



Universidade Federal do Pará
Programa de Pós Graduação em Engenharia Elétrica
Instituto de Tecnologia

**PROPOSIÇÃO DE *HARDWARE* E *SOFTWARES*
PARA TRANSMISSÃO DE DADOS VIA LoRa E GSM:
ESTUDO DE CASO NO SMART CAMPUS DA UFPA**

Joel Alison Ribeiro Carvalho

DM: 05/03

Belém - Pará, 16 de março de 2023

Joel Alison Ribeiro Carvalho

**PROPOSIÇÃO DE *HARDWARE* E *SOFTWARES* PARA
TRANSMISSÃO DE DADOS VIA LoRa E GSM: ESTUDO DE
CASO NO SMART CAMPUS DA UFPA**

Dissertação de Mestrado apresentada ao Programa de Pós Graduação em Engenharia Elétrica, vinculado à Universidade Federal do Pará para obtenção do título de Mestre em Engenharia Elétrica.

Universidade Federal do Pará – UFPA

Programa de Pós Graduação em Engenharia Elétrica – PPGEE

Orientador: Prof. Dr. Fabrício José Brito Barros

Belém - Pará

16 de março de 2023

**Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD
Sistema de Bibliotecas da Universidade Federal do Pará
Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)**

C331p Carvalho, Joel.
PROPOSIÇÃO DE HARDWARE E SOFTWARES PARA
TRANSMISSÃO DE DADOS VIA LoRa E GSM : ESTUDO DE
CASO NO SMART CAMPUS DA UFPA / Joel Carvalho. — 2023.
93 f. : il. color.

Orientador(a): Prof. Dr. Fabrício Barros
Dissertação (Mestrado) - Universidade Federal do Pará,
Instituto de Tecnologia, Programa de Pós-Graduação em
Engenharia Elétrica, Belém, 2023.

1. Serialização. 2. Software. 3. Hardware,. 4. Transmissão
de Dados. I. Título.

CDD 384

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA ELÉTRICA
**“PROPOSICAO DE HARDWARE E SOFTWARES PARA
TRANSMISSÃO DE DADOS VIA LORA E GSM: ESTUDO DE CASO NO
SMART CAMPUS DA UFPA”**

AUTOR: JOEL ALISON RIBEIRO CARVALHO

DISSERTAÇÃO DE MESTRADO SUBMETIDA À BANCA EXAMINADORA APROVADA PELO COLEGIADO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA, SENDO JULGADA ADEQUADA PARA OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA ELÉTRICA NA ÁREA DE TELECOMUNICAÇÕES.

APROVADA EM: 16/03/2023

BANCA EXAMINADORA

Prof. Dr. Fabrício José Brito Barros
(Orientador – PPGEE/UFPA)

Prof.^a Dra. Jasmine Priscyla Leite de Araújo
(Avaliadora Interna – PPGEE/UFPA)

Prof.^a Dra. Maria Emília de Lima Tostes
(Avaliadora Interna – PPGEE/UFPA)

Prof. Dr. Fabrício de Souza Farias
(Avaliador Externo ao Programa – CAMPUS CAMETÁ/UFPA)

Prof. Dr. Diego Lisboa Cardoso
(Coordenador do PPGEE//ITEC/UFPA)

Dedico este trabalho em memória à minha mãezinha Lourdes Ferreira.

Agradecimentos

Quando eu era criança, se me perguntassem o que eu queria ser, eu sem sombra de dúvidas responderia que eu gostaria de ser um grande maestro. Nesta etapa da minha vida eu ainda não conhecia minha aptidão pela matemática e pelas ciências. Foi graças ao meu pai que, me mostrando o mundo da matemática desde a quinta série, que eu passei a sonhar com o dia em que me tornaria engenheiro.

Mas apesar disso, pensar em estar dentro de um ambiente acadêmico com grandes profissionais, colegas e professores sempre pareceu algo muito distante, talvez até impossível. Na minha família nunca conheci alguém que tivesse chegado tão longe. Estar neste lugar que me encontro, para mim é um sonho.

Em toda a minha vida, sempre fui cercado de pessoas ricas em conhecimento e espírito. Para mim, todas essas coisas só provam a existência de Deus, já que sempre iluminou o meu caminho. Eu, de forma alguma, poderia deixar de dizer que sou grato a todas as coisas que ele fez por mim.

Minha vida profissional sempre foi cercada de desafios, e assuntos novos todos os dias. Ser engenheiro é utilizar a ciência para transformar o mundo em um lugar melhor para as pessoas viverem; um mundo com mais conforto, segurança e por que não dizer felicidade? Hoje é um dia de grande realização para mim, pois vejo o quanto eu lutei para conquistar este espaço.

Meus agradecimentos vão em primeiro lugar a Deus por ter me dado forças para nunca desistir dos meus sonhos, por ter enxugado minhas lágrimas quando eu não aguentava mais, quando colocou em toda minha trajetória pessoas excepcionais que sempre me estenderam a mão, pessoas que me ensinaram um pouco mais sobre profissionalismo, felicidade, humildade, e solidariedade.

Agradeço a minha mãezinha por ter me amado e expressado esse amor todas as vezes que fazia minha comida favorita quando eu chegava da escola. Algumas vezes faltava comida mas você sempre dava um jeito de conseguir nosso alimento. Adorava suas histórias, seu abraço. Perto dos seus últimos anos, eu tive a honra de cuidar de você e demonstrar minha gratidão que é enorme. Você era pobre, mas rica de amor, de respeito, de humildade. Lembro-me muito bem no dia que defendi meu trabalho de conclusão de curso e falei a você que tinha conseguido concluir minha graduação em engenharia elétrica e você começou a chorar. Você foi uma pessoa feliz e a razão dessa felicidade eram as nossas conquistas. De todas as coisas que Deus já deu, você foi o meu maior presente. Obrigado por tudo. Eu te amo.

E ao meu pai, eu deixo a minha gratidão por ser a pessoa que juntamente com minha mãezinha sempre proveu o sustento da casa, que sempre orientou meus estudos, que me deu educação, que foi o meu melhor professor de matemática que já tive. Você é talvez a única

referência familiar que eu tive de alguém que passou por muitas dificuldades para conseguir a estabilidade financeira como servidor do Estado. Você e minha mãezinha foram cruciais para que esse sonho se tornasse realidade.

Agradeço à melhor professora de português que já tive, Helena Reis. Eu tive a honra de ser seu aluno no ensino fundamental na escola Helena Guilhon. E mesmo que faça tanto tempo, eu nunca me esqueci de você e o quanto você envolvia os alunos com suas histórias sobre literatura; dos dias que você me dava carona quando me encontrava andando a caminho da escola. Toda a base de língua portuguesa que tenho é graças a você.

Às minhas orientadoras de extensão, as primeiras que tive desde que entrei na graduação: professoras Dra. Maria Emília, e Dra. Jasmine Araújo, eu prezo meus agradecimentos. Vocês foram as primeiras a me mostrar um mundo de possibilidades da engenharia que não vemos na graduação. Me ajudaram em momentos que precisei, sobretudo na pandemia, período que estudei Inteligência Computacional na Pós Graduação como aluno especial. Neste período, professora Jasmine, você me forneceu auxílio financeiro para internet móvel para que eu não perdesse suas aulas. Aquilo me marcou muito, pois era o período que eu estava cuidando da minha mãezinha e estava passando por algumas dificuldades financeiras.

Ao meu orientador, professor Dr. Fabrício Barros, por ter me dado a oportunidade de mostrar o meu trabalho, desde o meu ingresso na iniciação científica. Você sempre utilizou palavras de incentivo em momentos importantíssimos da minha pesquisa; momentos às vezes que eu duvidava se eu teria capacidades de progredir com a minha pesquisa. Essas palavras fizeram toda a diferença. Você me ajudou de diversas formas desde financeiramente, profissionalmente e me ensinou juntamente com as professoras Emília e Jasmine como um profissional de sucesso deve agir; você foi compreensivo no momento mais difícil da minha vida. Obrigado pelas oportunidades, pela orientação, pelo carinho, por tudo.

Ao meu amigo engenheiro Me. Filipe Cavalcanti, eu agradeço por ter me ensinado tanto; em áreas que vão desde programação, escrita de trabalhos científicos, orientações no TCC e principalmente música. Você além de me ensinar sobre diversos assuntos, me estendeu a mão em momentos da minha vida que eu não tinha proventos pra minha alimentação. Você nunca mediu esforços para me ajudar quando mais precisei. Eu agradeço de coração pela amizade e carinho.

Também agradeço ao Artur e seus pais: Seu Jorge e Dona Irene por terem me cedido um apartamento estruturado, temporariamente, que permitiu que eu desenvolvesse grande parte da minha pesquisa relacionada ao Mestrado

E quanto às instituições, gostaria de agradecer primeiramente à Universidade Federal do Pará (UFPA) por permitir a oportunidade de estudar em um ensino superior, ao PPGEE pela oportunidade de estar em uma pós graduação, ao Laboratório de Computação e Telecomunicações (LCT) por ter me fornecido estrutura para que minha pesquisa fosse desenvolvida, à Norte Energia S.A. por financiar o projeto Sistema Inteligente Multimodal da Amazônia (SIMA),

pois possibilitou o desenvolvimento desta pesquisa e a CAPES pela bolsa de mestrado que possibilitou a minha permanência na pós graduação.

Aos meus colegas de pesquisa, Caio Mateus, Artur Machado, Hugo Oliveira e Alex Sanches, gostaria de dizer que é uma honra trabalhar com todos vocês e desenvolver as pesquisas do LCT.

Agradeço à Aline, minha melhor amiga, pois desde que nos conhecemos você sempre foi uma pessoa amiga que sempre esteve por perto para me ajudar nos momentos tristes, sobretudo quando perdi minha mãezinha, período que eu não tinha forças nem para levantar da cama. Sua amizade é muito especial para mim. Obrigado.

À minha namorada Lucélia Cruz, por ter sido tão companheira desde o dia que nos conhecemos, por ter dividido comigo diversos momentos de felicidade, por ter sido meu ombro amigo em momentos que eu pensava em desistir de tudo. Essa conquista também é graças a você. Obrigado por ter entrado na minha vida.

E como eu poderia deixar de agradecer ao meu grande amigo Maxson Almeida. Você foi um irmão que eu consegui no período da pandemia. Você permitiu que eu fizesse grande parte da minha pesquisa do trabalho de conclusão de curso em sua residência. Você é uma das pessoas que eu mais confio e sempre esteve lá para me prestigiar em momentos de conquista. Você também foi de grande ajuda para que este momento fosse possível.

Aos demais amigos, prezo meus agradecimentos ao Brenda Silva, Hállan Rodrigues, Igor Cunha, Josias Cordeiro e Vanessa Martins. Todos vocês me proporcionaram diversos momentos de lazer e também me ajudaram em momentos difíceis.

Eu finalizo meus agradimentos, retomando as duas figuras mais importantes da minha vida: minha mãezinha e meu pai. Se eu fosse defini-los em uma palavra cada, seria: amor e educação, pois o amor nos dá forças, mas a educação transforma. Meus mais singelos agradecimentos a todos por fazer parte da realização de mais um sonho.

*O Amor
tudo sofre,
tudo crê,
tudo espera,
tudo suporta
1ª Coríntios 13:7.*

Resumo

A sociedade vem passando por grandes transformações na era da indústria 5.0, com maior interação entre homem e máquina além do aumento da conectividade. Fatores esses que tem proporcionado o crescimento da internet das coisas, sobretudo em cidades, onde são construídas as *Smart Cities* e os *Smart Campus*. Neste sentido, o desenvolvimento de hardwares possibilita maior controle sobre a integração dos dispositivos, já que atende as necessidades do projeto além de tornar o projeto escalável com o incremento de novos sensores à medida que o projeto se desenvolve. Para isto, neste trabalho são desenvolvidos *hardwares* e *softwares* para transmissão de dados via LoRa e GSM para uma plataforma de *middleware*, e para validar esta proposta foram realizadas transmissões de dados, considerando as aplicações do do *smart campus* da UFPA: Ônibus Circular, Ônibus Intermunicipal, Sistema Fotovoltáico e Sistema de Baterias. Os resultados mostraram que todos os equipamentos se comunicaram com a Dojot, já que todas as variáveis de interesse de cada aplicação puderam ser observadas no servidor.

Palavras-chaves: Serialização, Software, Hardware, Transmissão de Dados.

Abstract

Society has been undergoing major transformations in the industry 5.0 era, with greater interaction between man and machine in addition to increased connectivity. These factors have provided the growth of the internet of things, especially in cities, where Smart Cities and Smart Campus. In this sense, hardware development enables greater control over the integration of devices, as it meets the needs of the project in addition to to make the project scalable by adding new sensors as the project progresses develops. For this, in this work are developed hardware and software for transmission of data via LoRa and GSM to a middleware platform, and to validate this proposal data transmissions were carried out, considering the applications of the smart campus of UFPA: Circular Bus, Intercity Bus, Photovoltaic System and Battery System. You results showed that all equipment communicated with Dojot, as all Variables of interest for each application could be observed on the server.

Keywords: Serialization, Software, Hardware, Data Transmission.

Lista de ilustrações

Figura 1 – Estrutura do Projeto SIMA	2
Figura 2 – Estrutura geral de transmissão do <i>smart campus</i> da UFPA	4
Figura 3 – Arquitetura do sistema de rastreamento de veículos	7
Figura 4 – Cenário de medição - Campus Cameté	8
Figura 5 – Design do invólucro contendo o transmissor LoRa	9
Figura 6 – Unidade de Aquisição de Corrente e Tensão	11
Figura 7 – Total de indicadores encontrados por cada categoria	14
Figura 8 – Representação da técnica CSS mostrando o <i>down-chirp</i> e <i>up-chirp</i>	16
Figura 9 – Largura de banda na tecnologia LoRa	16
Figura 10 – Frequências LoRa utilizadas no Mundo	17
Figura 11 – Relação do parâmetro de cada SF com: tempo no ar, alcance e taxa de bits	18
Figura 12 – Representação de um pacote LoRa	19
Figura 13 – Arquitetura LoRaWAN	20
Figura 14 – Shields LoRa (a) LoRa Dragino Shield; (b) LoRa GPS Shield	21
Figura 15 – RAK7249 e suas 4 antenas	22
Figura 16 – Shield SIM900	24
Figura 17 – LEDs indicadores de status e botão que aciona o cartão SIM	24
Figura 18 – Estrutura do MQTT	25
Figura 19 – Quadro MODBUS de requisição	27
Figura 20 – Esquema de transmissão <i>slave-master</i> MODBUS (a) Variáveis Armazenadas nos Registradores do <i>slave</i> ; (b) Variáveis transmitidas para o <i>master</i>	28
Figura 21 – Unidade de Aquisição de Corrente e Tensão - Contínua	29
Figura 22 – Unidade de Aquisição de Corrente e Tensão - Trifásica	29
Figura 23 – RS485	29
Figura 24 – Configuração de leitura e escrita no RS485	30
Figura 25 – Ilustração de como funciona o GPS	31
Figura 26 – GPS Shield DUINOPEAK	31
Figura 27 – Flowbroker da Dojot com a criação do bloco Decifragem LoRa	33
Figura 28 – Ilustração de um dispositivo com suas variáveis sendo monitoradas individualmente	34
Figura 29 – Ilustração de um dispositivo com suas variáveis sendo monitoradas individualmente	34
Figura 30 – Instalação de gateways no <i>smart campus</i> da UFPA (a) Mirante; (b) Prédio do LEEB; (c) Prédio do CEAMAZON	35
Figura 31 – Serialização de Dados Transmitidos	36
Figura 32 – Ônibus circular elétrico	37

Figura 33 – Arquitetura de transmissão do Ônibus Circular	37
Figura 34 – Fluxograma do end-node da aplicação <i>Ônibus Circular</i>	38
Figura 35 – Ônibus elétrico intermunicipal	39
Figura 36 – Arquitetura de transmissão do Ônibus Intermunicipal	39
Figura 37 – Estruturação dos Dados em JSON	40
Figura 38 – Fluxograma de transmissão do end-node da aplicação Ônibus Intermunicipal	40
Figura 39 – Método de ativação automática do cartão SIM	41
Figura 40 – Esquema de Conexão do RS485 com a UACT e o Arduino	43
Figura 41 – Sistema Fotovoltáico	44
Figura 42 – Arquitetura de Transmissão do Sistema Fotovoltáico	45
Figura 43 – Fluxograma do transmissor do Sistema Fotovoltáico	48
Figura 44 – Eletrocentro com o Sistema de Baterias em destaque	49
Figura 45 – Arquitetura de Transmissão do Sistema de Baterias	49
Figura 46 – Fluxogramad do end-node do Sistema de Baterias	52
Figura 47 – Conguração do End-Node LoRa	53
Figura 48 – Conguração MQTT do transmissor GSM	53
Figura 49 – Configuração de acesso a Dojot	54
Figura 50 – (a) <i>Flowbroker</i> ; (b) Processamento de dados LoRa na Dojot; (c) Percurso dos dados até os dispositivos Virtuais na Dojot	55
Figura 51 – Instalação de gateways no <i>smart campus</i> da UFPA (a) Mirante; (b) Prédio do LEEB; (c) Prédio do CEAMAZON	58
Figura 52 – Resultados do teste de validação da serialização tendo como exemplo o Sistema Fotovoltáico	59
Figura 53 – Dispositivo Transmissor do Ônibus Circular	60
Figura 54 – Instalação do End-node do transmissor do Ônibus Circular	61
Figura 55 – Resultados da Transmissão do Ônibus Circular	61
Figura 56 – Geolocalização do Ônibus Circular visto no site do LCT	62
Figura 57 – Protótipo do transmissor do Ônibus Intermunicipal	63
Figura 58 – Transmissor do Ônibus Intermunicipal aprimorado	63
Figura 59 – Testes realizados em campo com o transmissor do <i>Intermunicipal</i> . (a) Sem a reconexão; (b) Com a reconexão	64
Figura 60 – Distâncias entre pontos os pontos de GPS capturados durante a rota	65
Figura 61 – Transmissor do Sistema Fotovoltáico	65
Figura 62 – End-node do Sistema Fotovoltáico	66
Figura 63 – Resultado do Teste realizado em laboratório visualizados na Dojot	66
Figura 64 – Transmissor do Sistema Fotovoltáico instalado ao lado do UACT 3F	67
Figura 65 – Quadro de energia que disponibiliza as informações ao UACT 3F	67
Figura 66 – Quadro de energia que disponibiliza as informações ao UACT 3F	68
Figura 67 – Instalação do Transmissor do Sistema de Baterias	69

Figura 68 – Resultado do teste em laboratório referente ao **Sistema de Baterias** 69

Lista de tabelas

Tabela 1 – Redes LPWANs e suas configurações principais	15
Tabela 2 – Diferentes configurações de transmissão no canal LoRa para AUS915	19
Tabela 3 – Serialização do Payload para o Ônibus Circular	38
Tabela 4 – Esquema de Conexão do RS485 com a UACT e o Arduino	42
Tabela 5 – Serialização do Payload para o Sistema Fotovoltáico	46
Tabela 6 – Serialização do Payload para o Sistema de Baterias	50
Tabela 7 – Síntese Sobre as 4 aplicações	56
Tabela 8 – Tempo no ar das 3 aplicações sem e com serialização	70

Lista de abreviaturas e siglas

CEAMAZON:	Centro de Excelência em Eficiência Energética
GSM:	<i>Global System for Mobile Communications</i>
H2M:	Pessoas para Máquinas
ID_1 :	Identificador do dispositivo UACT
ID_2 :	Identificador da aplicação
ISM:	<i>Industrial Scientific and Medical</i>
LABEX:	Laboratório de Programação Extrema
LASSE:	Laboratório de Sensores e Sistemas Embarcados
LCT:	Laboratório de Computação e Telecomunicações
LoRa:	<i>Long-Range</i>
LPWAN:	<i>Low Power Wide Area Network</i>
M2M:	Máquinas para Máquinas;
P2P:	Pessoas para Pessoas
SIMA:	Sistema Inteligente Multimodal da Amazônia
UACT 3F:	Unidade de Aquisição de Corrente e Tensão Trifásica
UACT CC:	Unidade de Aquisição de Corrente e Tensão Contínua
UFPA:	Universidade Federal do Pará

Sumário

1	Introdução	1
1.1	Introdução Geral	1
1.2	Contexto da Pesquisa	1
1.3	Arquitetura de Transmissão	3
1.4	Justificativa	4
1.5	Objetivos	5
1.5.1	Objetivo Geral	5
1.5.2	Objetivos Específicos	5
1.6	Estrutura do Trabalho	5
2	Revisão de Literatura	7
2.1	Conteúdo do Capítulo	7
2.2	Conclusão do Capítulo	11
3	Fundamentação Teórica	12
3.1	Conteúdo do Capítulo	12
3.2	A Indústria 5.0	12
3.3	A Internet das Coisas	12
3.4	<i>Smart Cities</i> e <i>Smart Campus</i>	13
3.5	Protocolos e Hardwares de Comunicação	14
3.5.1	LPWAN	14
3.5.2	A Tecnologia LoRa	15
3.5.2.1	Largura de Banda	16
3.5.2.2	Taxa de Código	18
3.5.2.3	Fator de Espalhamento - SF	18
3.5.2.4	Indicador de Intensidade do Sinal Recebido	19
3.5.2.5	Tempo de Transmissão do Pacote	19
3.5.3	O Protocolo LoRaWAN	20
3.5.3.1	Criptografia LoRaWAN	20
3.5.4	O End-node LoRa	21
3.5.4.1	Visão Geral	21
3.5.4.2	Biblioteca	21
3.5.5	O Gateway Utilizado para Comunicação LoRa	21
3.5.6	Sistema Global para Comunicações Móveis	22
3.5.7	O Hardware GSM/GPRS	23
3.5.7.1	Visão Geral	23
3.5.7.2	Características do <i>hardware</i>	23
3.5.7.3	Biblioteca	25

