização dos patinetes.
8. **Station**
Estações de carregamento dos patinetes.
9. **Supervisor**
Responsáveis por receber as notificações de pânico e acidente.
10. **User**
Usuários da API
11. **Login**
Rota utilizada para fazer login na API

Com exceção da rota login, todas as demais devem conter o header **x-access-token, com um json-web-token de autenticação válido. Para obter o token, deve-se utilizar a rota /login com um usuário e senha válidos.**

1.Client (10.108.70.165/api/client)

GET / - requisição de todos os clientes cadastrados

POST / - criação de um novo cliente

Formato esperado: json, contendo:

name: nome do cliente (string), ex: "Gabriel"

code: L0 do cliente (string), ex: "L04394"

GET /:id - requisição de um cliente especificado pelo ID

DELETE /:id - exclusão de um cliente

PATCH /:id - alteração dos dados de um cliente

Formato esperado: json, contendo os seguintes campos, sendo nenhum deles obrigatório:

name: nome do cliente (string), ex: "Gabriel"

code: L0 do cliente (string), ex: "L04394"

usageNum: número de vezes que o cliente utilizou veículos do sistema(integer, unsigned), ex: 17

totalTime : tempo total de utilização pelo cliente, em segundos (integer, unsigned), ex: 7129

totalKilometers: distância total percorrida pelo cliente, em km (number, unsigned), ex: 1.29

GET /:id/route - requisição dos percursos de um cliente

Parâmetros query(opcionais):

from: não recebe percursos anteriores a esta data (ISO 8601), ex: 2020-02-18T12:52:30Z ou 2020-02-19

to: não recebe percursos posteriores a esta data (ISO 8601), ex: 2020-02-18T12:52:30Z ou 2020-02-19

GET /:id/route/data - requisição dos percursos de um cliente, incluindo os dados na resposta

Parâmetros query(opcionais):

from: não recebe percursos anteriores a esta data (ISO 8601), ex: 2020-02-18T12:52:30Z ou 2020-02-19

to: não recebe percursos posteriores a esta data (ISO 8601), ex: 2020-02-18T12:52:30Z ou 2020-02-19

2. Vehicle (10.108.70.165/api/vehicle)

GET / - requisição de todos os veículos cadastrados

POST / - criação de um novo veículo

Formato esperado: json, contendo:

deviceEUI: Device EUI do dispositivo de rastreamento instalado no veículo (string composta por 16 dígitos hexadecimais), ex: "0DB4A901020304AB"

code: código do veículo da Emov (integer), ex: 328
appKey: chave do dispositivo de rastreamento instalado no veículo (string formada por 32 dígitos hexadecimais), ex:
"bb73476e6d1043e99721ad83783fe419"
type: tipo do veículo da Emov (enum ("Scooter", "Bicycle")), ex: "Scooter"

GET /:id - requisição de um veículo especificado pelo ID

DELETE /:id - exclusão de um veículo

PATCH /:id - alteração dos dados de um veículo

Formato esperado: json, contendo os seguintes campos, sendo nenhum deles obrigatório:

deviceEUI: Device EUI do dispositivo de rastreamento instalado no veículo (string composta por 16 dígitos hexadecimais), ex: 0DB4A901020304AB
code: código do veículo da Emov (integer), ex: 328
appKey: chave do dispositivo de rastreamento instalado no veículo (string formada por 32 dígitos hexadecimais), ex:
bb73476e6d1043e99721ad83783fe419

GET /:id/route - requisição dos percursos de um veículo

Parâmetros query(opcionais):

from: não recebe percursos anteriores a esta data (ISO 8601), ex:
2020-02-18T12:52:30Z ou 2020-02-19

to: não recebe percursos posteriores a esta data (ISO 8601), ex:
2020-02-18T12:52:30Z ou 2020-02-19

GET /:id/route/data - requisição dos percursos de um veículo, incluindo os dados na resposta

Parâmetros query(opcionais):

from: não recebe percursos anteriores a esta data (ISO 8601), ex:
2020-02-18T12:52:30Z ou 2020-02-19

to: não recebe percursos posteriores a esta data (ISO 8601), ex:
2020-02-18T12:52:30Z ou 2020-02-19

GET /:id/recharge

Parâmetros query(opcionais):

from: não recebe recargas anteriores a esta data (ISO 8601), ex:
2020-02-18T12:52:30Z ou 2020-02-19

to: não recebe recargas posteriores a esta data (ISO 8601), ex:
2020-02-18T12:52:30Z ou 2020-02-19

GET /:id/recharge/data

3. Route (10.108.70.165/api/route)

GET / - Requisição de todos os percursos salvos

Parâmetros query(opcionais):

from: não recebe percursos anteriores a esta data (ISO 8601), ex:
2020-02-18T12:52:30Z ou 2020-02-19

to: não recebe percursos posteriores a esta data (ISO 8601), ex:
2020-02-18T12:52:30Z ou 2020-02-19

GET /data - Requisição de todos os percursos salvo, incluindo os dados de cada um

Parâmetros query(opcionais):

from: não recebe percursos anteriores a esta data (ISO 8601), ex:
2020-02-18T12:52:30Z ou 2020-02-19

to: não recebe percursos posteriores a esta data (ISO 8601), ex:
2020-02-18T12:52:30Z ou 2020-02-19

GET /:id - Requisição de um percurso especificado pelo ID

GET/:id/data - Requisição de um percurso especificado pelo id, com seus respectivos dados

4. Recharge (10.108.70.165/api/recharge)

GET / - Requisição de todas as recargas salvas

Parâmetros query(opcionais):

from: não recebe recargas anteriores a esta data (ISO 8601), ex:
2020-02-18T12:52:30Z ou 2020-02-19

to: não recebe recargas posteriores a esta data (ISO 8601), ex:
2020-02-18T12:52:30Z ou 2020-02-19

GET /data - Requisição de todas as recargas salvas, incluindo os dados de cada uma

Parâmetros query(opcionais):

from: não recebe recargas anteriores a esta data (ISO 8601), ex:
2020-02-18T12:52:30Z ou 2020-02-19

to: não recebe recargas posteriores a esta data (ISO 8601), ex:
2020-02-18T12:52:30Z ou 2020-02-19

GET /:id - Requisição de uma recarga especificada pelo ID

GET:/id/data - Requisição de uma recarga especificada pelo id, com seus respectivos dados

5. General (10.108.70.165/api/general)

GET / - Requisição de alguns dados gerais e estatísticas de uso do sistema. Por enquanto, devolve somente o tempo médio de utilização de cada percurso e a média de usos por dia, considerando ou não os sábados como dias úteis.

Parâmetros query(opcionais):

from: a partir de que dia considerar os dados para as médias de uso(ISO 8601),

ex:

2020-02-18T12:52:30Z ou 2020-02-19

to: até qual dia considerar os dados para as médias de uso(ISO 8601), ex:

2020-02-18T12:52:30Z ou 2020-02-19

6.Config (10.108.70.165/api/config)

GET / - Requisição dos parâmetros de configuração do servidor

PATCH / - Alteração dos parâmetros de configuração do servidor

Formato esperado: json, contendo os seguintes campos, sendo nenhum deles obrigatório:

routeTXPeriod - tempo entre cada pacote enviado pelos dispositivos quando em percurso, em segundos (integer, mínimo: 5, padrão: 5)

rechargeTXPeriod - tempo entre cada pacote enviado pelos dispositivos quando em recarga, em segundos (integer, mínimo: 5, padrão: 60)

keepAliveTXPeriod - tempo entre cada pacote enviado pelos dispositivos quando em idle, em segundos (integer, mínimo: 5, padrão: 300)

velocityLimit - limite de velocidade dos veículos, em km/h (number, mínimo: 15.0, máximo: 25.0)

accidentNotificationPeriod - tempo entre cada notificação de possível acidente enviada aos supervisores enquanto não é confirmado o recebimento, em segundos (integer, mínimo: 15, máximo: 120)

panicNotificationPeriod - tempo entre cada notificação de botão de pânico acionado enviada aos supervisores enquanto não é confirmado o recebimento, em segundos (integer, mínimo: 15, máximo: 120)