3.5.8	O Protocolo MQTT	25
3.5.9	O Protocolo MODBUS	26
3.5.9.1	O Formato do quadro MODBUS de Requisição	26
3.5.9.2	O Formato do quadro MODBUS de Resposta	27
3.5.9.3	O Hardware <i>Slave</i> MODBUS	28
3.5.9.4	Interface RS485 Como Meio de Comunicação Entre Dispositivos MODBUS	29
3.5.10	O Sistema de Posicionamento Global	30
3.5.10.1	O Hardware GPS	31
3.6	Tipos de Variáveis	31
3.7	A DOJOT	32
3.7.1	O <i>Flowbroker</i>	33
3.8	Conclusão do Capítulo	34
4	Metodologia	35
4.1	Conteúdo do Capítulo	35
4.2	Instalação dos Gateways	35
4.3	Proposta para Serialização dos Dados	35
4.4	Aplicações Móveis	36
4.4.1	Ônibus Circular	36
4.4.1.1	Arquitetura de Transmissão do Ônibus Circular	36
4.4.2	Serialização do Payload para o Ônibus Circular	37
4.4.2.1	Fluxograma do transmissor Ônibus Circular	37
4.4.3	Ônibus Intermunicipal	38
4.4.3.1	Arquitetura de transmissão do Ônibus Intermunicipal	39
4.4.4	Estruturação dos Dados em JSON	40
4.4.4.1	Instalação do CFP 100	40
4.4.4.2	Fluxograma do transmissor Ônibus Intermunicipal	40
4.4.4.3	Metodologia de Ativação Automática do Cartão SIM e Reconexão da rede GSM	41
4.5	Aplicações Estáticas	42
4.5.1	Esquema de Conexão do RS485 com a UACT e o Arduino	42
4.5.2	Criação da biblioteca MODBUS em C/C++ para Recebimento dos Dados	42
4.5.3	Sistema Fotovoltáico	44
4.5.3.1	Arquitetura de Transmissão do Sistema Fotovoltáico	44
4.5.3.2	Serialização do Payload para o Sistema Fotovoltáico e Algoritmo do End-Node	44
4.5.3.3	Método de Agragação de Dados	47
4.5.4	Sistema de Baterias	48
4.5.4.1	Arquitetura de Transmissão do Sistema de Baterias	49

4.5.4.2	Serialização do Payload para o Sistema de Baterias e Algoritmo do End-Node	50
4.6	Configuração dos End-Nodes	52
4.7	Configuração dos Gateways	53
4.8	Configuração da Dojot	54
4.9	Síntese Sobre as 4 Aplicações	55
4.10	Conclusão do capítulo	57
5	Resultados	58
5.1	Conteúdo do Capítulo	58
5.2	Análise de Cobertura dos Gateways	58
5.3	Teste de Validação da Serialização	59
5.4	Ônibus Circular	59
5.4.1	Montagem do Transmissor	59
5.4.2	Instalação do Transmissor	60
5.4.3	Resultados da Transmissão do Ônibus Circular	61
5.5	Ônibus Intermunicipal	62
5.5.1	Aprimoramento do Transmissor do Ônibus Intermunicipal	62
5.5.2	Teste em Campo	63
5.6	Sistema Fotovoltáico	65
5.6.1	Setup do Transmissor	65
5.6.2	Testes em Laboratório	65
5.6.3	Instalação do Transmissor e Teste em Campo	67
5.7	Sistema de Baterias	68
5.7.1	Montagem e Instalação do Transmissor	68
5.7.2	Testes Realizados no Laboratório	68
5.8	Análise do <i>time-on-air</i> para as Aplicações LoRa	69
5.9	Conclusão do Capítulo	70
	Conclusão	71
	Trabalhos Futuros	71
	Referências	73

1 Introdução

1.1 Introdução Geral

A Internet das Coisas (IoT) é uma tecnologia que tem transformado a forma como as empresas e os consumidores utilizam a tecnologia. De acordo com Borgia (2014), a IoT envolve objetos cotidianos que se conectam à internet e se comunicam entre si, tornando possível uma série de novas possibilidades para a automação e o monitoramento na transmissão de dados.

Para que a IoT funcione adequadamente, é essencial desenvolver hardwares e softwares de transmissão de dados eficientes, confiáveis e seguros. E para o desenvolvimento de sistemas IoT, podem existir diferentes hardwares em que cada um deles tenha suporte para uma tecnologia distinta que atenda os requisitos de implementação que envolvem baixa latência, baixo custo, segurança e escalabilidade. Os dispositivos IoT, atualmente, mais utilizados atualmente possuem suporte para as tecnologias Wi-Fi, Bluetooth, Zigbee, LoRaWAN, muito utilizada no paradigma da Internet das Coisas (ATZORI; IERA; MORABITO, 2010; JIN et al., 2017).

Em resumo, o desenvolvimento de hardwares e softwares de transmissão de dados para a IoT é um campo de pesquisa e desenvolvimento em constante evolução, fundamental para a criação de soluções inovadoras e eficientes para a automatização de processos e a melhoria da qualidade de vida das pessoas. Além disso, o desenvolvimento de dispositivos transmissores com softwares e bibliotecas prontas, possibilita que outros pesquisadores e até empresas utilizem estas informações para o desenvolvimento de seus projetos IoT.

Um dos grandes desafios dos projetistas IoT é encontrar soluções que se adequem a sua problemática. Os dispositivos IoT que são vendidos como soluções finais, como os da ALEXA tem a finalidade de atender uma demanda local onde eles se conectam por Wi-Fi, em que o usuário pode controlar diversos de seus eletrodomésticos apenas com o comando de voz (DUNN, 2016). Por outro lado, hardwares compatíveis com LoRa Sigfox ou GSM têm como finalidade principal o desenvolvimento de projetos que se adequem a realidade da região onde deseja-se instalar o sistema IoT por não se limitarem a conexão Wi-Fi ou ethernet como é o caso dos dispositivos ALEXA (DEVALAL; KARTHIKEYAN, 2018). Dessa forma, estes dispositivos podem ser configurados e integrados com sensores e microcontroladores para o desenvolvimento de soluções mais robustas e econômicas.

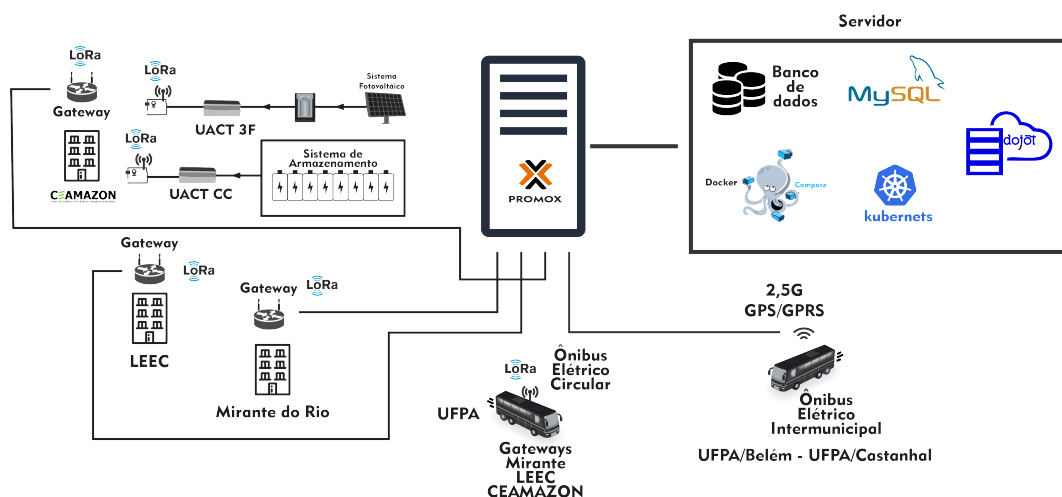
1.2 Contexto da Pesquisa

Em 2019, o Pará iniciou seu primeiro sistema inteligente multimodal interconectado. Chamado de Projeto de Pesquisa e Desenvolvimento (P& D) do Sistema Inteligente Multimodal

da Amazônia (SIMA). Tal projeto tem a atuação de diversos centros de pesquisa da UFPA, tais como: Centro de Excelência em Eficiência Energética (CEAMAZON), Laboratório de Sensores e Sistemas Embarcados (LASSE), Laboratório de Computação e Telecomunicações (LCT), Laboratório de Programação Extrema (LABEX), Engenharia Naval, entre outros.

O projeto SIMA é uma parceria da UFPA com a Norte Energia S.A (financiadora do projeto) para tornar a universidade um *campus inteligente* contendo veículos movidos a energia elétrica (ônibus e barco elétricos), bem como os sistemas de suporte a esses veículos (eletropostos), além dos sistemas de geração e armazenamento de energia (sistema fotovoltaico e sistema de baterias). Cada uma dessas aplicações são monitoradas em uma plataforma *middleware* brasileira conhecida como Dojot, cuja transmissão de dados é um dos focos deste trabalho. A Dojot, dessa forma, tem como principal objetivo receber, armazenar e enviar dados quando forem solicitados para suas respectivas aplicações (LOBATO et al., 2021). A estrutura resumida do projeto SIMA contendo apenas as aplicações descritas neste trabalho, está ilustrada na Fig. 1.

Figura 1 – Estrutura do Projeto SIMA



Fonte: Adaptado de (LOBATO, 2022)

Foram instalado 3 gateways LoRa nos prédios da Universidade Federal do Pará: Mirante do Rio, Laboratório de Engenharia Elétrica e Computação (LEEC) e Centro de Excelência e Eficiência Energética (CEAMAZON); com o objetivo prover conectividade ao campus. As aplicações das quais este trabalho trata são as aplicações móveis: **Ônibus Circular** e **Intermunicipal**, ambos elétricos; e as aplicações estáticas (**Sistema de Baterias** e **Sistema Fotovoltaico**). Dessas 4 aplicações 1 delas transmite as informações diretamente para o servidor utilizando os protocolos GSM/GPRS/MQTT, porém as outras 3 transmitem dados utilizando a tecnologia LoRa. Os gateways tem objetivo de receber a informação LoRa, demodular e retransmitir para o servidor.

Boa parte da pesquisa envolvendo as aplicações móveis foi desenvolvida pelos pesquisadores Luz (2022) e Silva (2022), porém um dos objetivos deste trabalho é trazer algumas melhorias e/ou análises mais detalhadas destas transmissões que os autores não fazem. Quanto

as aplicações estáticas, foi necessário criar dispositivos que medissem as informações elétricas do Sistema Fotovoltáico e Sistema de Armazenamento de Energia e enviassem para end-node transmitir por LoRa. Desta forma uma equipe de pesquisadores do Laboratório de Sensores e Sistemas Embarcados (LASSE) desenvolveu dois dispositivos para realizar esta tarefa denominados de Unidades de Aquisição de Corrente e Tensão. Para medição das variáveis do **Sistema Fotovoltáico** eles desenvolveram o UACT 3F (analisada no trabalho de Souza (2023)), e para o **Sistema de Baterias** foi criado a UACT CC (trabalho que ainda será publicado).

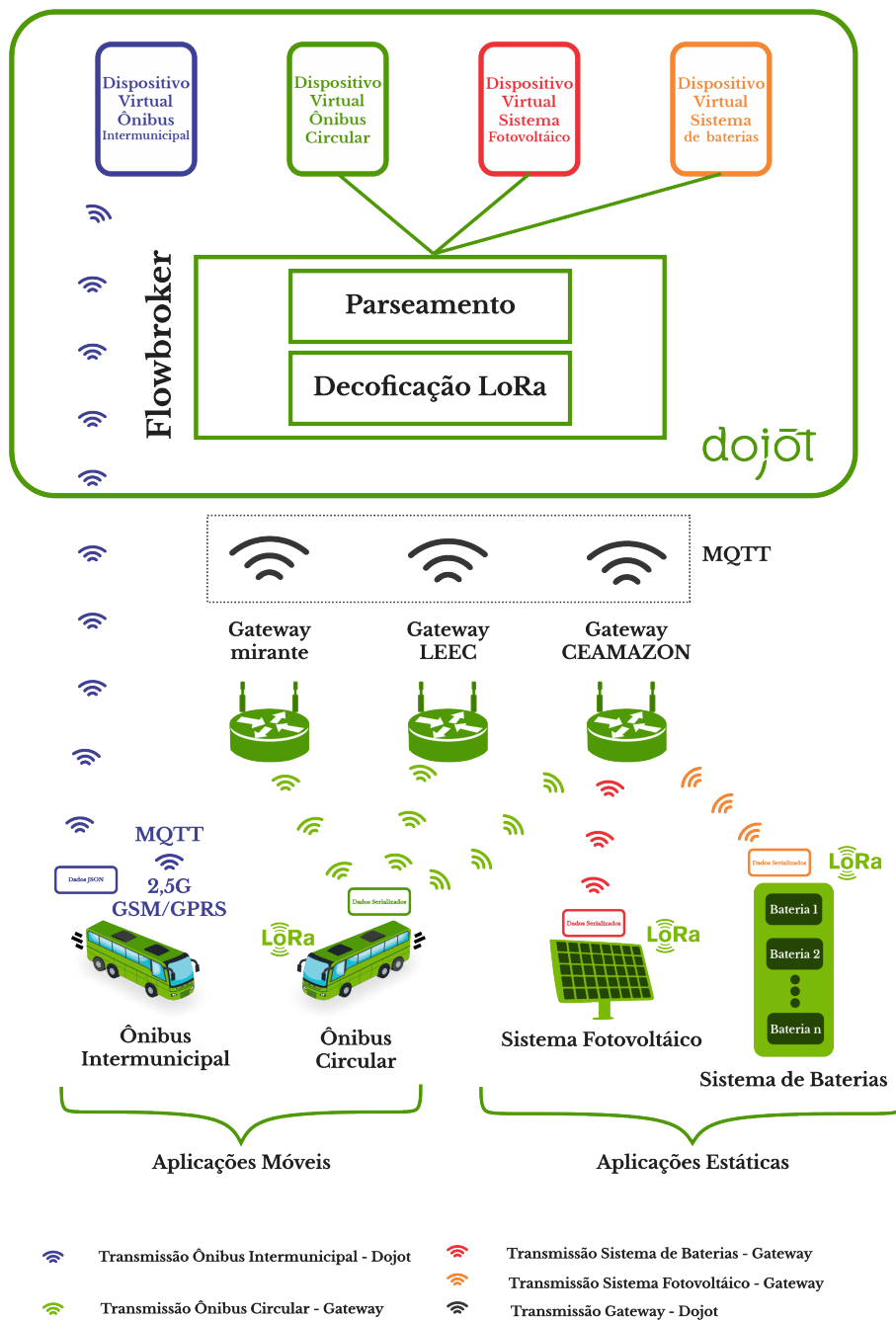
O protocolo de comunicação implementado pelo LASSE para acesso das variáveis de cada aplicação foi o MODBUS e como dispositivo possui algumas especificidades, foi necessário a criação de uma biblioteca para acesso desses dados (explicado melhor no capítulo de Metodologia) e disponibilizada no Github.

Para instalação da Dojot, foi utilizado o software Promox, desenvolvido para criação de máquinas virtuais em linux. Nele foram instalados a máquina virtual que contém a plataforma de middleware Dojot que além de possuir um banco de dados em MySQL para armazenar grande quantidade de informações ainda foram instalados dois ambientes com a Dojot, sendo um de desenvolvimento (*docker compose*) e um de produção (*kubernetes*). O ambiente utilizado neste trabalho foi o *kubernetes*.

1.3 Arquitetura de Transmissão

Para enfatizar a transmissão de dados em si, a Fig. 2 ilustra como ocorre a transmissão de dados relacionadas as 4 aplicações. É importante observar que no servidor existe um componente denominado de *flowbroker* que é responsável por decodificar a informação que é enviada criptografada desde o end-node (criptografia AES128) e desta forma o bloco de decodificação possui as mesmas chaves que as configuradas no transmissor LoRa. O bloco de parseamento é utilizado para enviar as informações já descriptografadas para os respectivos dispositivos virtuais da Dojot, onde cada variável pode ser monitorada individualmente. Como já mencionado, o **Ônibus Intermunicipal** envia dados diretamente para a Dojot e não utiliza o gateway, tampouco o *flowbroker* Isto por que a tecnologia de envio GSM já garante a segurança dos dados e o protocolo MQTT permite o envio diretamente para o dispositivo virtual.

Figura 2 – Estrutura geral de transmissão do smart campus da UFPA



Fonte: Elaborado pelo Autor

1.4 Justificativa

Um dos principais diferenciais deste trabalho é a produção dos hardwares transmissores LoRa e GSM de forma *opensource* para que outros projetistas, profissionais e acadêmicos possam replicar, testar e aprimorar seus próprios projetos baseados em sistemas IoT. Além disso, neste trabalho será utilizada uma plataforma brasileira de *middleware* conhecida como Dojot, que além de ser *opensource* e gratuita, ela foi desenvolvida para atender demandas de projetos IoT para *Smart Cities* (CPQD, 2017). Além disso, a estrutura apresentada neste trabalho apesar de

ter sido testada no *Smart Campus* da UFPA, pode ser utilizada em outros sistemas IoT como *Smart Homes*, *Smart Cities*, desde que o projetista faça os devidos ajustes para que a solução se adeque a sua realidade.

Um outro ponto que vale ressaltar é a escalabilidade que este projeto traz, já que a serialização de dados possui uma estrutura que permite que novas variáveis sejam incluídas no payload de transmissão LoRa, que além de garantir a confiabilidade dos dados devido a criptografia AES128, permite que os dispositivos sejam monitorados individualmente nos dispositivos virtuais da Dojot.

Além disso, este trabalho tem grande relevância para as pesquisas que envolvem Internet das Coisas que é um dos assuntos mais estudados na atualidade, devido sua grande quantidade de trabalhos que podem ser desenvolvidos para promover o desenvolvimento tecnológico de cidades. Este trabalho contribui significativamente para o desenvolvimento do *Smart Campus* da UFPA e como mencionado anteriormente pode ser reutilizado em outros locais.

1.5 Objetivos

1.5.1 Objetivo Geral

Perante as informações mencionadas, este presente trabalho tem por objetivo integrar hardwares e softwares, utilizando as tecnologias LoRa e GSM para transmissão de dados, tendo como estudo de caso o *Smart Campus* da Universidade Federal do Pará.

1.5.2 Objetivos Específicos

- Realizar a serialização dos dados para as aplicações, **Ônibus Circular**, **Sistema Fotovoltáico** e **Sistema de Baterias**;
- Realizar metodologia de ativação automática e reconexão da rede GSM relacionada à aplicação do **Ônibus Intermunicipal**;
- Desenvolver biblioteca em C/C++ (*software*) para recebimento dos dados via protocolo MODBUS para as aplicações: **Sistema Fotovoltáico** e **Sistema de Baterias**;
- Desenvolver os end-nodes (*hardware*) para integração MODBUS/LoRa das aplicações **Sistema Fotovoltáico** e **Sistema de Baterias**;

1.6 Estrutura do Trabalho

Este trabalho está dividido em 6 capítulos com acréscimo da conclusão e referências. A descrição de cada um, com exceção da introdução, está descrita abaixo:

- **Capítulo 2:** Contém o estado da arte, relacionada aos trabalhos científicos recentes a respeito das tecnologias utilizadas nesta dissertação, bem como os trabalhos correlatos que tiveram relação direta para o desenvolvimento desta pesquisa;
- **Capítulo 3:** Aborda toda a teoria utilizada no trabalho, relacionada aos protocolos de comunicação e o funcionamento da Dojot;
- **Capítulo 4:** Aborda todos os métodos utilizados, desde a arquitetura individual de cada aplicação, como os dispositivos que as compõe, a elaboração e funcionamento dos códigos e seus respectivos fluxogramas.
- **Capítulo 5:** Descreve os resultados obtidos com a implementação da metodologia.