GPSErrorThreshold - módulo do erro do GPS a partir do qual os dados de geolocalização são considerados inválidos (number, unsigned, padrão: 3.0)

perimeterId - ID do perímetro cuja área interior é a área legal para utilização dos veículos (integer, unsigned, deve existir um perímetro com este id)

7. Perimeter (10.108.70.165/api/perimeter)

GET / - Requisição dos perímetros cadastrados no servidor

POST / - Criação de um novo perímetro no servidor

Formato esperado: json, contendo:

name: nome do perímetro (string), ex: "Estacionamento do Lactec - teste"
 coordinateArray: vetor de pontos de geolocalização que definem o perímetro/polígono (array de objetos). ex: [{latitude: -25.4464348, longitude: -49.2435019}, {latitude: -25.4470518, longitude: -49.2455011}, {latitude: -25.4538183, longitude: -49.2431014}]

DELETE / - Exclusão de um perímetro existente no servidor

8. Station (10.108.70.165/api/station)

GET / - Requisição das estações de carregamento existentes no banco de dados

9. Supervisor (10.108.70.165/api/supervisor)

GET / - Requisição dos supervisores salvos

POST / - Criação de um novo supervisor

Formato esperado: json, contendo:

name: nome do supervisor (string)
 email: e-mail do supervisor (string, domínio Lactec.org.br)
 receiveAccidentNotification: define se o supervisor criado deve receber notificações de possível acidente (boolean)
 receivePanicNotification : define se o supervisor criado deve receber notificações de "botão de pânico pressionado" (boolean)

GET /:id - Requisição de um supervisor específico

DELETE /:id- Exclusão de um supervisor

PATCH/:id - Alteração dos dados de um supervisor específico

Formato esperado: json, contendo os seguintes dados, sendo nenhum deles obrigatório:

name: nome do supervisor (string)
 email: e-mail do supervisor (string, domínio Lactec.org.br)
 receiveAccidentNotification: define se o supervisor criado deve receber notificações de possível acidente (boolean)
 receivePanicNotification : define se o supervisor criado deve receber notificações de "botão de pânico pressionado" (boolean)

10. User (10.108.70.165/api/user)

GET / - Requisição dos usuários (somente username) existentes no banco de dados

POST / - Criação de um novo usuário

Formato esperado: json, contendo os seguintes dados:

username: nome de usuário (string)

password: senha do usuário (string)

11. Login (10.108.70.165/api/login)

POST / - Requisita um token de autenticação para a API

Formato esperado: json, contendo:

username: nome de usuário (string)

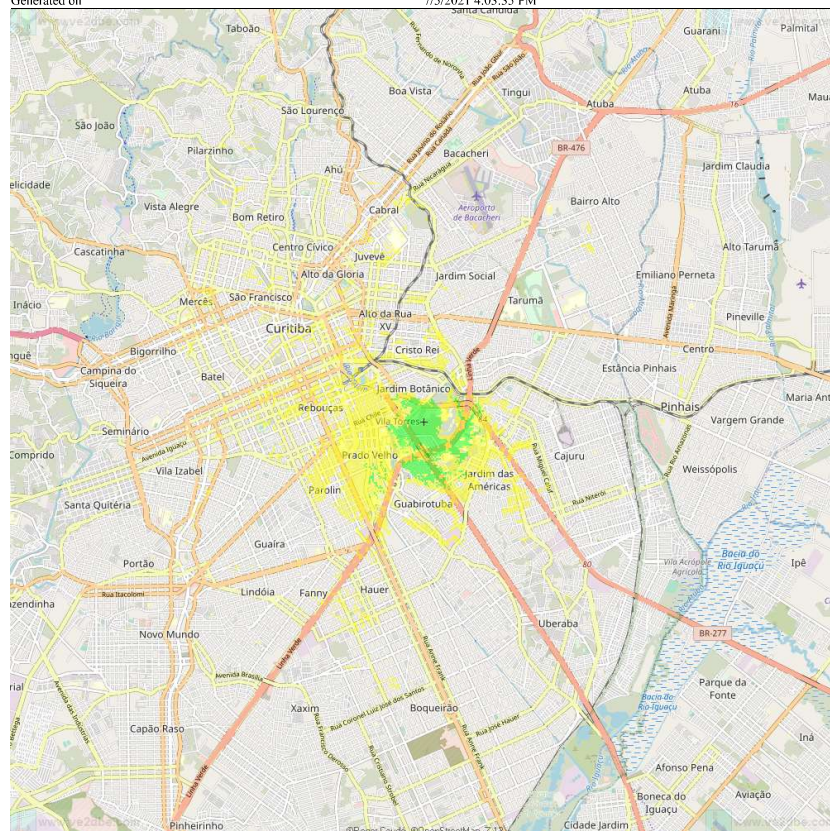
password: senha do usuário (string)