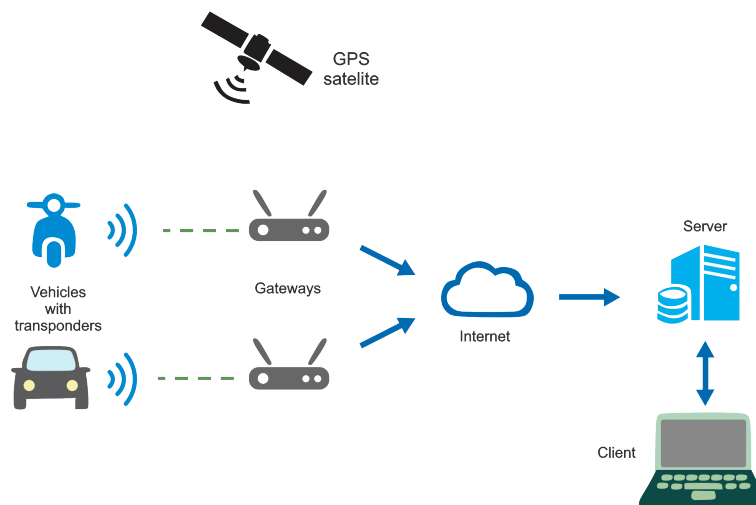
2 Revisão de Literatura

2.1 Conteúdo do Capítulo

Este capítulo discorre sobre que possuem relação com esta dissertação desde trabalhos que foram com relação direta ao Smart Campus da UFPA, como outros que utilizam ferramentas e tecnologias semelhantes às descritas nesta dissertação.

Ramli et al. (2019) desenvolvem em seu trabalho um sistema IoT para monitoramento de geolocalização e velocidade de um automóvel. A arquitetura (Fig. 3) consiste em basicamente três componentes: (i) O *shield* LoRa GPS dentro de um veículo móvel controlado por um Arduino Uno que transmite sua geolocalização; (ii) o gateway LoRa Dragino LG01 (dispositivo fixo) que recebe os dados de geolocalização, e (iii) um servidor desenvolvido pelos autores com base no software *open source* LAMP (Linux, Apache, MySQL e PHP). A velocidade pode ser estimada através do *timestamp*¹ e das distâncias entre as coordenadas geográficas latitude e longitude.

Figura 3 – Arquitetura do sistema de rastreamento de veículos



Fonte: (RAMLI et al., 2019)

O end-node desenvolvido pelos autores possui um bom acabamento com os componentes dentro de um acrílico, porém é alimentado externamente por uma bateria, o que torna o projeto menos automático, haja vista que necessita a troca da bateria quando sua carga se esgota.

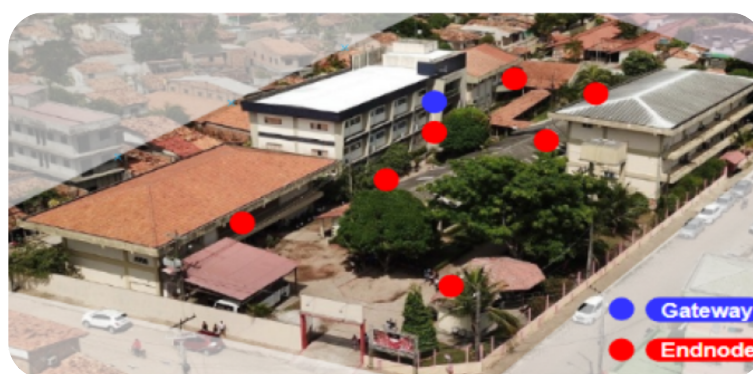
Já em (FACINA RAUL G. TANCREDO, 2021), os autores desenvolvem um setup de transmissão LoRa para avaliar o alcance da comunicação na zona rural. O setup de transmissão se baseia em 2 dispositivos LoRa: um transmissor formado por um módulo LoRa SX1276 e um dispositivo GPS NEO-6M-M-001 conectados a um esp 32; e um gateway de baixo custo formado

¹ Na computação *timestamp* é uma variável que guarda informações de hora e data

por um transceptor LoRa acoplado a um Raspberry Pi. Para avaliar o alcance da transmissão os autores propuseram enviar dados de RSSI² para o gateway. Porém os dados são enviados em formato .txt no cartão SD do Raspberry Pi e avaliados de forma offline e dessa forma os autores não possuem servidor para avaliação em tempo real dos dados.

Em Carvalho, Sá e Farias (2021), os autores utilizam um sistema IoT para monitoramento ambiental em um cenário de *smart campus*, de forma que os dados a serem analisados são meteorológicos e de gases efeito estufa: temperatura, umidade, CO , CO_2 e C_4H_{10} . Os sensores posicionados em locais estratégicos na UFPA de Cametá (Fig. 4), são transmitidos utilizando a tecnologia LoRa. Tais informações são recebidas por um gateway que retransmite para a Dojot utilizando o protocolo MQTT.

Figura 4 – Cenário de medição - Campus Cametá



Fonte: (CARVALHO; SÁ; FARIAS, 2021)

Os autores demonstraram a importância da utilização de uma arquitetura IoT que contém tecnologias como LoRa e Dojot, pois além de permitir monitoramento em tempo real de mais de uma variável, ainda promovem o desenvolvimento tecnológico da universidade, transformando-a em um *smart campus*.

Os autores (MANUEL et al., 2022) desenvolvem em seu artigo um sistema IoT para rastreamento de ônibus em tempo real no *smart campus* da Universidade Nacional de Engenharia (*Universidad Nacional de Ingeniería - UNI*) no Peru. Sua arquitetura consiste em 3 componentes:

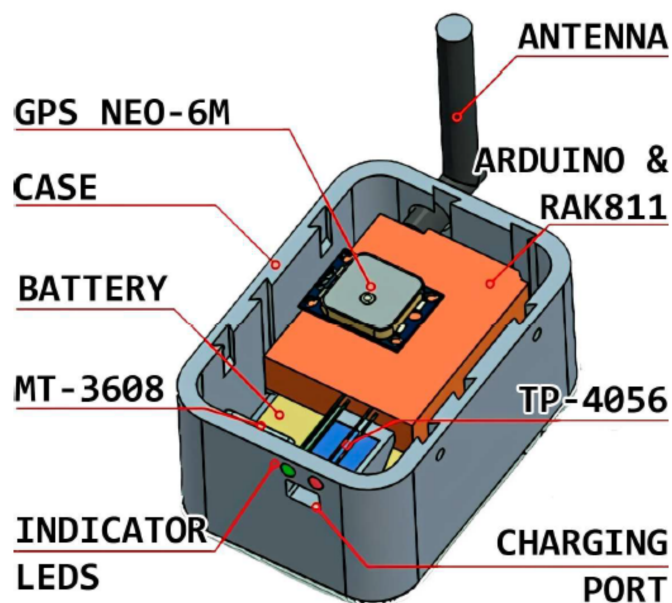
- **End-node:** instalado no interior do ônibus que circula no campus da UNI, o end-node é responsável por transmitir a geolocalização para o gateway utilizando tecnologia LoRa. Este dispositivo possui algumas características:
 - O dispositivo utilizado foi o LoRaWAN RAK811 V1.2;
 - O end-node não é alimentado diretamente no ônibus;
 - End-node precisa de alimentação própria e por isso, utiliza uma bateria de 5.000mA.

² Em telecomunicações, este parâmetro é utilizado para medir a qualidade de transmissão

- **Transmissão não contínua:** Para economizar energia, os autores optaram por realizar a transmissão da geolocalização a cada 5 minutos.
- **Gateway:** o gateway utilizado foi um gateway LoRaWAN denominado RAK 7258;
- **Servidor:** Para que o ônibus seja monitorado em tempo real, os autores optaram por utilizar a LoRaWAN server denominada *The Things Network*.

O projeto 3D desse invólucro foi projetado no software GRABCAD, como é mostrado na Fig. 5.

Figura 5 – Design do invólucro contendo o transmissor LoRa



Fonte: (MANUEL et al., 2022)

citeonlinesemish descreve a instalação do *cluster kubernetes*, um dos ambientes de desenvolvimento da Dojot utilizada neste trabalho. Em seu artigo, o autor demonstrou através de simulações, que com a instalação deste sistema é possível realizar conectar diversos dispositivos transmitindo dados paralelamente além de consumir apenas 31,46% de CPU em seu valor máximo. Através deste trabalho foi possível otimizar a Dojot para se adequar ao projeto de *Smart Campus* da UFPA.

Lobato (2022) desenvolve um *middleware* escalável para aplicações IoT. O projeto desenvolvido pela autora tem como objetivo personalizar o servidor brasileiro da Dojot, instalado na UFPA, para promover o desenvolvimento tecnológico do campus da UFPA e gerenciar as diversas aplicações presentes no campus com a criação do projeto SIMA.

Dentre as diversas contribuições da autora, podem ser citadas:

- Instalação do ProxMox, Criação de máquinas virtuais, instalação e configuração da Dojot;
- Criação de dispositivos que consistem nas aplicações que contém as variáveis para monitoramento;
- Criação dos perfis de usuários, já que o projeto SIMA contém diversos grupos de pesquisa;

A principal semelhança do trabalho de Lobato (2022) e o descrito nesta dissertação é que ambos se referem ao projeto de *Smart Campus* da UFPA, entretanto a autora se dedica ao desenvolvimento da camada de servidor enquanto que nesta dissertação o objetivo é a transmissão de dados para a camada de servidor, através do desenvolvimento de hardwares e softwares que cumpram este objetivo.

Em (SILVA, 2022), a autora desenvolve um protótipo de um dispositivo que transmite informações do circular, ônibus que transita no campus da UFPA, para a Dojot. Tal aplicação possui o CFP 100 (dispositivo contador de pessoas), GPS e sensor de temperatura conectado ao Arduino que envia tais informações por LoRa. Para que as informações cheguem a Dojot, um gateway é utilizado para receber as informações em LoRaWAN e retransmitir para o servidor (Dojot). Como uma de suas contribuições, esta dissertação analisa a área de cobertura dos 3 gateways instalados no *smart campus* da UFPA.

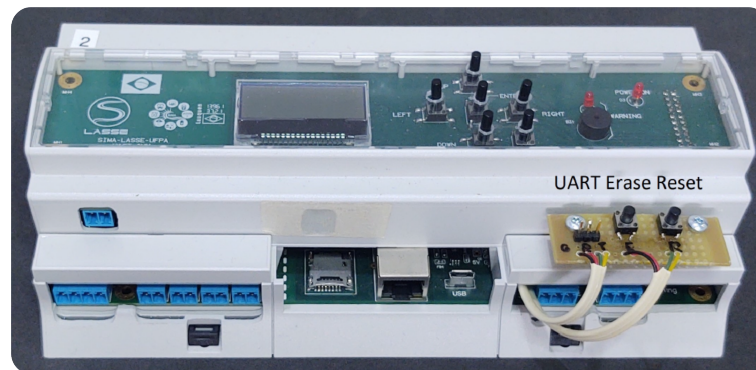
Luz (2022) desenvolve um protótipo para transmissão de dados do ônibus intermunicipal (UFPA belém - UFPA-castanhal) utilizando tecnologia GSM. A aplicação também possui objetivo de enviar informações de GPS, temperatura e contador de pessoas (CFP 100). A proposta de melhoria implementada em relação ao trabalho de (LUZ, 2022), foi tornar o processo de conexão e reconexão da rede mais automático além de melhorar o setup do circuito transmissor.

Além disso, os autores Luz (2022) e Silva (2022) não mostram a metodologia de serialização de dados além de não detalharem o método utilizado na Dojot para recuperação dos dados.

Em (JÚNIOR, 2022), o autor desenvolve um sistema de geração e armazenamento de energia para o campus UFPA. Destacam-se o sistema fotovoltaico e o sistema de armazenamento de energia. O primeiro é responsável por converter parte da energia solar em eletricidade, e o segundo, por armazenar a energia elétrica excedente para alimentar o prédio do CEAMAZON em situação de possíveis *blackouts*. Tais aplicações foram umas das responsáveis pela elaboração dessa pesquisa já que um dos objetivos é realizar suas transmissões para a Dojot.

Através do trabalho de Souza (2023) foi possível acessar as variáveis do sistema fotovoltaico, aplicação também relacionada ao *smart campus*, devido o desenvolvimento da Unidade de Aquisição de Corrente e Tensão Trifásica - UACT 3F, descrito em seu trabalho. O dispositivo desenvolvido tem por finalidade capturar as variáveis da aplicação, armazenando essas informações em registradores e disponibilizando via comunicação serial, através do protocolo MODBUS. O dispositivo desenvolvido pelo autor pode ser observado na Fig. 6.

Figura 6 – Unidade de Aquisição de Corrente e Tensão



Fonte: (SOUZA, 2023)

2.2 Conclusão do Capítulo

Este capítulo apresentou alguns trabalhos envolvendo as tecnologias utilizadas nesta dissertação sendo elas LoRa, GSM e MODBUS, ressaltando suas importâncias para a construção de um sistema IoT além de mostrar a flexibilidade que essas tecnologias possuem em variadas arquiteturas.

3 Fundamentação Teórica

3.1 Conteúdo do Capítulo

Para compreender os resultados obtidos neste trabalho é necessário antes abordar conceitos imprescindíveis. Este capítulo apresentará conteúdos relacionados aos protocolos de comunicação, as tecnologias e hardwares utilizados nesta dissertação.

3.2 A Indústria 5.0

A Indústria 5.0, ou quinta revolução industrial, é uma etapa emergente na sociedade atual, que incorpora tecnologias digitais avançadas, como a Internet das Coisas (IoT), a inteligência artificial (IA), a robótica e a computação em nuvem. A Indústria 5.0 visa integrar essas tecnologias para criar sistemas de produção mais eficientes e flexíveis que possam se adaptar rapidamente às demandas do mercado e dos consumidores.

Conforme a pesquisa realizada por Xu et al. (2021), a Indústria 5.0 é a próxima fase da evolução da indústria, que será marcada pela cooperação entre humanos e sistemas autônomos inteligentes, em que os sistemas autônomos ajudarão os humanos a tomar decisões mais informadas e a aumentar a eficiência da produção.

A implementação da Indústria 5.0 trará diversos benefícios para as empresas, incluindo a redução de custos de produção, a melhoria da qualidade do produto, a redução do tempo de ciclo e a capacidade de atender às demandas do mercado com mais rapidez. Além disso, a Indústria 5.0 também pode melhorar a segurança e a saúde dos trabalhadores, bem como a sustentabilidade ambiental da produção industrial (XU et al., 2021).

Dentro deste contexto a Internet das Coisas se destaca já que neste paradigma diversos objetos podem se conectar

A Internet das Coisas desempenha um papel fundamental na Indústria no paradigma da indústria 5.0, já que permite que os dispositivos conectados em rede troquem informações e dados em tempo real, o que pode ajudar a otimizar processos de fabricação, identificar e corrigir falhas rapidamente, e melhorar a eficiência energética, entre outras coisas (NARDO; YU, 2021).

3.3 A Internet das Coisas

A Internet das Coisas (IoT) é uma rede de dispositivos interconectados que coletam e trocam dados entre si sem intervenção humana direta. Esses dispositivos incluem desde sensores e termostatos inteligentes em casas conectadas, até equipamentos industriais em fábricas e

veículos autônomos nas ruas. A IoT tem o potencial de transformar drasticamente vários setores, desde saúde e agricultura até transporte e energia.

Segundo um relatório anual da Cisco (2020), em 2023 o mundo terá mais de 29 bilhões de dispositivos conectados. Isto impacta diretamente no fluxo de capital que este paradigma poderá causar e de acordo com uma pesquisa da Company (2015), o valor econômico gerado pela IoT em todo o mundo pode chegar a 11,1 trilhões de dólares até 2025, equivalente a 60,93 trilhões de reais considerando a cotação de R\$5,30. Isso significa que a IoT não é apenas uma tendência, mas uma revolução tecnológica que já está mudando a forma como as pessoas vivem, trabalham e interagem com o meio em que se vive.

No entanto, a IoT também apresenta desafios significativos em relação à privacidade e segurança de dados. Com tantos dispositivos conectados, é crucial que os desenvolvedores busquem soluções que visem dentre outros fatores, a segurança da informação para garantir confiabilidade na transmissão de dados. Uma das formas de garantir esta segurança é a utilização da criptografia na camada de enlace cujas chaves devem ser as mesmas na camada de aplicação, segundo o modelo OSI (S; J, 2011).

Muitos conceitos estão associados a esse paradigma (IoT) e atualmente muito se discute a respeito de casas inteligentes *smart homes*, cidades inteligentes *smart cities* e campus inteligentes *smart campus*.

3.4 *Smart Cities e Smart Campus*

Para compreender a definição de *smart campus* é necessário, entender o de cidades inteligentes. Segundo Matos (2021), *smart cities* são cidades que utilizam tecnologia da informação para melhorar a qualidade de vida de seus habitantes e contribuir para um desenvolvimento sustentável. Para que isto ocorra é necessário uma série de requisitos tais como: capital, liderança, planejamento, mão de obra qualificada, economia, sustentabilidade e principalmente tecnologia.

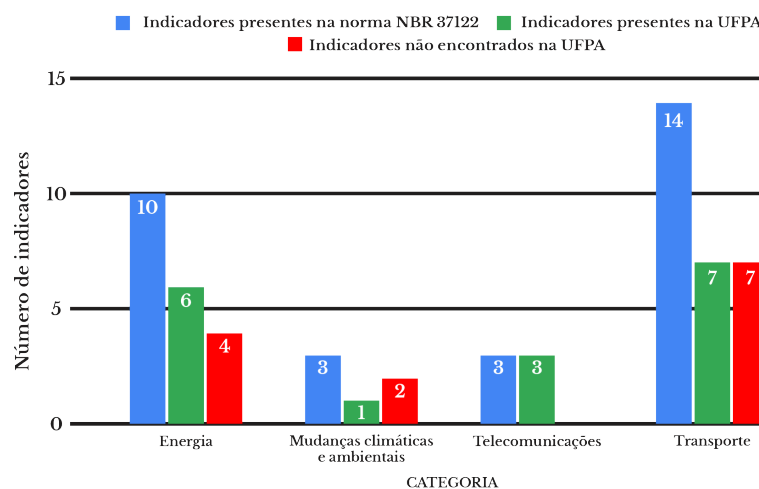
Pode-se entender o conceito de *smart campus* como o de uma *smart city*, pois segundo Pagliaro et al. (2016) os *campi* universitários podem ser adaptados às cidades. Para Sánchez-Torres et al. (2018), um campus inteligente é uma entidade universitária que usa tecnologia para suportar sua infraestrutura e seus processos, com o objetivo de melhorá-los para as pessoas. Tais conceitos são semelhantes às de uma cidade inteligente, e embora haja claras diferenças quanto ao tamanho e estrutura, o modelo de cidade inteligente pode ser adaptado para construir um *campus* inteligente. Além disso, assim como as cidades, as universidades apresentam desafios e dificuldades semelhantes tais como: impacto ambiental, ineficiência, falta de serviços, corte de verbas, entre outros.

Desta forma é necessário otimizar os recursos da universidade para prover o desenvolvimento tecnológico e melhorar a qualidade de vida dos estudantes e da comunidade local, bem

como promover o desenvolvimento sustentável. Em razão dos fatos mencionados, este trabalho desenvolve uma arquitetura de comunicação para transmissão de dados, relacionados ao sistema IoT implantado no *smart campus* da UFPA.

Lobato et al. (2021) analisam os indicadores de *smart cities* presentes no campus da UFPA devido a implantação do projeto SIMA na universidade, baseado na norma brasileira NBR 37122/2020, em que foram encontrados 17 indicadores no campus dentre os 30 presentes na norma NBR 37122 (cerca de 57%) entre as categorias: Energia, mudanças climáticas e ambientais, telecomunicações e transporte. O gráfico dos indicadores encontrados no campus, está ilustrado na Fig. 7.

Figura 7 – Total de indicadores encontrados por cada categoria



Fonte: Adaptado de (LOBATO et al., 2021)

3.5 Protocolos e Hardwares de Comunicação

O projeto SIMA, como já foi explicado, engloba várias aplicações. Cada aplicação possui seus próprios requisitos de implementação e dessa forma é necessário diferentes tecnologias de comunicação para que a transmissão funcione de maneira eficaz.

3.5.1 LPWAN

Conhecida como *Low-Power-Wide-Area-Network* atuam em comunicações de longa distância, que como o nome sugere, são conhecidas pelo baixo consumo energético. As LPWANs foram desenvolvidas para aplicações em IoT e muitas pesquisas são desenvolvidas a respeito dessa tecnologia. Todavia em um sistema IoT complexo é necessário utilizar diversas tecnologias que atendam os requisitos de cada etapa ou aplicação do sistema IoT. Dentre as tecnologias LPWAN estão a Sig-fox, LoRa e NB-IoT como mostra a Tabela 1.

De forma geral a Sigfox utiliza Chaveamento de Troca de Fases Binárias Diferenciais (*Differential Binary Phase-Shift Keying* - DBPSK) e opera com uma largura de banda de 100 Hz,

Tabela 1 – Redes LPWANs e suas configurações principais

	Sigfox	NB-IoT	LoRa
Frequência (MHz)	868,1 – 868,3 <i>uplink</i> 869,425 – 869,625 <i>downlink</i>	832 – 862 <i>uplink</i> 791 – 821 <i>downlink</i>	Frequências ISM não licenciadas
Tx Power (dBm)	14 - <i>uplink</i> 27 - <i>downlink</i>	23 - <i>uplink</i> 37 - <i>downlink</i>	14 - <i>uplink</i> 27 - <i>downlink</i>
Modulação	DBPSK - <i>uplink</i> GSFK - <i>downlink</i>	GMSK	Chirp Spread Spectrum
Largura de banda	100 Hz - <i>uplink</i> 600 Hz - <i>downlink</i>	180 KHz	125,250 e 500 KHz
Tamanho máximo do payload ¹ (bytes)	12 - <i>uplink</i> 8 - <i>downlink</i>	128 - <i>uplink</i> 68 - <i>downlink</i>	242

Adaptado de (VEJLGAARD et al., 2017)

enquanto que a NB-IoT utiliza Modulação estreita da faixa (*Gaussian Minimum Shift Keying*) com largura de banda de 180 KHz. A tecnologia LoRa, um dos focos deste trabalho, opera sobre a modulação *Chirp Spread Spectrum* ou CSS operando sobre diferentes larguras de banda: 125, 250 e 500 KHz.

3.5.2 A Tecnologia LoRa

Do acrônimo *Longe Range* (longo alcance), a LoRa foi criada para reduzir o uso de repetidores (WIXTED et al., 2016) e desenvolvida para aplicações IoT, por oferecer baixo consumo energético, longo alcance, confiabilidade, segurança na transmissão dos dados e economia. O alcance de transmissão desta tecnologia é em torno de 2 km e 20 km para áreas urbanas e rurais respectivamente (RAMLI et al., 2019).

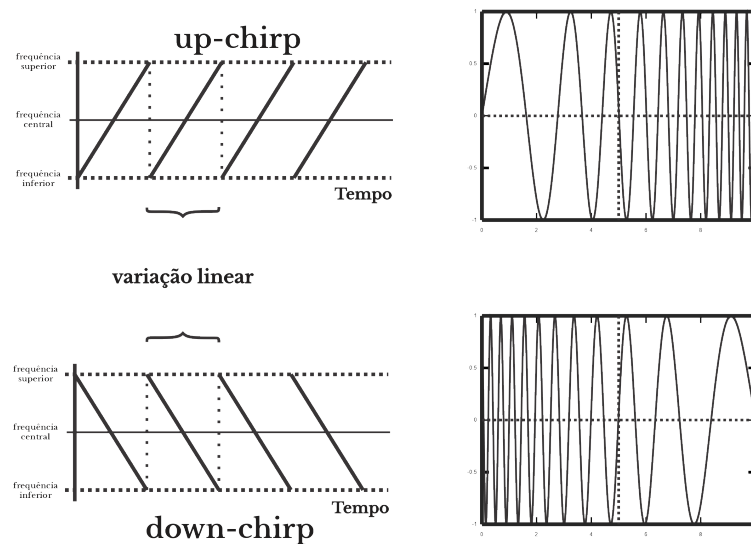
A comunicação LoRa pode ser considerada assíncrona, pois o transmissor pode enviar uma mensagem a qualquer momento, sem precisar esperar por uma resposta imediata do receptor. O receptor por sua vez escuta continuamente o canal de comunicação em busca de sinais, e quando recebe uma mensagem, ele a decodifica e envia uma resposta (se houver) para o transmissor. Isso permite que a transmissão LoRa seja mais eficiente em termos de consumo de energia e alcance, especialmente em ambientes com interferências de sinal (SEMTECH, 2022).

Para obter este baixo consumo de energia, a tecnologia LoRa utiliza a modulação CSS (*Chirp Spread Spectrum*) que mantém a característica de baixa potência da FSK (*Frequency Shifting Keying*), porém aumenta o alcance consideravelmente. Este tipo de modulação consiste em utilizar pulsos de chirp de forma que o sinal gerado é transformado e espalhado no domínio da frequência. Este passa a ter uma largura de banda maior que o sinal original, o que torna a transmissão mais robusta, pois reduz a interferência. Esta técnica tem sido muito utilizada por militares e por boa parte de aplicações IoT, haja vista que possui resiliência a ruído e interferência

(SEMTECH, 2022).

Um pulso de chirp é um sinal que varia sua frequência monotamente, de forma que quando a frequência aumenta tem-se o *up-chirp* e quando decresce tem-se o *down-chirp*. A Fig. 8 mostra como a técnica CSS varia os chirps em frequência linear.

Figura 8 – Representação da técnica CSS mostrando o *down-chirp* e *up-chirp*



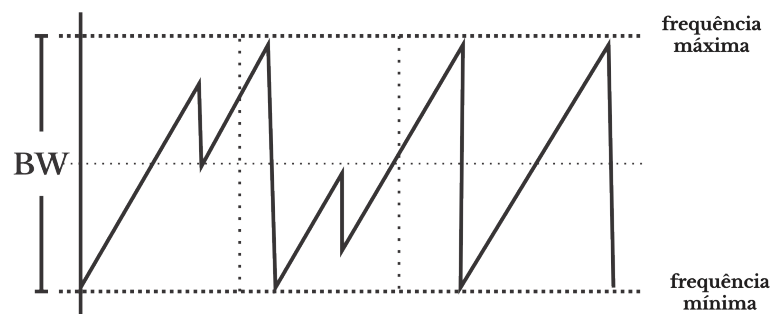
Fonte: Elaborado pelo Autor

A transmissão LoRa possui algumas características importantes, tais como: largura de banda, taxa de código, fator de espalhamento (SF) e tempo de transmissão que serão discutidos no próximo tópico.

3.5.2.1 Largura de Banda

A largura de banda ou *bandwidth* (BW) é definida como a faixa de frequência que o sinal opera sendo a diferença entre a frequência máxima e mínima, como mostra a Fig. 9.

Figura 9 – Largura de banda na tecnologia LoRa



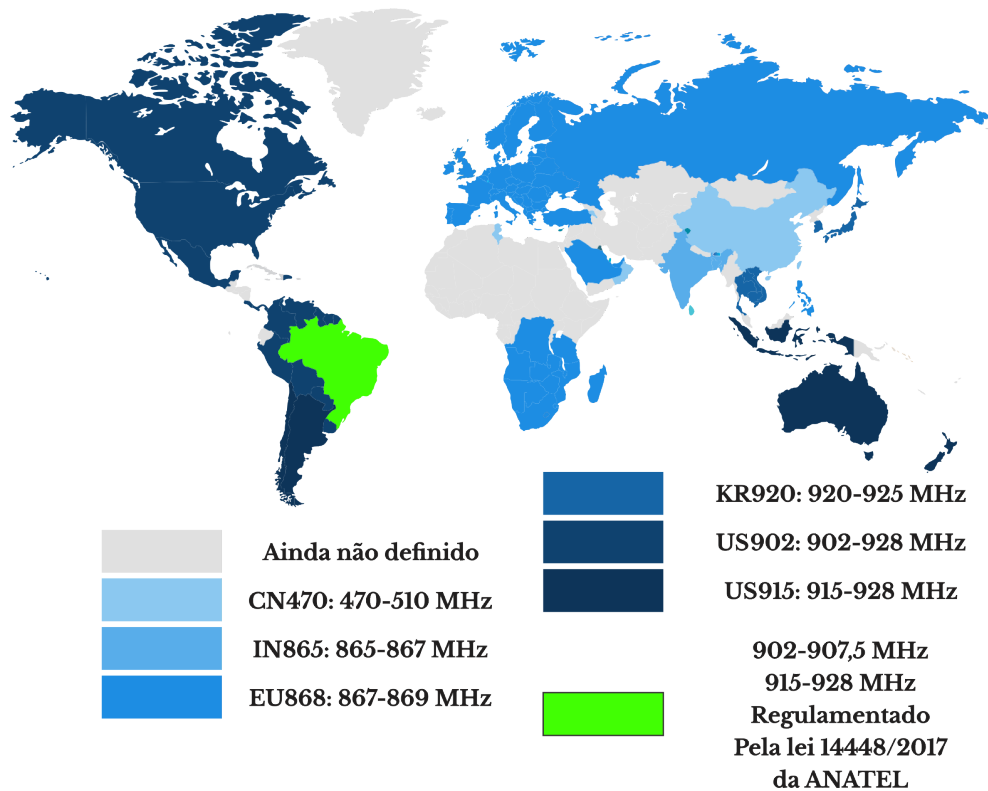
Fonte: Elaborado pelo Autor

A largura de banda pode sofrer um deslocamento superior ou inferior de até 20%, todavia esta variação não afeta a decodificação, haja vista que esta utiliza cristais que não necessitam de

extrema precisão, mantendo dessa forma o baixo custo. Os valores de largura de banda utilizados na LoRa são de 125, 250 e 500KHz (LOPES, 2021).

Para transmitir os dados, a LoRa utiliza frequências não licenciadas ISM (*Industrial Scientific and Medical*). Essas frequências são bandas específicas utilizadas no mundo todo para fins industriais e médicos e cada região opera em uma determinada faixa de frequência. Os Estados Unidos por exemplo utilizam a banda da América do Norte (902 – 928 MHz), enquanto que a China opera nas frequências de 470 – 510 MHz. A Agência Nacional de Telecomunicações (ANATEL) estabelece que o Brasil opere nas frequências de 902 – 907,5 MHz e 915 – 928 MHz (ANATEL, 2012). A Fig. 10 ilustra como as bandas ISM estão distribuídas no mundo todo.

Figura 10 – Frequências LoRa utilizadas no Mundo



Fonte: Elaborado pelo Autor

3.5.2.2 Taxa de Código

A taxa de código ou *Code Rate* (CR) é um parâmetro que acrescenta redundância ao pacote, de forma que a quantidade de bits geradas (n) é menor que a de bits úteis (k), definindo dessa forma a taxa de código $\frac{k}{n}$. Desta forma a taxa de códigos é dada pela Eq. 3.1.

$$CR = \frac{4}{4+n} \tag{3.1}$$

com $n \in \mathbb{N} | n = [1,4]$.

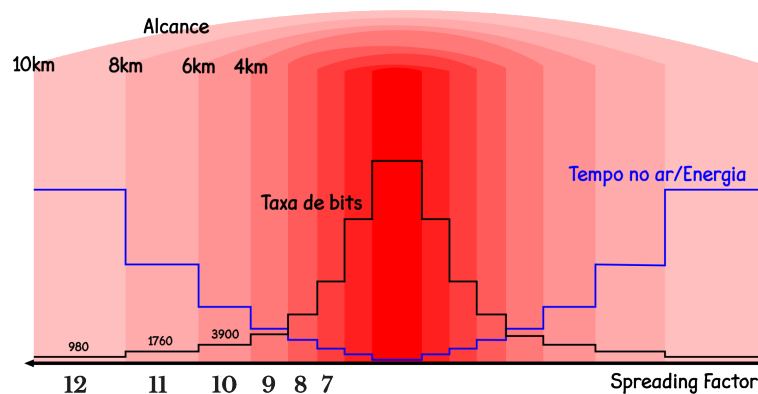
3.5.2.3 Fator de Espalhamento - SF

O fator de espalhamento ou *Spreading Factor* (SF) consiste na razão entre o logaritmo de base 2 da taxa de chirps (R_c) pela taxa de símbolos (R_s), como mostra a Eq. 3.2 (ADELANTADO et al., 2017).

$$SF = \log_2 \left(\frac{R_c}{R_s} \right) \tag{3.2}$$

Esse parâmetro possui 6 variações: SF7 ao SF12. Cada SF possui alcance, taxa de bits e gasto de energia específicos. A Fig. 11 mostra como essas três grandezas estão relacionadas com cada SF.

Figura 11 – Relação do parâmetro de cada SF com: tempo no ar, alcance e taxa de bits



Fonte: Elaborado pelo Autor

É possível notar um aumento significativo no alcance do sinal quando o SF é maior, porém há redução na taxa de bits. Além disso, cada SF opera em larguras de bandas específicas e cada associação entre largura de banda e SF possui uma capacidade máxima de transmissão de *payload*. A Tabela 2 mostra o tamanho máximo de payloads, considerando a banda US915 - 915 – 928MHz utilizada no Brasil.

Tabela 2 – Diferentes configurações de transmissão no canal LoRa para AUS915

Mode	Max payload size (bytes)
SF7/125kHz	242
SF8/125kHz	125
SF9/125kHz	53
SF10/125kHz	11
SF7/500kHz	222
SF8/500kHz	222
SF9/500kHz	222
SF10/500kHz	222
SF11/500kHz	109
SF12/500kHz	33

(TTN, 2023)

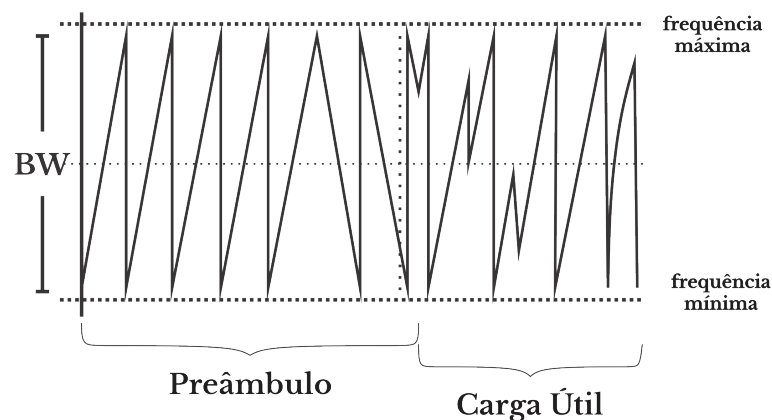
3.5.2.4 Indicador de Intensidade do Sinal Recebido

Na LoRa o RSSI (*Received signal strength indicator*) é utilizado para avaliar a qualidade do sinal transmitido. É através deste parâmetro que pode-se determinar se a força em que um dispositivo pode captar os sinais de rádio transmitidos. A medida que a distância em que um sinal é transmitido aumenta, a intensidade com a qual ele é captado diminui tanto em *uplink* como em *downlink* Lopes (2021).

3.5.2.5 Tempo de Transmissão do Pacote

Segundo, Emmanuel et al. (2020) um pacote LoRa é formado basicamente por um preâmbulo e uma carga útil (Fig. 12) e desta forma o tempo de transmissão (*time-on-air*) é dado pela soma individuais de cada parte do pacote. O *time-on-air* é o tempo de transmissão do nó LoRa para o gateway.

Figura 12 – Representação de um pacote LoRa



Fonte: (EMMANUEL et al., 2020)

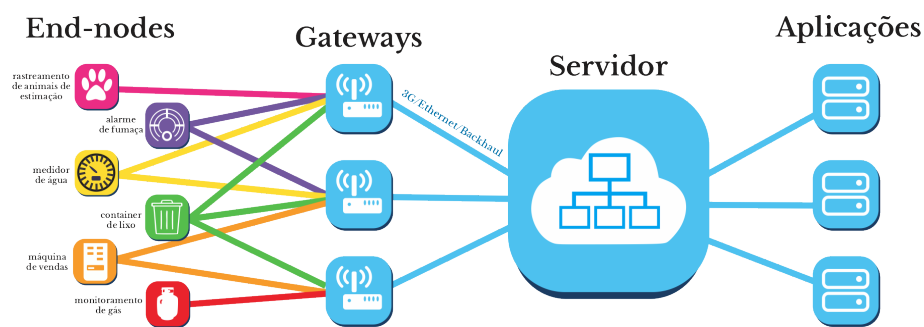
É notório que se há mais de uma variável transmitida é de suma importância que essas variáveis sejam concatenadas a fim de otimizar o tempo de transmissão, já que caso se opte por

transmiti-las individualmente de forma sequencial, o tempo de transmissão seria maior devido o grande número de preâmbulos no pacote. Em contrapartida, com a informação serializada, existe apenas um único preâmbulo por pacote.

3.5.3 O Protocolo LoRaWAN

LoRaWAN é o protocolo do qual a LoRA compõe a camada física. Projetado pela LoRa Alliance, a LoRaWAN é um protocolo de transmissão sem fio utilizado em sistemas IoT que possui de forma geral duas camadas: **a camada física** (LoRa) pela qual o link de comunicação é habilitado; e a **camada de protocolo** LoRaWAN. Este protocolo influencia diretamente o tempo de vida útil das baterias e possui um sistema de criptografia que garante a segurança nos dados (LORA ALLIANCE, 2015). Sua arquitetura estrela é mostrada na Fig. 13, em que é possível notar uma rede IoT com diversas aplicações interconectadas com o servidor LoRa.

Figura 13 – Arquitetura LoRaWAN



Fonte: Adaptado de (LORA ALLIANCE, 2015)

A arquitetura LoRaWAN pode ser dividida em 3 categorias:

- **End-nodes:** Também chamados de dispositivos finais, os end-nodes são sensores acoplados a microcontroladores que se comunicam com gateways, enviando payloads de informação criptografados.
- **Gateways:** São dispositivos intermediários capazes de receber os dados dos end-nodes e transmitir para os servidores.
- **Servidores:** São responsáveis por receber os dados advindos do gateway e decodificá-los para retransmitir para suas devidas aplicações.

3.5.3.1 Criptografia LoRaWAN

Para que a transmissão ocorra, mantendo a segurança e integridade dos dados, a LoRaWAN utiliza o padrão de criptografia AES128, que consiste, em suma, na utilização de 2 parâmetros: **Chave AppSKey:** Utilizada pelo nó e pelo servidor para criptografia e descryptografia,

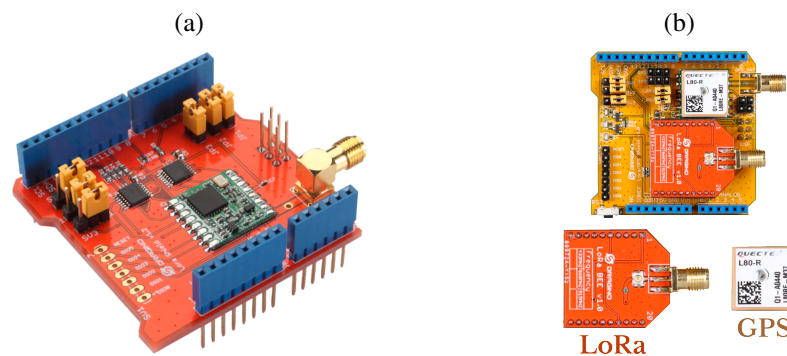
respectivamente; **Chave NwkSKey**: Tem a mesma função da *AppSKey*, porém esta verifica o código de integridade de todas as mensagens de dados;

3.5.4 O End-node LoRa

3.5.4.1 Visão Geral

O Dragino SX127x é um shield Arduino desenvolvido para aplicações LoRa. Desenvolvido com base no chip SEMTECH 12776/1278, O LoRa Shield Dragino permite que usuários enviem dados a longas distâncias com baixo consumo de corrente, tendo como alvo aplicações com sistemas de irrigação, medição inteligente, cidades inteligentes, entre outras aplicações IoT. Ainda existe o LoRa GPS Shield que engloba a tecnologia LoRa com o sistema de geolocalização e transmite informações de hora e data(*timestamp*), utilizado neste trabalho para uma das aplicações móveis que utiliza a tecnologia LoRa. Estes dois shields estão ilustrados nas Figs. 14a e 14b.

Figura 14 – Shields LoRa (a) LoRa Dragino Shield; (b) LoRa GPS Shield



Fonte: (SEMTECH, 2019)

3.5.4.2 Biblioteca

A biblioteca utilizada neste trabalho para transmissões LoRa foi a **lmic**², uma biblioteca robusta composta por diversas funções que permitem configurar parâmetros como SF e potência de transmissão. Também é possível configurar os parâmetros da criptografia AES128 utilizada pela LoRa (*DevAddr*, *NwkSKey* e *AppSKey*). E para utilização do módulo GPS presente no shield LoRa/GPS pode-se utilizar a biblioteca TinyGPS³.

3.5.5 O Gateway Utilizado para Comunicação LoRa

Em aplicações que utilizam a LoRa para transmitir informações, é necessário um gateway para que este atue como dispositivo intermediário entre o end-node e o servidor. Para isso foi

² disponível em: <github.com/matthijskooijman/arduino-lmic>

³ Disponível em: <<https://github.com/mikalhart/TinyGPS>>

utilizado o RAK7249, um dispositivo com capacidade de receber dados em LoRa, que além de ser projetado em *outdoor*⁴, possui uma interface gráfica, em que é possível configurar diversos parâmetros tanto de comunicação com o end-node quanto com o servidor instalado no sistema IoT. O RAK7249 possui um sistema de antenas que o faz ter compatibilidade com outras três tecnologias além da LoRa: LTE, GPS e WiFi, como pode ser ilustrado na Fig. 15.

Figura 15 – RAK7249 e suas 4 antenas



Fonte: Elaborado pelo Autor

3.5.6 Sistema Global para Comunicações Móveis

Do inglês *Global System for Mobile Communications*, GSM é uma tecnologia criada na década de 80 no contexto da criação da tecnologia de rádio digital (PIROTTI; ZUCCOLOTTO, 2009), para resolver o problema da incompatibilidade de tecnologias de comunicação.

O GSM consolidou-se como um padrão internacional de tecnologia para comunicações móveis em telecomunicações, sendo base para sistemas de rádio usados em celulares. Necessitando de comunicação síncrona⁵, o GSM é caracterizado por ser digital com frequência de operação de 1850 à 1990 MHz e sua modulação é a GMSK mencionado em 3.5.1 que é a mesma modulação usada pela NB-IoT, uma vez que sua base é o GSM (SCHILLER, 2003).