ANEXO

ANEXO A – SIMULAÇÃO DE COBERTURA DOS GATEWAYS *LORAWAN* POR MEIO DA FERRAMENTA VE2DBE

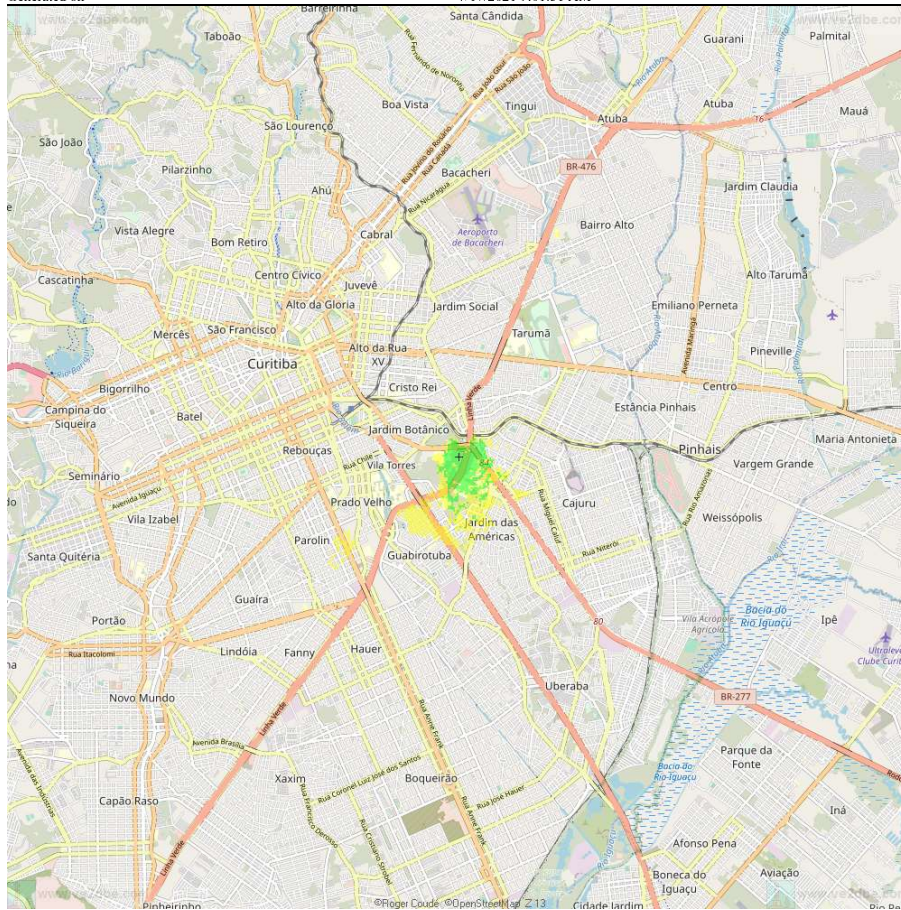
Radio Mobile Online Coverage report

Description	Gateway_SEDE
Frequency	915 MHz
Base Name	GatewaySEDE
Latitude	-25.44735680 °
Longitude	-49.24323820 °
Latitude	25° 26' 50.48"S
Longitude	049° 14' 35.66"W
QRA	GG54JN
UTM (WGS84)	22J E676645 S7184352
Elevation	936 m
Base Antenna Height	18.77 m
Base Antenna Gain	5.0 dBi
Base Antenna Type	omni
Base Antenna Azimuth	0 °
Base Antenna Tilt	0 °
Mobile Antenna Height	0.1 m
Mobile Antenna Gain	-3.8 dBi
Tx Power	0.10000 W
Tx Line Loss	3.0 dB
Rx Line Loss	0.5 dB
Rx Threshold	0.250 µV (-119.0 dBm)
Required Reliability	99%
Strong signal margin	13.0 dB
Weak signal field	21.2 dBµV/m
Strong signal field	34.2 dBµV/m
Weak signal covered area	11 km ²
Strong signal covered area	2 km ²
Weak signal population reached	39916 pop
Strong signal population reached	5807 pop
Landcover used	Yes
Two rays method used	Yes
User ID	geov
Radio coverage ID	RM74F9234CA12C_0
Generated on	7/5/2021 4:03:35 PM



Radio Mobile Online Coverage report

Description	Gateway LEME
Frequency	915 MHz
Base Name	LEME_2
Latitude	-25.44582153 °
Longitude	-49.23590247 °
Latitude	25° 26' 44.96"S
Longitude	049° 14' 09.25"W
QRA	GG54JN
UTM (WGS84)	22J E677385 S7184512
Elevation	920.1 m
Base Antenna Height	9.8 m
Base Antenna Gain	5.0 dBi
Base Antenna Type	omni
Base Antenna Azimuth	0 °
Base Antenna Tilt	0 °