Segundo Schiller (2003), a arquitetura GSM é composta basicamente por 3 subsistemas:

- **Rede e Subsistema de Comutação (NSS):** É considerado o "coração" do sistema GSM (Sistema Global para Comunicações Móveis). O NSS é responsável por conectar a rede sem

⁴ Expressão utilizada para designar transmissões em ambientes externos que estão suscetíveis a sol e chuva (PANITZ, 2003)

⁵ Neste tipo de comunicação, os dispositivos precisam estar conectados à rede a todo momento. Caso essa conexão falhe, o dispositivo para de enviar e receber informações, necessitando reiniciar o processo síncrono de comunicação

fio com redes públicas padrão, realizar transferências entre diferentes BSSs (camada que cuida do acesso do usuário a rede), fornecer funções para localização mundial de usuários e suportar cobrança, contabilidade e roaming de usuários entre diferentes provedores em diferentes países. O NSS desempenha um papel importante no suporte ao funcionamento do sistema GSM como um todo, permitindo a conexão de usuários de telefonia móvel em todo o mundo;

- **Subsistema de Radio (RSS):** O RSS contém os componentes que são necessários para alocar os recursos de rádio para os assinantes individuais. Consiste principalmente nos componentes móveis (telefone móvel ou estação móvel, MS) e no subsistema da estação base (BSS);
- **Subsistema de Operação e Suporte (OSS):** A terceira parte de um sistema GSM, o subsistema de operação (OSS), contém o funções necessárias para a operação e manutenção da rede.

Apesar de todas essas funcionalidades, uma das maiores limitações da rede GSM era o não acesso a internet. No início da década de 2000 os telefones celulares utilizavam GSM, porém os serviços ofertados eram limitados a voz e SMS. Por esta razão a tecnologia GPRS foi criada como um complemento a rede GSM, pois insere os protocolos TCP/IP e da início a geração 2,5G com o início da internet nos telefones celulares, consolidada apenas com o surgimento do 3G (PINHÃO, 2006). Vale ressaltar que a rede GSM pode funcionar sem a GPRS, porém o inverso não é válido, já que a GPRS depende da GSM para funcionar.

3.5.7 O Hardware GSM/GPRS

3.5.7.1 Visão Geral

O SIM900 é um shield Arduino utilizado para comunicação GSM. Além de possibilitar os diversos serviços GSM, tais como: chamada de voz, mensagens SMS e GPRS (conexão com a internet), pode ser utilizado para transmissões via protocolo TCP/IP, HTTP e para comunicação entre máquinas o MQTT. Ele possui suporte para inserção de um cartão SIM e o projetista tem a liberdade de escolher qual operadora vai ser utilizada para a contratação dos serviços GSM/GPRS.

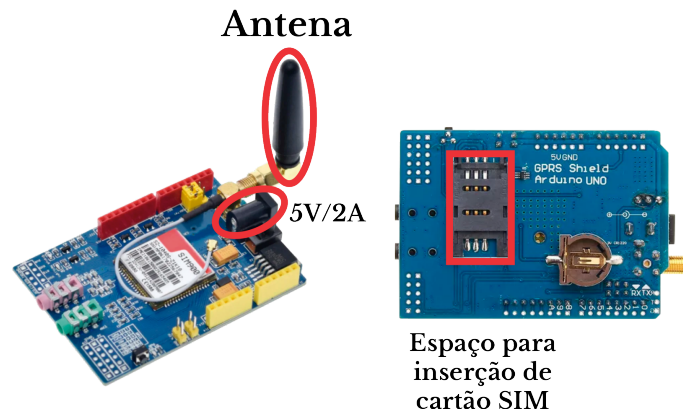
Quanto a sua alimentação, o SIM900 precisa 5V/2A para funcionar corretamente (SIM-COM, 2013). O *hardware* é ilustrado na Fig. 16.

3.5.7.2 Características do *hardware*

Este hardware possui 3 LEDs específicos que informam visualmente algumas informações a respeito do shield:

- **PWR:** Este LED indica se o shield está alimentado.

Figura 16 – Shield SIM900



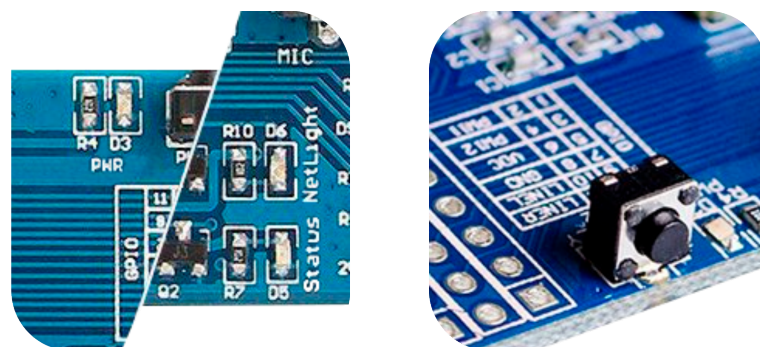
Fonte: Elaborado pelo Autor

- **Status:** Indica se o cartão SIM está funcionando corretamente. Caso esteja de alguma forma danificado ou mesmo desligado, este LED permanecerá apagado.
- **Netlight:** Este LED pode fornecer diferentes informações de acordo com a frequência em que ele acende e apaga:
 - **64ms ligado e 800ms desligado:** O cartão SIM está funcionando porém ainda não há conexão com a rede de telefonia;
 - **64ms ligado e 3s desligado:** O cartão SIM pôde se conectar a rede GSM, e dessa forma pode enviar mensagens de voz e SMS;
 - **64ms ligado e 300ms desligado:** O cartão está conectando a rede GPRS e portanto pode acessar a internet.

Há ainda um botão que permite ativar ou desativar o chip que por padrão vem desativado, quando se energiza o shield. Logo este botão precisa ser pressionado para ativar o cartão SIM.

Os LEDs mencionados e o botão estão ilustrados na Fig. 17.

Figura 17 – LEDs indicadores de status e botão que aciona o cartão SIM



Fonte: Elaborado pelo Autor

3.5.7.3 Biblioteca

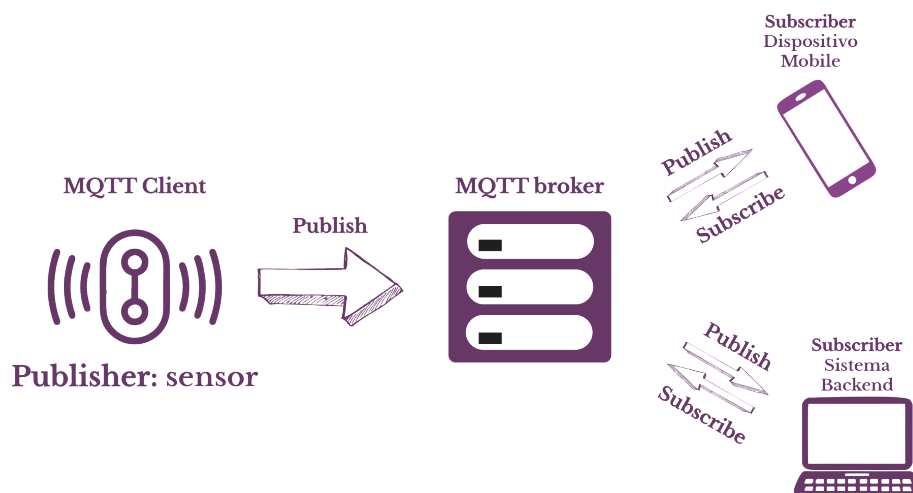
Para gerenciar os serviços de GSM e GPRS foi utilizada a biblioteca **TinyGsm**⁶ que possui diversas funções que facilitam a implementação do projeto desejado. Orientada a objetos, esta biblioteca possui diversos métodos, entre os quais podem se destacar:

- **isNetworkConnected:** Método booleano que verifica se o shield conseguiu se conectar a rede GSM;
- **gprsConnect:** Método booleano que verifica se o hardware se conectou a GPRS, portanto à internet;
- **Restart:** Método que inicia os processos do SIM900.

3.5.8 O Protocolo MQTT

Este protocolo foi desenvolvido utilizando como base o TCP/IP, porém voltado para comunicação entre máquinas e portanto suas regras são mais simples. Sua estrutura é mostrada na Figura 18. O broker é o componente intermediário que recebe os dados vindos de um *publisher* e os envia para o *subscriber* (OASIS, 2019).

Figura 18 – Estrutura do MQTT



Fonte: Elaborado pelo Autor

Os componentes do MQTT são:

- **Publisher:** São os clientes que possuem as informações de interesse para os dispositivos finais (*subscribers*) da rede MQTT. São representados por sensores, que geralmente estão acoplados a alguma tecnologia com suporte ao MQTT.

⁶ Disponível em: <github.com/vshymansky/TinyGSM>

- **Broker:** Componente intermediário que recebe as informações publicadas pelo *publisher* e envia para os *subscribers*;
- **Subscriber:** São os dispositivos finais que processam a informação dos *publishers* e podem ser dispositivos móveis ou servidores na camada *backend*.

3.5.9 O Protocolo MODBUS

Segundo sua documentação (MODBUS, 2012), o MODBUS é um protocolo de mensagens da camada de aplicação para comunicação cliente/servidor entre dispositivos conectados em diferentes tipos de barramentos ou redes. O seu funcionamento, de forma geral, pode ser entendido como a comunicação entre dois dispositivos: o que faz a requisição (*master*) de uma ou mais variáveis através de uma mensagem que deve atender alguns requisitos do protocolo (quadro MODBUS de requisição); e o que recebe a requisição (*slave*) e envia os dados de interesse (quadro MODBUS de resposta) se o quadro de requisição estiver dentro das normas do protocolo.

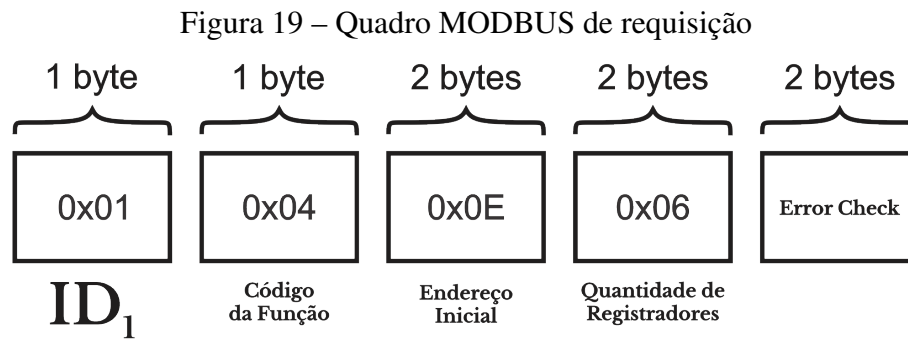
3.5.9.1 O Formato do quadro MODBUS de Requisição

A mensagem que deve ser enviada ao *master* possui 8 bytes e contem os parâmetros na representação hexadecimal como descrito abaixo:

- **Endereço do dispositivo (ID_1) - 1 byte:** *Device Address* é o endereço de acesso do dispositivo que responderá as requisições. Este código é usado caso haja mais de um *master* conectado a um único *slave*.
- **Código da função - 1 byte:** O protocolo MODBUS possui 21 códigos de função (*Function Code*), porém para este trabalho apenas o código 0x04, que representa o acesso aos registradores do tipo *input*, será utilizado.
- **Endereço inicial - 2 bytes:** *Init Address* é utilizado para especificar o endereço do registrador que contém o início dos dados desejados.
- **Quantidade de registradores - 2 bytes:** *Quantity Registers* representa a quantidade de registradores que será lido a partir do registrador inicial, de forma que cada registrador comporta 2 bytes.

O quadro modbus ainda reserva os dois últimos bytes para um código, denominado de *Error Check*, que verifica a integridade do mensagem de requisição. O código abaixo mostra como se calcula o *Error Check* em C++ criando uma classe que retorna uma variável de 2 bytes (*uint16_t*).

A Figura 19 mostra um exemplo de um quadro MODBUS de requisição.



Fonte: Elaborado pelo Autor

Neste exemplo: tem-se o endereço do registrador inicial tendo o valor de E em hexadecimal, ou seja, a carga útil começa pelo registrador 14 em decimal e possui 6 registradores a partir do registrador 14, sendo: 14, 15, 16, 17, 18, 19, comportando 2 bytes cada. Este quadro MODBUS está requisitando 12 bytes ou 3 variáveis float que já são transmitidas no formato IEEE754 *single precision*.

3.5.9.2 O Formato do quadro MODBUS de Resposta

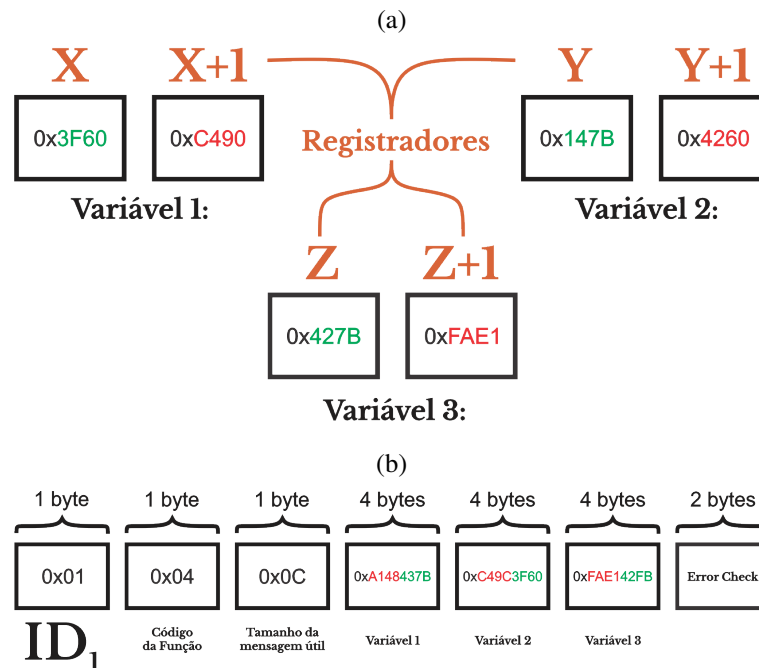
Por padrão, o *slave* mantém os 2 parâmetros iniciais da mensagem do *master*, logo estará contido nos dois primeiros bytes do quadro MODBUS de resposta: o endereço do dispositivo (ID_1) e o código da função (0x04). Na próxima posição do *array*⁷ será preenchido por um parâmetro de 1 byte que representa a quantidade de bytes de carga útil (payload) ou pode ser entendido como o tamanho em bytes da mensagem útil.

A carga útil e o *Error Check* do quadro MODBUS de resposta vem em sequência. A Fig. 20a mostra como as variáveis são armazenadas nos registradores do dispositivo que recebe as requisições (*master*), considerando que as 3 variáveis que serão transmitidas são: 251.63 – 0x437BA148, 0.878 – 0x3F60C49C e 125.99 – 0x42FBFAE1. A Fig. 20b mostra o formato do quadro MODBUS de resposta, em que é possível notar como os 2 bytes mais significativos estão invertidos em relação aos menos significativos. Isto ocorre por que a leitura dos registradores é feita da direita para a esquerda.

Dessa forma, para que a mensagem seja enviada de forma correta, antes é necessário realizar um tratamento destes dados.

⁷ pode-se entender um pacote como um *array* de bytes

Figura 20 – Esquema de transmissão *slave-master* MODBUS (a) Variáveis Armazenadas nos Registradores do *slave*; (b) Variáveis transmitidas para o *master*



Fonte: Elaborado pelo Autor

3.5.9.3 O Hardware *Slave* MODBUS

Os dispositivos utilizados como *Slave* na comunicação MODBUS, utilizados neste trabalho, foram desenvolvidos por Souza (2023) e pelo LASSE. Como é utilizado em duas aplicações

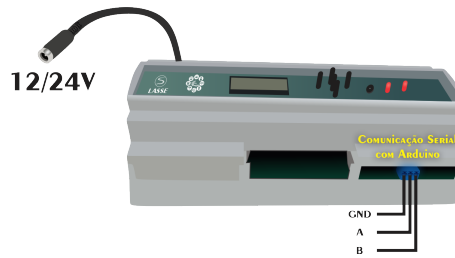
Para extrair informações das aplicações **Sistema Fotovoltáico** e **Sistema de Baterias** e disponibilizar tais informações para o Arduino foram desenvolvidos por Souza (2023) e pelo LASSE as Unidades de Aquisição de Corrente e Tensão (UACTs). Esta comunicação se dá por meio do protocolo MODBUS em que o Arduino é o dispositivo *master* e a UACT é o *slave*.

O primeiro dispositivo *slave* denominado UACT CC, desenvolvido pelo LASSE consiste em realizar medição e armazenamento de tensão e corrente no domínio contínuo do tempo, a sua alimentação também é contínua (12-24 volts). Para comunicação com o Arduino são utilizadas três portas: GND, A e B, sendo as duas últimas portas utilizadas para se comunicar com o RS485. A Fig. 21 ilustra o UACT CC desenvolvido pelo LASSE.

Já o segundo dispositivo foi desenvolvido por ??, para medição e armazenamento de variáveis de tensão e corrente trifásicas (UACT 3F) foi desenvolvido por Souza (2023). Sua alimentação é de 127 volts alternados e também possui a mesma interface de comunicação com o Arduino que o UACT CC. Este dispositivo é ilustrado na Fig. 22.

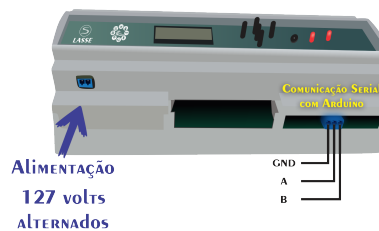
O dispositivo *master* que foi desenvolvido nesta dissertação será detalhado no capítulo de Metodologia.

Figura 21 – Unidade de Aquisição de Corrente e Tensão - Contínua



Fonte: Elaborado pelo Autor com base no dispositivo real desenvolvido pelo LASSE

Figura 22 – Unidade de Aquisição de Corrente e Tensão - Trifásica

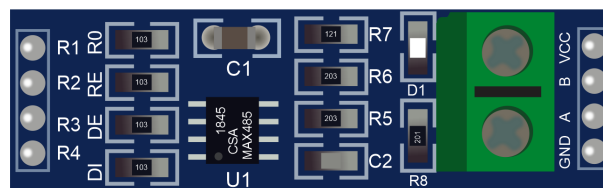


Fonte: Elaborado pelo Autor com base no design real do dispositivo desenvolvido por Souza (2023)

3.5.9.4 Interface RS485 Como Meio de Comunicação Entre Dispositivos MODBUS

O RS485 um periférico utilizado para intermediar comunicações seriais, que neste trabalho ocorre entre o Arduino e o UACT. Permitindo comunicação bidirecional, o RS485 tem como característica principal a comunicação diferencial balanceada e isto possibilita maior imunidade a ruídos externos (JIA; GUO, 2010), (CAMPOS, 2002). Este periférico, ilustrado na Fig. 23 foi utilizado para intermediar a comunicação entre Arduino e a Unidade de Aquisição de Corrente e Tensão, discutida no próximo tópico.

Figura 23 – RS485



Fonte: Elaborado pelo Autor

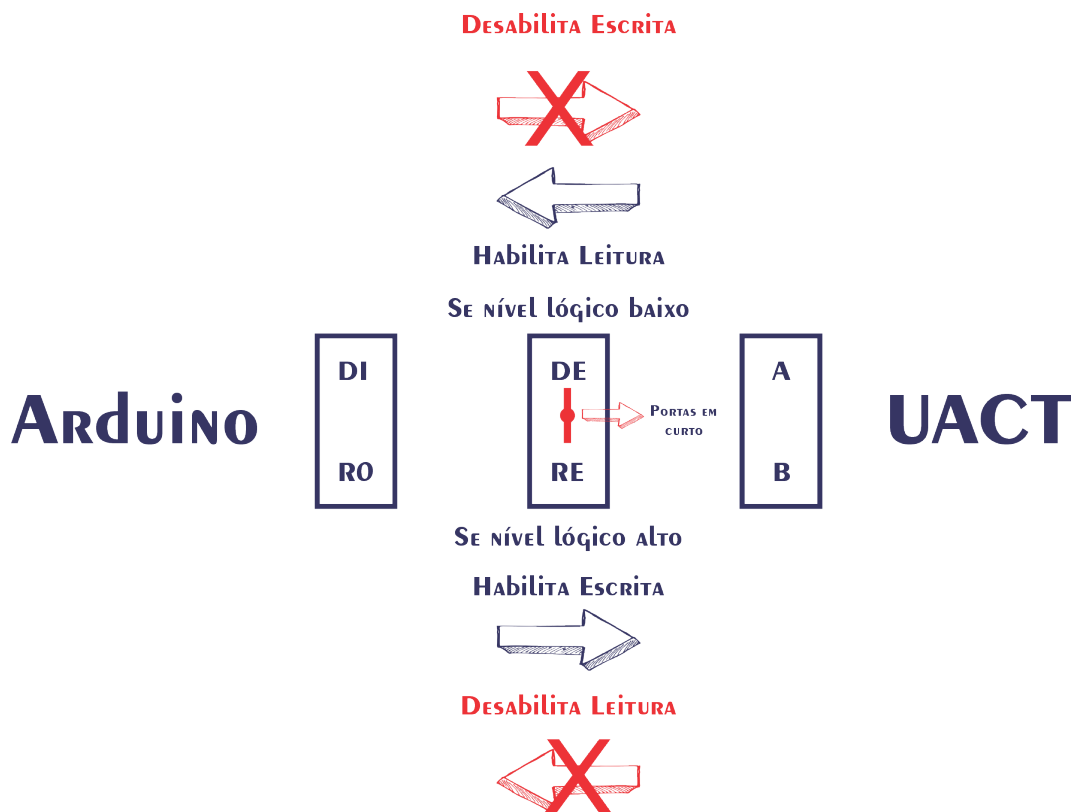
O RS485 possui ao todo 8 portas que possuem as seguintes funções:

- **A e B:** Utilizadas para comunicação entre o UACT e o RS485.
- **GND E VCC:** São respectivamente o aterramento e a alimentação de 5 volts que devem ser conectadas ao Arduino.

- **DE e RE:** São portas que alternam entre leitura (nível lógico alto) e escrita (nível lógico baixo). E para que este método funcione, tais portas devem ser curto-circuitadas e conectadas a alguma porta do Arduino para controle de leitura e escrita.
- **DI e R0:** Comunicação Serial do Arduino com o RS485.

A Figura 24 ilustra como ocorre o processo de leitura e escrita e suas relações com as portas do RS485, em que é possível observar a alternância entre essas duas funções.

Figura 24 – Configuração de leitura e escrita no RS485

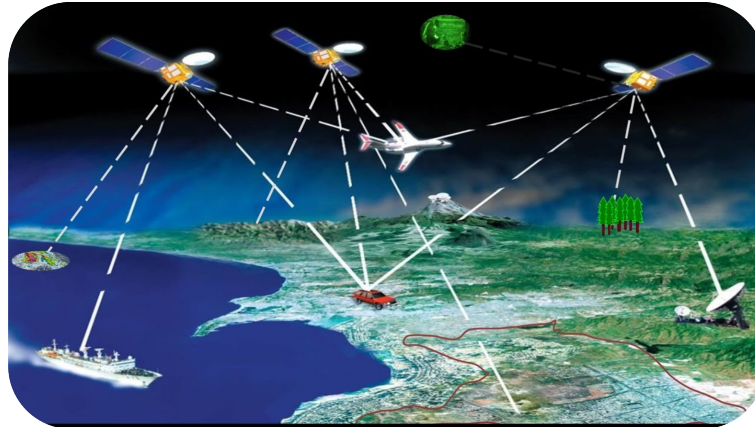


Fonte: Elaborado pelo Autor

3.5.10 O Sistema de Posicionamento Global

Do inglês *Global Position System*, a tecnologia GPS foi criada em 1972 pelo departamento de defesa dos Estados Unidos com a finalidade de desenvolver um sistema de posicionamento capaz de monitorar continuamente a localização de pessoas e objetos em tempo real. (MONICO, 2007). O GPS utiliza vários satélites e antenas que atuam em conjunto para determinar a localização desejada, em latitude e longitude, como é ilustrado na Fig. 25.

Figura 25 – Ilustração de como funciona o GPS

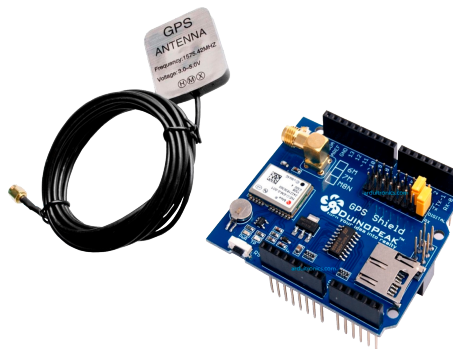


Fonte: <<https://www.youtube.com/watch?v=EDVfF8igHE0&t=188sd>>

3.5.10.1 O Hardware GPS

Para aplicações que monitoram a geolocalização foi utilizado o GPS *Shield* DUINOPEAK, visto na Fig. 26. Atravé deste shield também é possível acessar informações de hora e data em que a geolocalização foi capturada. O controle é feito pelo Arduino por comunicação Serial, através da biblioteca **TinyGPS**.

Figura 26 – GPS Shield DUINOPEAK



Fonte: Retirado do encurtador: <bityli.com/q9PtI>

3.6 Tipos de Variáveis

Para melhor compreensão da metodologia adotada para tratamento de dados deve-se entender os conceitos relacionados aos tipos de variáveis e deslocamento de bits. Os tipos de variáveis estão relacionadas a natureza do conteúdo que será alocado em sua memória. Os tipos de variáveis utilizadas neste trabalho são:

- **int** - 2 bytes: Valores inteiros entre 32.768 e 32.767;

- **float** - 4 bytes: Valores com ponto flutuantes (que possuem casas decimais) entre $-3,40 \times 10^{38}$ e $3,40 \times 10^{38}$
- **byte** - 1 byte: Variável inteira que assume valores de 0 a 255;

Para economizar memória, é de suma importância que os dados sejam transmitidos em bits. Por isso os institutos ANSI (*American National Standard Institute*) recomendam o padrão IEEE754 que se refere as normas de representação e armazenamento de dados de números com ponto flutuante. A utilização desta norma na transmissão de dados permite maior portabilidade com os *softwares* numéricos, haja vista que os computadores atuais juntamente com os compiladores e boa parte de programas utilizam este padrão para processar essas informações (KAHAN, 1996)

De acordo com o tipo de variável o padrão IEEE754 apresenta uma metodologia diferente para conversão de decimal para binário. Variáveis do tipo float são consideradas de precisão simples (4 bytes ou 32 bits) enquanto que double são de precisão dupla e ocupam 8 bytes ou 64 bits. Para as variáveis do tipo float deste trabalho serão utilizados este padrão.

3.7 A DOJOT

Criada em 2017 pelo Centro de Pesquisa e Desenvolvimento em Telecomunicações (CPqD), a Dojot é uma plataforma brasileira *open source* criada com o objetivo de desenvolver tecnologia para *smart cities*. Seu foco principal está nos pilares: **segurança pública, mobilidade urbana e saúde** (CPQD, 2017). Dentre as diversas funcionalidades da Dojot podem ser citadas:

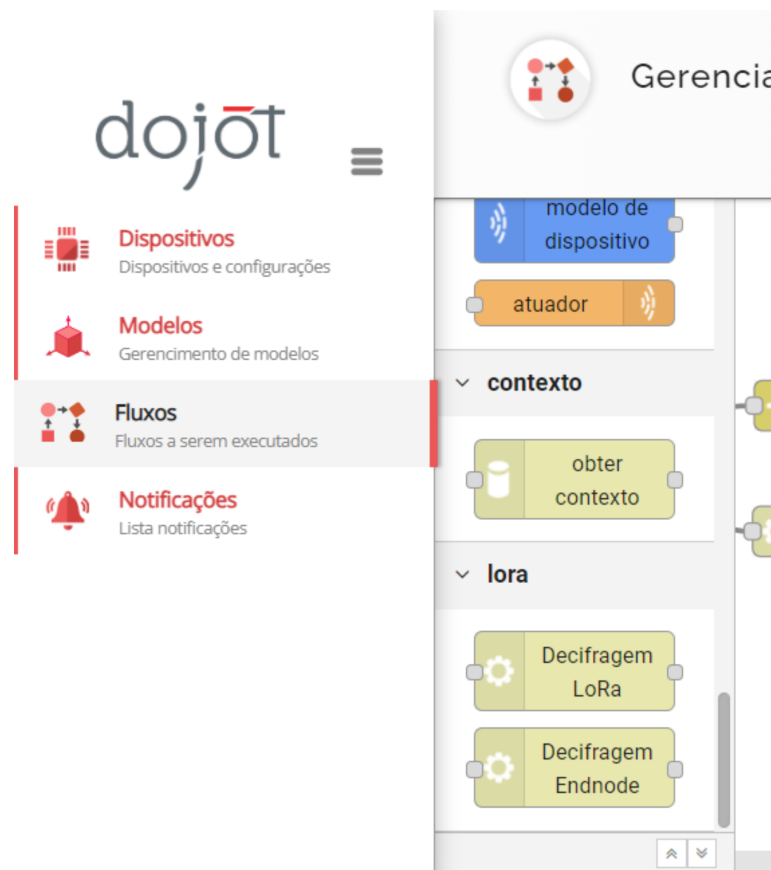
- APIs abertas facilitando sua utilização pelo desenvolvedor;
- Gerenciamento do ciclo de vida de dispositivos através do monitoramento de dados;
- Construção de fluxos (*flowbroker*) de dados e regras de forma visual para processamento de dados em tempo real;
- Monitoramento de aplicações móveis;
- Um banco de dados contendo o histórico dos dados de todas as aplicações para análises *offline*.

Neste projeto a Dojot foi instalada em um computador localizado no CEAMAZON para que este atue como receptor dos dados na camada de aplicação.

3.7.1 O Flowbroker

O *flowbroker* é o dispositivo responsável por fornecer os mecanismos de construção de fluxo de processamento de dados, em que é possível criar regras para gerenciamento e transmissão de dados. Lobato (2022), em seu trabalho desenvolve uma versão do flowbroker mais customizada com a criação de diversas funções, em que se destaca a **Decifragem LoRa**, responsável pela descryptografia AES128, como pode ser observado na Fig. 27.

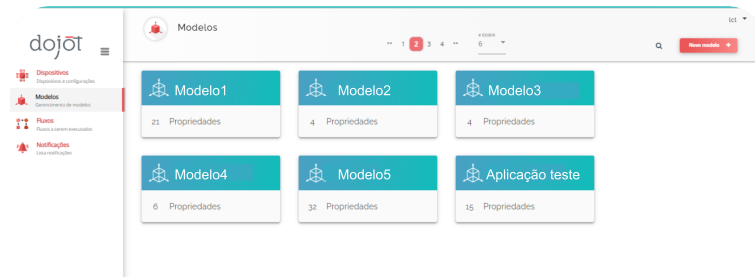
Figura 27 – Flowbroker da Dojot com a criação do bloco **Decifragem LoRa**



Fonte: Elaborado pelo Autor

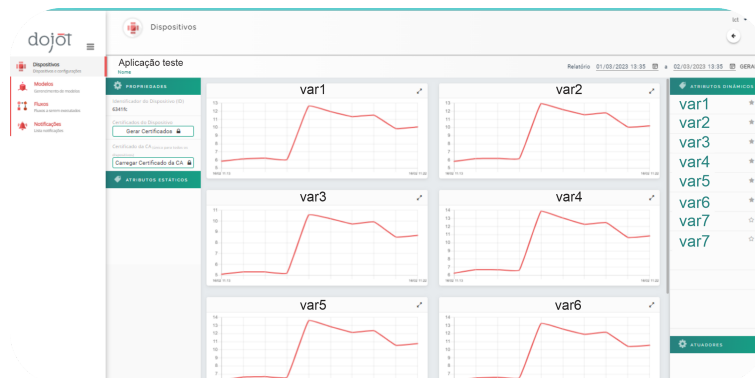
Além do flowbroker, destacam-se outras duas janelas na Fig. 27: a de modelos cuja função consiste em criar a lista de variáveis que cada dispositivo receberá para criação dos dispositivos (Fig. 28); a de dispositivos responsável por criar uma representação digital de cada aplicação para visualização de seus atributos individualmente em gráficos (Fig. 29). Tais dispositivos podem ser end-nodes, gateways ou dispositivos virtuais (LOBATO, 2022; CPQD, 2017).

Figura 28 – Ilustração de um dispositivo com suas variáveis sendo monitoradas individualmente



Fonte: Elaborado pelo Autor

Figura 29 – Ilustração de um dispositivo com suas variáveis sendo monitoradas individualmente



Fonte: Elaborado pelo Autor

3.8 Conclusão do Capítulo

Neste capítulo foram discutidos os conceitos primordiais para o entendimento deste trabalho, bem como a descrição dos diferentes protocolos utilizados nesta dissertação. É notório afirmar sobre a importância de todas essas tecnologias para aplicações IoT que podem possuir diversas aplicações com diferentes requisitos de implementação.

4 Metodologia

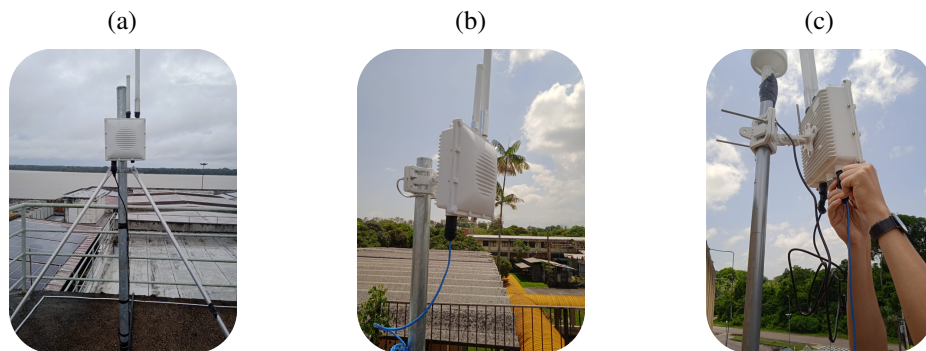
4.1 Conteúdo do Capítulo

Neste capítulo serão descritas as metodologias de transmissão de cada aplicação do projeto SIMA, desde a arquitetura de comunicação como os diferentes algoritmos e tratamento dos dados.

4.2 Instalação dos Gateways

Com a finalidade de cobrir todo o campus, foram instalados 3 gateways (RAK7249) na UFPA, sendo um no mirante do rio (Fig. 30a), um no prédio do LEEB (Fig. 30b) e por último um no prédio do CEAMAZON (Fig. 30c). A princípio os gateways instalados tem por objetivo capturar informações da aplicação do *Ônibus Circular* que transita no campus, porém a proposta é escalável, já que novas aplicações LoRa podem ser desenvolvidas.

Figura 30 – Instalação de gateways no *smart campus* da UFPA (a) Mirante; (b) Prédio do LEEB; (c) Prédio do CEAMAZON

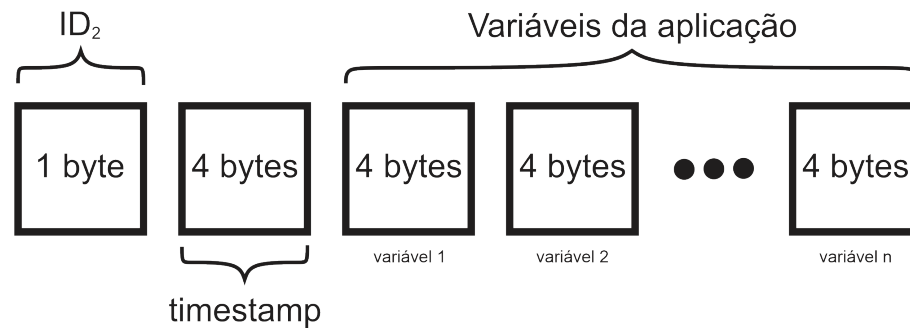


Fonte: Elaborado pelo Autor

4.3 Proposta para Serialização dos Dados

A serialização desenvolvida nesta dissertação consiste em concatenar as variáveis da aplicação em um único *array*, diminuindo dessa forma o *time-on-air* devido a existência de um único preâmbulo. Desta forma utiliza-se o primeiro *byte* deste vetor para a identificar a aplicação (chamaremos de ID_2), para que o *flowbroker* reconheça os dispositivos virtuais que estão relacionados às respectivas aplicações. Os 4 bytes seguintes são formados pelo *timestamp*, variável que contém informações de hora e data que o dado foi gerado e posteriormente seguem as variáveis da aplicação específica (Fig. 31).

Figura 31 – Serialização de Dados Transmitidos



Fonte: Elaborado pelo Autor

Porém é importante destacar que esta serialização é utilizada apenas para transmissões que utilizam a LoRa. Na transmissão do **Ônibus Intermunicipal** que utiliza o GSM, os dados são serializados em json e são transmitidos diretamente para os dispositivos virtuais (como será discutido posteriormente).

4.4 Aplicações Móveis

As aplicações móveis do projeto SIMA já implementadas atualmente são o **Ônibus Intermunicipal** e o **Ônibus Circular**, porém até o término deste ano está previsto a criação do **Barco Elétrico**. A princípio o objetivo é realizar o monitoramento da geolocalização de cada uma dessas aplicações, portanto elas possuem entre outras tecnologias a GPS.

4.4.1 Ônibus Circular

O **Ônibus Circular** do projeto SIMA (Fig. 32) é um veículo que funciona a base de eletricidade, criado para diminuir o custo de energia com combustível e dessa forma promover a sustentabilidade. Seu objetivo é transitar dentro do campus da UFPA - Belém e ajudar os alunos a se locomoverem dentro da universidade.

4.4.1.1 Arquitetura de Transmissão do Ônibus Circular

A arquitetura de transmissão é mostrada na Fig. 33 onde é possível observar diversos componentes necessários para a transmissão.

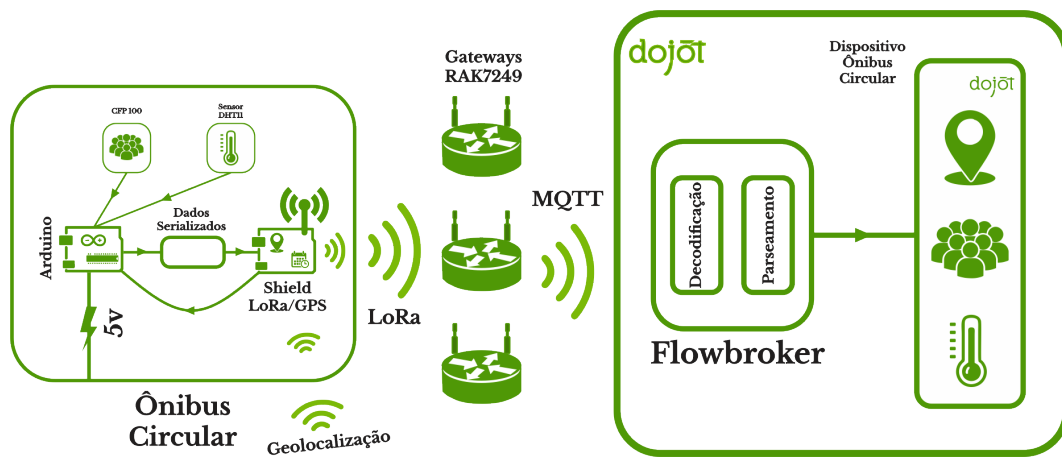
O end-node, desenvolvido por Silva (2022), é composto pelo Arduino, sensor de temperatura (DHT11), dispositivo contador de pessoas (CFP 100), e shield LoRa/GPS. O Arduino captura informações de: temperatura; contagem de pessoas total e atual, geolocalização e *timestamp* (módulo GPS presente no *shield* LoRa/GPS); realiza a serialização, e envia para um dos Gateways por LoRa (módulo LoRa presente no *shield* LoRa/GPS). Um ou mais gateways que recebem a informação e retransmitem para o servidor (Dojot).

Figura 32 – Ônibus circular elétrico



Fonte: Elaborado pelo Autor

Figura 33 – Arquitetura de transmissão do Ônibus Circular



Fonte: Elaborado pelo Autor

4.4.2 Serialização do Payload para o Ônibus Circular

A Tab. 3 contém as informações transmitidas do **Ônibus Circular** de acordo com a metodologia descrita em 4.4.2.1. O payload do **Ônibus Circular** é formado por 13 bytes, e seguindo a Tabela 2, verifica-se que o SF ideal é o SF9, pois é o parâmetro com o maior alcance que suporta transportar essa quantidade de bytes.

A **quantidade de passageiros** indica quantas pessoas estão no interior do veículo, enquanto que a quantidade acumulada se refere ao número de pessoas que entraram no ônibus tomando como período 1 dia.

4.4.2.1 Fluxograma do transmissor Ônibus Circular

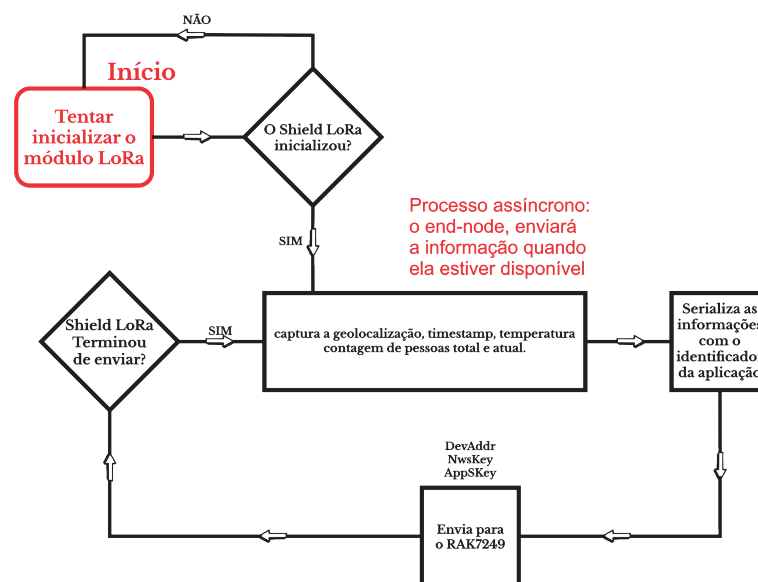
Neste algoritmo o Arduino tenta inicializar o *shield* LoRa, assim que obtiver sucesso nesta etapa, obtem-se a geolocalização, *timestamp*, temperatura, contagem de pessoas, e então

Tabela 3 – Serialização do Payload para o Ônibus Circular

Variáveis	Tipo	Quantidade de Bytes
ID da aplicação	byte	1 byte
Timestamp	float	4 bytes
Temperatura	float	4 bytes
Geolocalização	float	4 bytes
Quantidade de Passageiros	int	2 bytes
Quantidade de Passageiros Acumulados	int	2 bytes

Fonte: Elaborado pelo Autor

realiza a serialização dessas informações (Conforme) a fim de transmiti-las para um dos gateways instalados no *smart campus* da UFPA. O Fluxograma do código no Arduino é mostrado na Fig. 34.