Mobile Antenna Height	0.1 m
Mobile Antenna Gain	-3.8 dBi
Tx Power	0.10000 W
Tx Line Loss	3.0 dB
Rx Line Loss	0.5 dB
Rx Threshold	0.250 µV (-119.0 dBm)
Required Reliability	99%
Strong signal margin	13.0 dB
Weak signal field	21.2 dBµV/m
Strong signal field	34.2 dBµV/m
Weak signal covered area	2 km²
Strong signal covered area	1 km²
Weak signal population reached	8369 pop
Strong signal population reached	2901 pop
Landcover used	Yes
Two rays method used	Yes
User ID	geov
Radio coverage ID	RM74F9234CA12C_1
Generated on	7/17/2021 9:01:51 AM



Radio Mobile Online Coverage report

Description	Gateway CEHPAR
Frequency	915 MHz
Base Name	CEHPAR
Latitude	-25.45129028 °
Longitude	-49.23608645 °
Latitude	25° 27' 04.65"S
Longitude	049° 14' 09.91"W
QRA	GG54JN
UTM (WGS84)	22J E677358 S7183907
Elevation	909 m
Base Antenna Height	8.5 m
Base Antenna Gain	5.0 dBi
Base Antenna Type	omni
Base Antenna Azimuth	0 °
Base Antenna Tilt	0 °
Mobile Antenna Height	0.1 m
Mobile Antenna Gain	-3.8 dBi
Tx Power	0.10000 W
Tx Line Loss	3.0 dB
Rx Line Loss	0.5 dB
Rx Threshold	0.250 µV (-119.0 dBm)
Required Reliability	99%
Strong signal margin	13.0 dB
Weak signal field	21.2 dBµV/m
Strong signal field	34.2 dBµV/m
Weak signal covered area	2 km ²
Strong signal covered area	1 km ²
Weak signal population reached	8038 pop
Strong signal population reached	2605 pop
Landcover used	Yes
Two rays method used	Yes
User ID	geov
Radio coverage ID	RM74F9234CA12C_2
Generated on	7/17/2021 9:03:46 AM