Figura 34 – Fluxograma do end-node da aplicação *Ônibus Circular*

Fonte: Elaborado pelo Autor

4.4.3 Ônibus Intermunicipal

Diferente do ônibus elétrico que circula na UFPA, o **Ônibus Intermunicipal** (Fig. 35), também elétrico, transita em uma rota que contém mais de 77Km¹, com o objetivo de ajudar os alunos que residem em belém mas estudam na UFPA de castanhal e vice versa. Esta distância é maior que o alcance suportado pela tecnologia LoRa (2km para áreas urbanas). Desta forma foi pesquisada uma tecnologia com capacidade de transmissão entre cidades, e por isso a GSM/GPRS foi escolhida já que com ela é possível utilizar a rede de telefonia móvel para transmissão de dados.

¹ segundo o google maps

Figura 35 – Ônibus elétrico intermunicipal

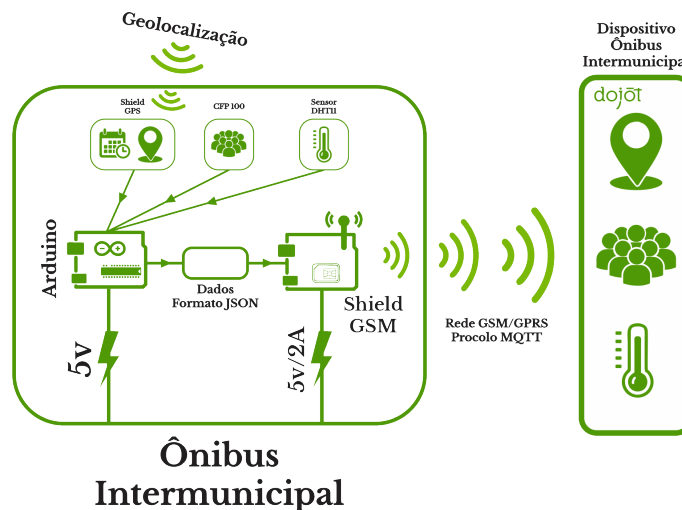


Fonte: Elaborado pelo Autor

4.4.3.1 Arquitetura de transmissão do Ônibus Intermunicipal

Sua arquitetura desenvolvida por Luz (2022) se baseia na transmissão da geolocalização utilizando a rede GSM/GPRS e o protocolo MQTT para estabelecer comunicação com a Dojot. Desta forma um gateway não é necessário já que a própria tecnologia GPRS pode se conectar à internet e transmitir os dados pelo protocolo MQTT, desde que o end-node possua IP e porta da Dojot. A arquitetura é mostrada na Fig. 36.

Figura 36 – Arquitetura de transmissão do **Ônibus Intermunicipal**



Fonte: Elaborado pelo Autor

O end-node captura de geolocalização e *timestamp* (shield GPS DUINOPEAK), temperatura (sensor DHT11) e contagem de pessoas total e atual (CFP 100). Logo em seguida esses dados são estruturados em JSON e são transmitidos pelo shield GSM por MQTT diretamente para o dispositivo virtual **Ônibus Intermunicipal** na Dojot.

4.4.4 Estruturação dos Dados em JSON

As informações do Ônibus Circular são enviadas por MQTT para a Dojot, e diferentemente da serialização proposta em 4.4.2.1, na transmissão do **Ônibus Intermunicipal** as informações são estruturadas em json como mostra a Fig. 37.

Figura 37 – Estruturação dos Dados em JSON

```

json["qtdA"] = cfp.actualIn;
json["qtdT"] = cfp.totalIn;
json["temperatura"] = temperatura;
json["localizacao"] = String(latitude, 5)+" "+String(longitude, 5);
json["timestamp"] = getTimeStamp();
    
```

Fonte: Adaptado de Luz (2022)

de forma que **qtdA** representa a quantidade atual de passageiros, e **qtdT** representa a quantidade total.

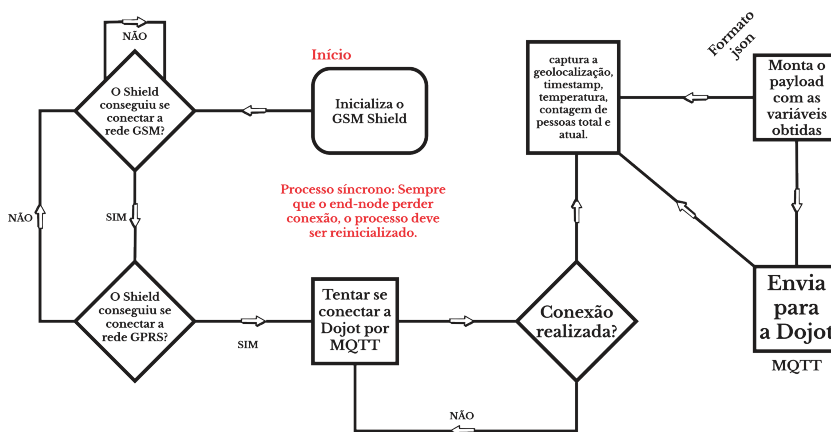
4.4.4.1 Instalação do CFP 100

A configuração deste dispositivo para o **Intermunicipal** foi desenvolvida por Luz (2022) e sua instalação é mais simples, haja vista que este ônibus possui apenas 1 porta dianteira permitindo a passagem de apenas uma fileira de alunos, portanto é necessário que apenas 1 CFP seja instalado.

4.4.4.2 Fluxograma do transmissor Ônibus Intermunicipal

Também desenvolvido por Luz (2022), o código apresenta sequência lógica que envolve mais estágios se comparado com o **Ônibus Circular**. A Fig. 38 ilustra o fluxograma que representa o envio de dados do end-node da aplicação do **Ônibus Intermunicipal**.

Figura 38 – Fluxograma de transmissão do end-node da aplicação **Ônibus Intermunicipal**



Fonte: Elaborado pelo Autor

A sequência do algoritmo segue da seguinte forma: (i) inicialmente o *shield* GSM deve ser inicializado; (ii) então o hardware precisa verificar a conexão na rede GSM e GPRS nessa ordem; (iii) em seguida é necessário realizar conexão a Dojot por meio do protocolo MQTT com os dados de IP e porta do servidor; (iv) posteriormente captura-se a geolocalização e *timestamp* (*shield* GPS DUINOPEAK), temperatura interna (sensor DHT11), e contagem de pessoas (CFP 100), (v) monta o payload no formato JSON; (vi) envia diretamente para o dispositivo virtual. Valendo ressaltar que o processo é síncrono, logo o envio acontecerá enquanto as comunicações GSM/GPRS/MQTT estiverem funcionando.

4.4.4.3 Metodologia de Ativação Automática do Cartão SIM e Reconexão da rede GSM

Para tornar o processo de ativação e reconexão automática da rede GSM foram utilizadas três melhorias nesta dissertação: uma de hardware e 2 de software. No dispositivo SIM900 foi realizado a soldagem do pino R13 que está conectado ao botão utilizado para ativar o cartão SIM manualmente como descrito em 3.5.7.2. Este pino está diretamente conectado a porta 9 do *shield* SIM900 e este foi conectado a porta 9 do Arduino para realizar a alteração de software.

Foi acrescentado uma função que consiste em verificar o LED **status** do *shield* SIM900, através da função **getSimStatus** da biblioteca **TinyGSM**. Se este LED estiver apagado, o cartão SIM está desligado. Logo a porta 9 conectada ao botão do *shield* deve ser setada em nível lógico baixo, alto e baixo respectivamente, simulando a ação de apertar o botão. A Fig. 39 mostra a função criada para ativar o cartão SIM automaticamente.

Figura 39 – Método de ativação automática do cartão SIM

```
void verifica(const int botao){
  if(!modem.getSimStatus(5000)){
    Serial.println("Luz do Status Desligada. O SIM Card está desligado;");
    digitalWrite(botao,LOW);
    delay(1000);
    digitalWrite(botao,HIGH);
    delay(2000);
    digitalWrite(botao,LOW);
    delay(3000);
  }
}
```

Fonte: Elaborado pelo Autor

Quanto a reconexão foi utilizada a biblioteca **SoftwareReset** para reiniciar o microcontrolador. Nesta metodologia, a reinicialização do Arduino ocorre em períodos em que o *shield* GSM perde conexão com a rede, e desta forma todo o processo deve ser reinicializado em ordem: conexão GSM, GPRS e MQTT. Além disso este método permite reinicializar o dispositivo caso ele perca conexão GSM em uma tentativa de conexão com a Dojot por MQTT. Se o processo não for reinicializado nesta etapa, o end-node ficará em loop infinito com a tentativa de se conectar à Dojot sem a rede GSM. Por isto este processo é chamado de transmissão síncrona.

4.5 Aplicações Estáticas

As duas aplicações descritas nestas seções serão o **Sistema Fotovoltáico** e o **Sistema de Baterias** que utilizam as UACTs para medição e transferência de dados via protocolo MODBUS para o Arduino.

4.5.1 Esquema de Conexão do RS485 com a UACT e o Arduino

O esquema de conexão mostrado na Tab. 4 e na Fig. 40 são análogos ao transmissor do **Sistema Fotovoltáico** e do **Sistema de Baterias**. Neste esquema é mostrado como o RS485 se conecta ao Arduino e à UACT, em que é válido ressaltar as portas RE e DE que, como explicado anteriormente, devem ser curtadas e conectadas a uma porta do Arduino. E neste projeto foi escolhida a porta 23. Além disso, a GND da UACT e do RS485 devem ser conectadas à GND do Arduino.

Tabela 4 – Esquema de Conexão do RS485 com a UACT e o Arduino

Portas	Conexão
VCC	VCC do Arduino
GND	GND da UACT e do Arduino
A e B	A e B da UACT
DI (RX)	Porta 18 (TX1) do Arduino
R0 (TX)	Porta 19 (RX1) do Arduino
DE e RE	Porta 23 do Arduino

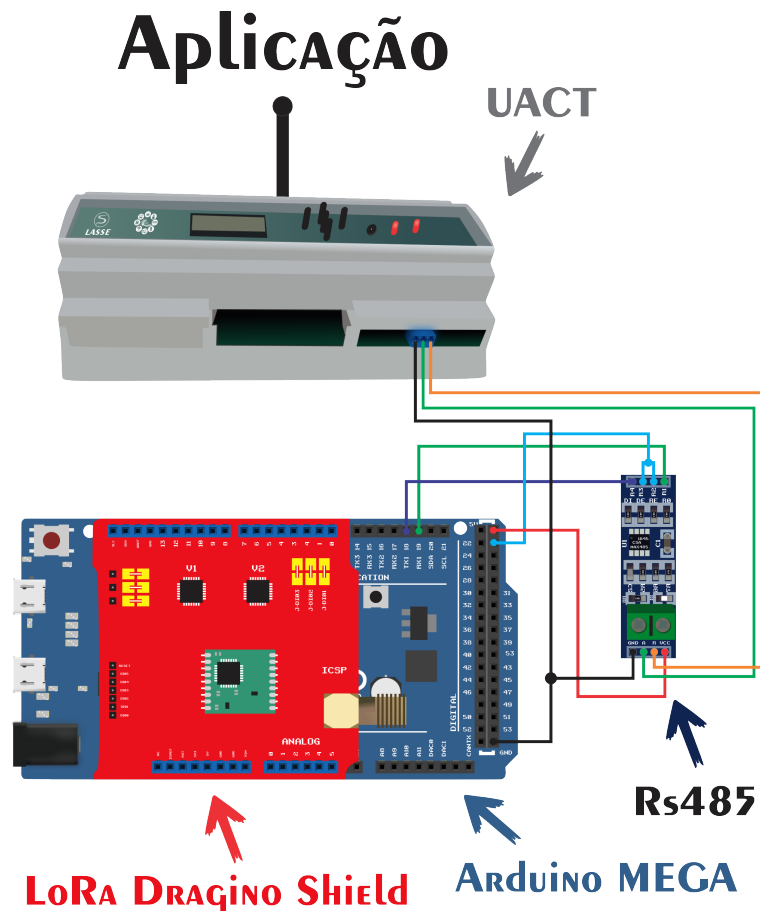
Fonte: Elaborado pelo Autor

4.5.2 Criação da biblioteca MODBUS em C/C++ para Recebimento dos Dados

Devido este trabalho ter muitas especificidades que não são atendidas pelas bibliotecas MODBUS presentes na internet, foi desenvolvida, nesta dissertação, uma biblioteca MODBUS para recebimento dos dados via comunicação serial para estabelecer comunicação entre Arduino e UACT. O código foi desenvolvido através dos conceitos de programação orientada a objetos que possui a classe principal chamada MODBUS com diversos parâmetros para a aplicação do protocolo:

- **EnviarRequisicao:** Método do tipo void que envia a requisição para a UACT. Este método tem os seguintes atributos:
 - **ID:** Variável do tipo *byte* que representa o endereço do dispositivo;
 - **CodigoFuncao:** Variável *byte* que contém o código da função utilizada, que neste trabalho sempre tem o valor hexadecimal **0x04**;
 - **EnderecoInicial:** Variável do tipo *uint16_t* de 2 bytes que representa o endereço inicial do registrador que deseja-se acessar;

Figura 40 – Esquema de Conexão do RS485 com a UACT e o Arduino



Fonte: Elaborado pelo Autor

- **QuantidadeRegistradores:** Também do tipo *uint16_t* que representa a quantidade de registradores lidos a partir do registrador inicial.
- **ErrorCheck:** Método do tipo *uint16_t* que retorna o *Error Check* de 2 bytes;
- **ValidacaoPacote:** Método booleano que utiliza o **ErrorCheck** para verificar a integridade do pacote.
- **pinoRS485Comunicacao:** Variável do tipo *int* que conterá a porta Arduino que estará conectada as entradas **RE** e **DE** do RS485 para controle de leitura e escrita.

Os projetos dos transmissores do **Sistema Fotovoltáico** e **Sistema de Baterias** estão disponíveis, respectivamente nos links: <github.com/joelengdesign/TransmissorSistemaFotovoltaiico> e <github.com/joelengdesign/TransmissorSistemaDeBaterias>. Ambos possuem a biblioteca MODBUS na pasta libs.

4.5.3 Sistema Fotovoltáico

Desenvolvida por Júnior (2022), esta aplicação está situada próxima ao prédio do CEAMAZON (Fig. 41) e como o nome sugere, é um conjunto de placas solares responsáveis por converter a energia do sol em eletricidade. O quadro de energia do **Sistema Fotovoltáico** está situado no interior do eletrocentro.

Figura 41 – Sistema Fotovoltáico



Placas Fotovoltáicas

Fonte: Elaborado pelo Autor

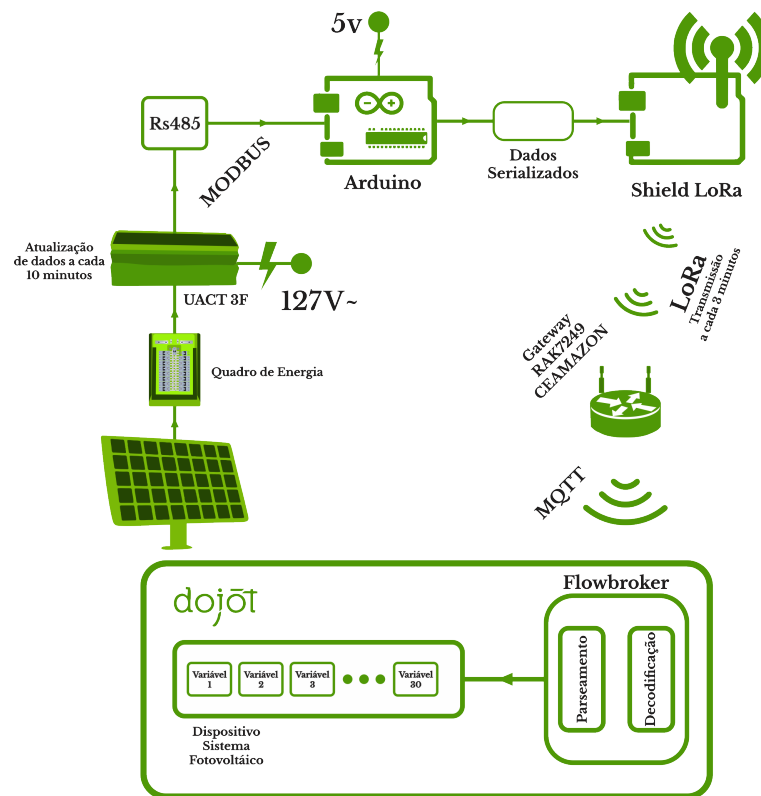
4.5.3.1 Arquitetura de Transmissão do Sistema Fotovoltáico

A arquitetura de transmissão proposta para o **Sistema Fotovoltáico** possui um dispositivo que atua como sensor, enviando dados para o Arduino, como mostra a Fig. 42, em que é possível verificar que o UACT 3F realiza a medição e armazena os dados (SOUZA, 2023). O Arduino acessa essas variáveis através do protocolo MODBUS com o RS485 intermediando os dispositivos (UACT e Arduino); Uma vez que as informações são requisitadas e tratadas, são serializadas conforme 4.4.2.1; O end-node envia as informações via LoRa (Shield LoRa) para o Gateway instalado no CEAMAZON por ser o imediatamente mais próximo da aplicação. No servidor a informação é decodificada no *flowbroker* e enviada para o dispositivo virtual do Sistema Fotovoltáico.

4.5.3.2 Serialização do Payload para o Sistema Fotovoltáico e Algoritmo do End-Node

Nesta aplicação do sistema fotovoltaico, desconsiderando os harmônicos que compõe outra aplicação do *smart campus*, há 30 variáveis mais o *timestamp*, o que totaliza 125 bytes. Para este tamanho de payload optou-se pelo SF8 pois sua capacidade de armazenamento é exatamente 125 bytes (Tabela 2) e o alcance maior que o SF7.

Figura 42 – Arquitetura de Transmissão do Sistema Fotovoltáico



Fonte: Elaborado pelo Autor

A Tab. 5 mostra a serialização das variáveis para o **Sistema Fotovoltáico**, mostrando o endereço de cada uma delas nos registradores da UACT 3F. Dessa forma, foi proposta uma metodologia para aquisição desses dados utilizando o MODBUS, realizando 1 requisição de 60 (0x3C) registradores à UACT 3F.

Tabela 5 – Serialização do Payload para o Sistema Fotovoltáico

Variável	Tipo	Quantidade de Bytes	REGISTRADORES
ID da Aplicação	byte	1 bytes	-
<i>Timestamp</i>	float	4 bytes	0004 e 0005 Início da primeira requisição
Informação não utilizada	-	-	0002 a 000B
Tensão RMS (Fase A)	float	4 bytes	000C e 000D
Tensão RMS (Fase B)	float	4 bytes	000E e 000F
Tensão RMS (Fase C)	float	4 bytes	0010 e 0011
Corrente RMS (Fase A)	float	4 bytes	0012 e 0013
Corrente RMS (Fase B)	float	4 bytes	0014 e 0015
Corrente RMS (Fase C)	float	4 bytes	0016 e 0017
Corrente RMS (neutro)	float	4 bytes	0018 e 0019
Frequência (Fase A)	float	4 bytes	001A e 001B
Frequência (Fase B)	float	4 bytes	001C e 001D
Frequência (Fase C)	float	4 bytes	001E e 001F Fim da primeira requisição
Potência Ativa (Fase A)	float	4 bytes	0020 e 0021 Início da segunda requisição
Potência Ativa (Fase B)	float	4 bytes	0022 e 0023
Potência Ativa (Fase C)	float	4 bytes	0024 e 0025
Potência Ativa (Total)	float	4 bytes	0026 e 0027
Potência Reativa (Fase A)	float	4 bytes	0028 e 0029
Potência Reativa (Fase B)	float	4 bytes	002A e 002B
Potência Reativa (Fase C)	float	4 bytes	002C e 002D
Potência Reativa (Total)	float	4 bytes	002E e 002F
Potência Aparente (Fase A)	float	4 bytes	0030 e 0031
Potência Aparente (Fase B)	float	4 bytes	0032 e 0033
Potência Aparente (Fase C)	float	4 bytes	0034 e 0035
Potência Aparente (Total)	float	4 bytes	0036 e 0037
Fator de Potência (Fase A)	float	4 bytes	0038 e 0039
Fator de Potência (Fase B)	float	4 bytes	003A e 003B
Fator de Potência (Fase C)	float	4 bytes	003C e 003D
Fator de Potência (Total)	float	4 bytes	003E e 003F Fim da segunda requisição
Consumo (Fase A)	float	1 byte	Variável não disponível
Consumo (Fase B)	float	1 byte	Variável não disponível
Consumo (Fase C)	float	1 byte	Variável não disponível
Consumo (Total)	float	1 byte	Variável não disponível

Fonte: Elaborado pelo Autor

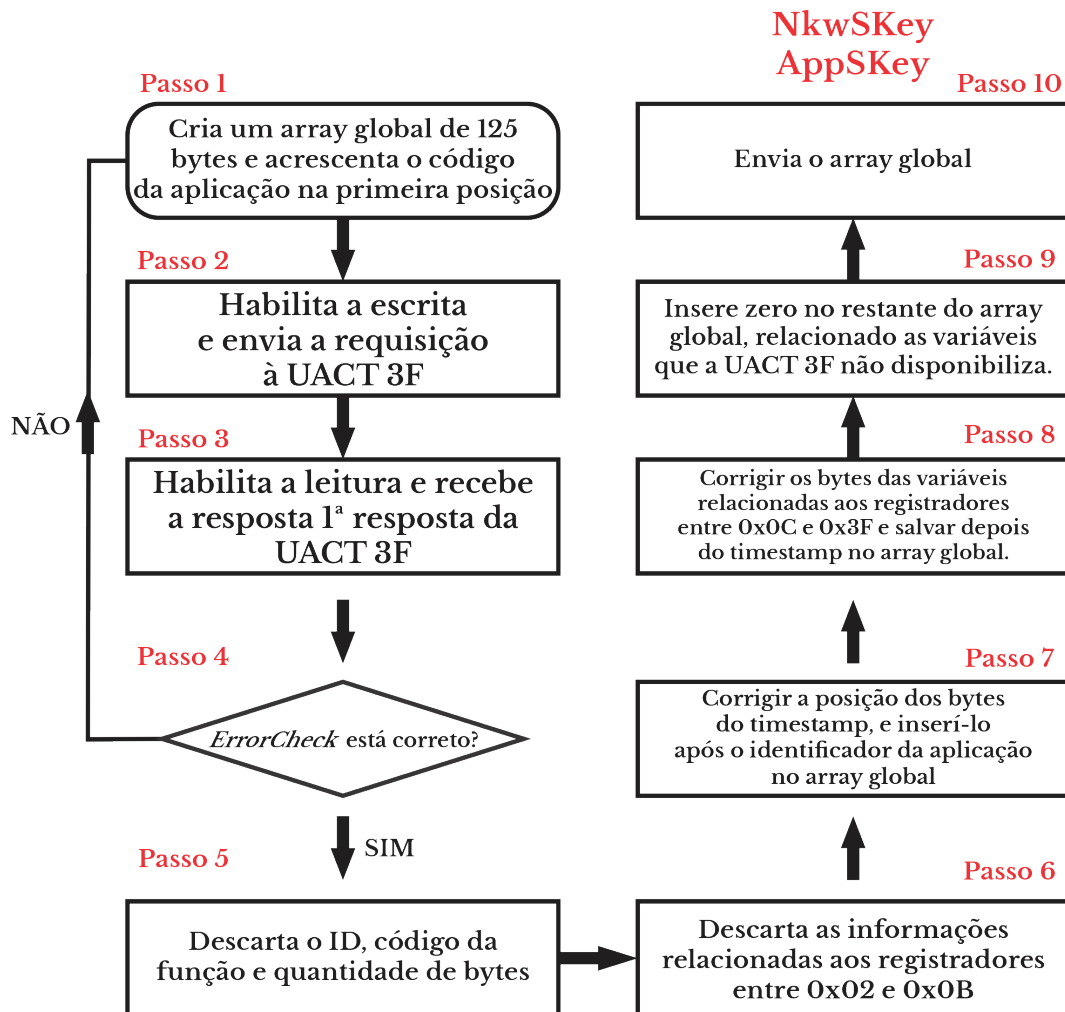
O algoritmo é descrito em 8 passos (também ilustrado na Fig. 43):

- **Passo 1:** Inserir o código da aplicação (ID_2) no início do pacote que representa o código do **Sistema Fotovoltáico** na Dojot - 1ª parte do pacote;
- **Passo 2:** Habilita a escrita do RS485 e envia a primeira requisição a UACT CC:
 - **ID₁**: identificador do dispositivo - **0x01** para a UACT CC;
 - **Function Code**: Código da função - **0x04**;
 - **Init Address**: **0x04** (Registrador inicial com endereço **4**);
 - **Quantity Registers**: **0x3C** (60 registradores a partir do **4**).
- **Passo 3:** Habilita a leitura do RS485 e recebe os dados;
- **Passo 4:** O *Error Check* está correto?
 - **SIM**: armazena os dados em uma variável auxiliar e prossegue para o passo 5;
 - **NÃO**: Retorna ao passo 2.
- **Passo 5:** Descarta o ID (ID_1), código da função e quantidade de bytes do quadro MODBUS de resposta;
- **Passo 6:** Descarta as informações dos registradores entre **0x06** e **0x0B**, pois não são usadas neste trabalho;
- **Passo 7:** Armazena o *timestamp* depois de corrigir a posição dos bytes dos registradores **0x00** e **0x01** - Indica a 2ª parte do pacote.
- **Passo 7:** Corrigir a posição dos bytes, variável por variável (4 em 4 bytes) que estão contidas nos registradores entre **0x0C** e **0x3F** e salvar - Indica a 3ª parte do pacote;
- **Passo 8:** Acrescentar **0** nas variáveis: **Consumo Fase A**, **Consumo Fase B**, **Consumo Fase C** e **Consumo Total** - Indica a última parte do pacote;
- **Passo 8:** Com o pacote criptografado (AES128) devidamente montado, o end-node deve enviar para o RAK7249 por LoRa.

4.5.3.3 Método de Agragação de Dados

As variáveis do **Sistema Fotovoltáico** tensão e corrente são alternadas, porém para a aplicação foi estabelecido que os dados transmitidos seriam os valores eficazes relacionados a forma de onda de cada fase (A, B, C e neutro) do circuito trifásico. Souza (2023) desenvolve um método de agragação de tal forma que os cálculos de tensão, corrente e potência são feitos sob

Figura 43 – Fluxograma do transmissor do Sistema Fotovoltáico



Fonte: Elaborado pelo Autor

180 ciclos de forma de onda, e em 10 minutos é realizada a agragação e cálculo desses valores, e durante os outros 10 minutos essas informações ficam disponíveis para o end-node acessá-las por MODBUS.

Sendo assim optou-se por realizar a transmissão desses valores 3 vezes a cada dez minutos para evitar problemas com a eventual perda de pacote.

4.5.4 Sistema de Baterias

O **Sistema de Baterias** caracteriza-se por armazenar a energia excedente e utilizar para alimentar o prédio do CEAMAZON (JÚNIOR, 2022). É composta por 8 baterias, cuja tensão é a mesma em todo barramento, porém cada uma delas possui corrente distinta. A Fig. 44 mostra o **Sistema de Baterias** no interior do eletrocentro.

Figura 44 – Eletrocentro com o **Sistema de Baterias** em destaque

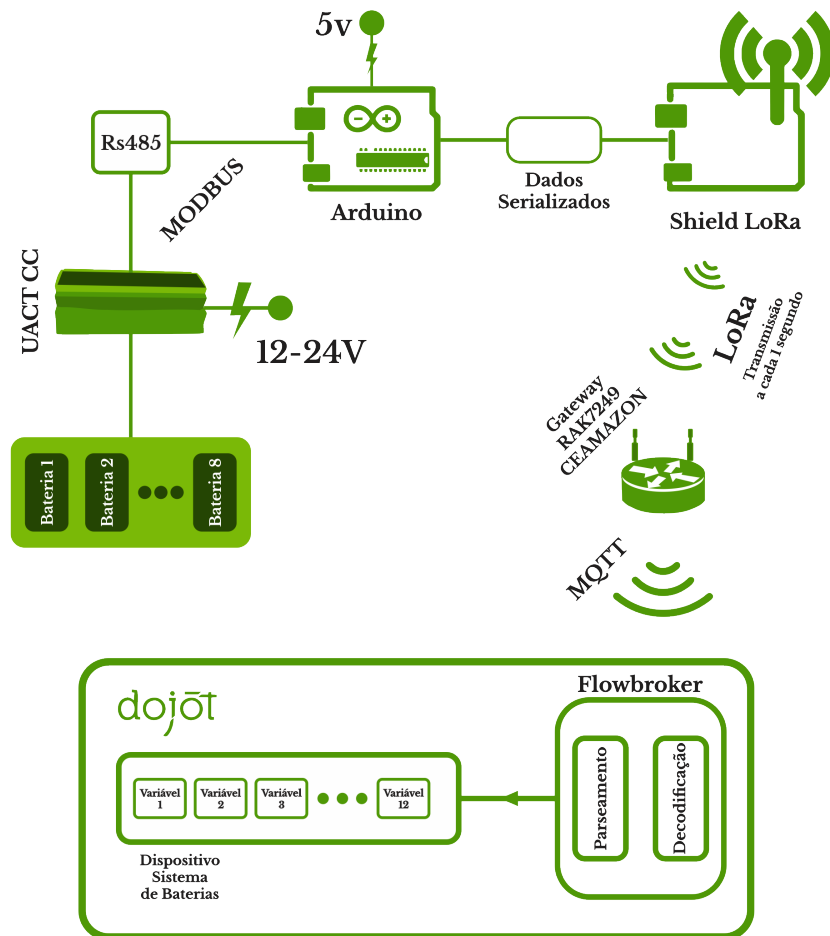


Fonte: Elaborado pelo Autor

4.5.4.1 Arquitetura de Transmissão do Sistema de Baterias

Sua arquitetura (Fig. 45) é análoga ao **Sistema Fotovoltáico**, diferindo pelo número de variáveis e pelo identificador da aplicação.

Figura 45 – Arquitetura de Transmissão do **Sistema de Baterias**



Fonte: Elaborado pelo Autor

4.5.4.2 Serialização do Payload para o Sistema de Baterias e Algoritmo do End-Node

Esta aplicação possui apenas 13 variáveis para monitoramento, porém 3 destas o UACT CC ainda não disponibiliza para envio (Tabela 6), portanto o objetivo é transmitir 53 bytes. E desta forma escolheu-se transmitir utilizando o SF9, pois ele possui o maior alcance que pode alocar essa quantidade de bytes, conforme a Tabela 2. Desta forma, a metodologia consiste em apenas 1 requisição a UACT CC.

Tabela 6 – Serialização do Payload para o Sistema de Baterias

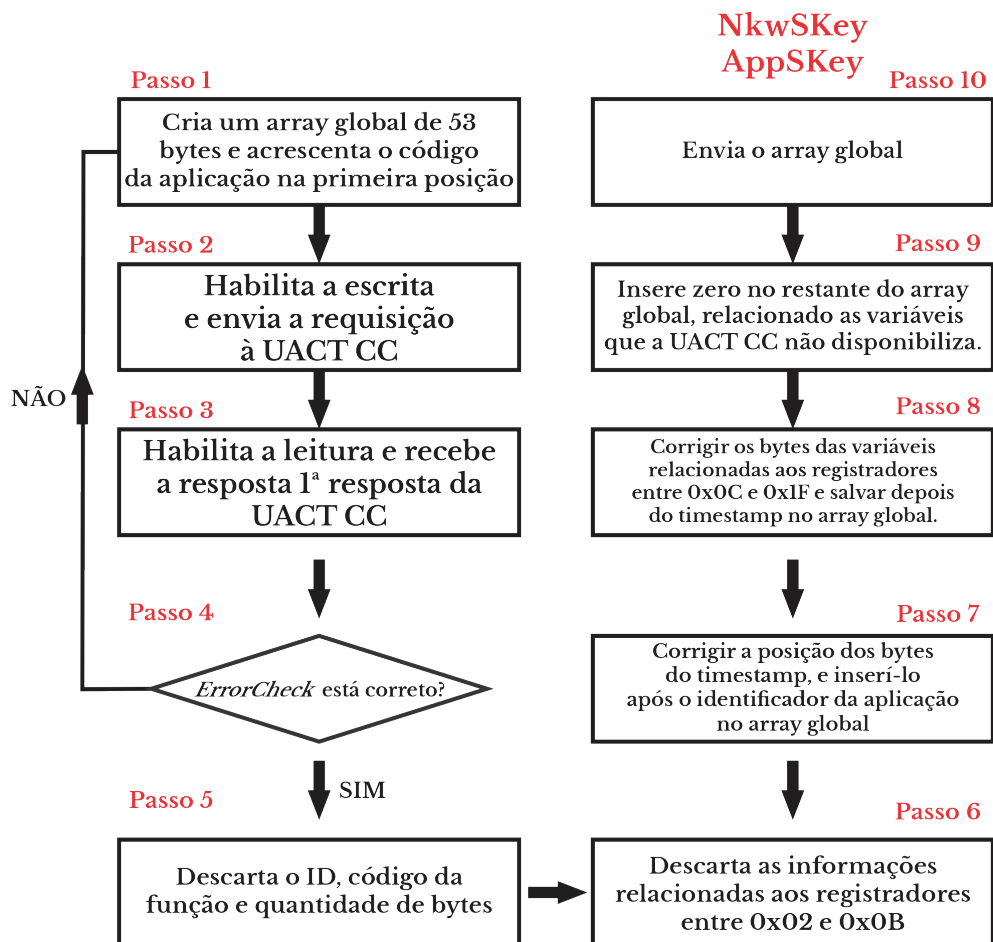
Variável	Tipo	Quantidade de Bytes	REGISTRADORES
ID da Aplicação	byte	1 bytes	-
<i>Timestamp</i>	float	4 bytes	0000 e 0001 Início da requisição
Informação não utilizada	-	-	0002 a 000B
Tensão CC	float	4 bytes	000C e 000D
Corrente CC 1	float	4 bytes	000E e 000F
Corrente CC 2	float	4 bytes	0010 e 0011
Corrente CC 3	float	4 bytes	0012 e 0013
Corrente CC 4	float	4 bytes	0014 e 0015
Corrente CC 5	float	4 bytes	0016 e 0017
Corrente CC 6	float	4 bytes	0018 e 0019
Corrente CC 7	float	4 bytes	001A e 001B 001C e 001D Fim da requisição
Corrente CC 8	float	4 bytes	
Potência CC	float	4 bytes	Variável não disponível
Energia Fornecida	float	4 bytes	Variável não disponível
Energia Consumida	float	4 bytes	Variável não disponível

Fonte: Elaborado pelo Autor

A descrição do algoritmo está descrito logo abaixo e a Fig. 46 também mostra o algoritmo, porém de forma mais esquemática para facilitar o entendimento.

- **Passo 1:** Inserir o código da aplicação (ID_2) no início do pacote, pois representa o código do **Sistema de Baterias** na Dojot - 1ª parte do pacote;
- **Passo 2:** Habilita a escrita do RS485 e envia a primeira requisição a UACT CC:
 - **ID₁**: identificador do dispositivo - **0x01** para a UACT CC;
 - **Function Code**: Código da função - **0x04**;
 - **Init Address**: **0x00** (Registrador inicial com endereço **0**);

- *Quantity Registers*: **0x1E** (30 registradores a partir do **0**).
- **Passo 3**: Habilita a leitura do RS485 e recebe os dados;
- **Passo 4**: O *Error Check* está correto?
 - *SIM*: armazena os dados em uma variável auxiliar e prossegue para o passo 5;
 - *NÃO*: Retorna ao passo 2.
- **Passo 5**: Descarta o ID (ID_1), código da função e quantidade de bytes do quadro MODBUS de resposta;
- **Passo 6**: Descarta as informações dos registradores entre **0x02** e **0x0B**, pois não são usadas neste trabalho;
- **Passo 7**: Armazena o *timestamp* depois de corrigir a posição dos bytes dos registradores **0x00** e **0x01** - Indica a 2ª parte do pacote.
- **Passo 7**: Corrigir a posição dos bytes, variável por variável (4 em 4 bytes) que estão contidas nos registradores entre **0x0C** e **0x1D** e salvar - Indica a 3ª parte do pacote;
- **Passo 8**: Acrescentar **0** nas variáveis: **Potência CC**, **Energia Fornecida**, **Energia Consumida** - Indica a última parte do pacote;
- **Passo 8**: Com o pacote criptografado (AES128) devidamente montado, o end-node deve enviar para o RAK7249 por LoRa.

Figura 46 – Fluxogramad do end-node do **Sistema de Baterias**

Fonte: Elaborado pelo Autor

É importante destacar que tanto no **Sistema Fotovoltáico** quanto no **Sistema de Baterias**, o **end-node atua como gateway**, pois utiliza duas tecnologias distintas para estabelecer comunicações entre duas redes: a rede MODBUS e a rede LoRa.

Além disso, diferente do UACT 3F, o UACT CC não realiza cálculos matemáticos sob tensão e corrente alternadas. Dessa forma é necessário realizar a leitura dos dados constantemente. Assim, optou-se por transmitir os dados do **Sistema de Baterias** a cada 1 segundo para a Dojot.

4.6 Configuração dos End-Nodes

Para transmissão de dados de end-nodes que utilizam LoRa (**Ônibus Circular**, **Sistema Fotovoltáico** e **Sistema de Baterias**), é necessário configurar as chaves da criptografia AES128 o *Spreading Factor*. A Fig. 47 mostra os campos da biblioteca Imic, que são utilizados para configurar a transmissão LoRa e a função de envio.

Para transmissão GSM (**Ônibus Intermunicipal**), foi necessário configurar 5 parâmetros MQTT (Fig. 48) utilizando a biblioteca TinyGSM:

Figura 47 – Conguração do End-Node LoRa

Definição dos parâmetros da criptografia AES128

```
static const PROGMEM u1_t NWKSKEY[16] = {0x12, 0x34, 0x56, 0x18, 0xb4,
static const u1_t PROGMEM APPSKEY[16] = {0x30, 0xa2, 0xac, 0xca, 0xe1,
static const u4_t DEVADDR = 0xa62108A3 ;
```

Criptografia dos dados

```
LMIC_setSession (0x1, DEVADDR, NWKSKEY, APPSKEY);
```

Envio dos dados

```
LMIC_setTxData2(1, payload, sizeof(payload), 0);
```

Configuração do SF e potência de transmissão em dBm

```
LMIC_setDrTxpow(DR_SF7, 14);
```

Fonte: Elaborado pelo Autor

- **IP e porta:** Utilizado para conexão com a Dojot;
- **Tópico MQTT e Usuário MQTT:** Identifica o cliente destinatário que receberá a mensagem no servidor para publicar (*publish*) os dados. Neste trabalho o cliente é o dispositivo virtual **Ônibus Intermunicipal**;
- **Cliente MQTT:** Utilizado para criar um cliente no end-node para conexão MQTT.
- **password:** Senha de acesso ao servidor.

Figura 48 – Conguração MQTT do transmissor GSM

Configuração dos parâmetros MQTT

```
const char* IP =
const uint16_t porta =
const char* mqttClientId =
const char* mqttUser =
const char* mqttPassword =
const char* mqttTopic =
```

Função de conexão com o servidor

```
mqtt.setServer(IP, porta);
```

Publica a informação

```
mqtt.publish(mqttTopic, payload);
```

Função que verifica se a comunicação foi obtida

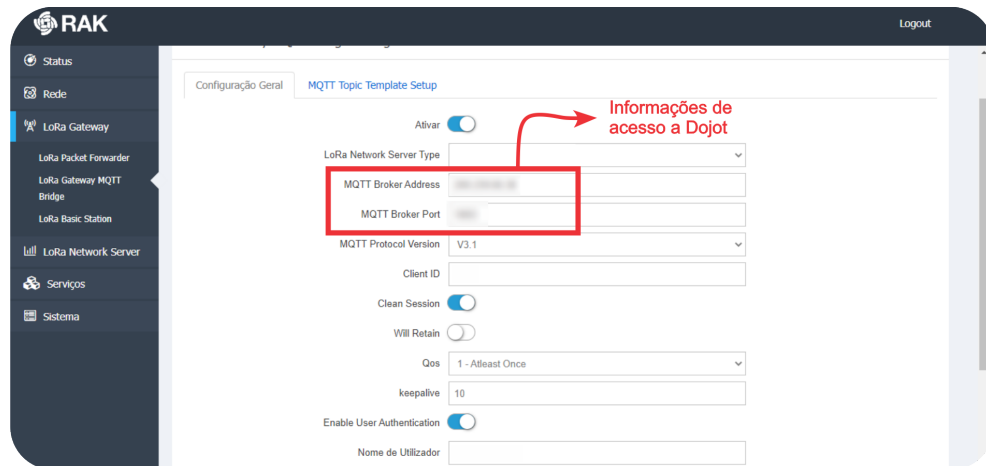
```
boolean status = mqtt.connect(mqttClientId, mqttUser, mqttPassword);
```

Fonte: Elaborado pelo Autor

4.7 Configuração dos Gateways

Para que cada gateway envie os dados para O servidor, é necessário que eles tenham informações de IP e porta da Dojot afim de que estabeleçam comunicação. Desta forma, a Fig. 49 mostra o campo onde se encontra o IP da Dojot (informação confidencial) no RAK7249.

Figura 49 – Configuração de acesso a Dojot



Fonte: Elaborado pelo Autor

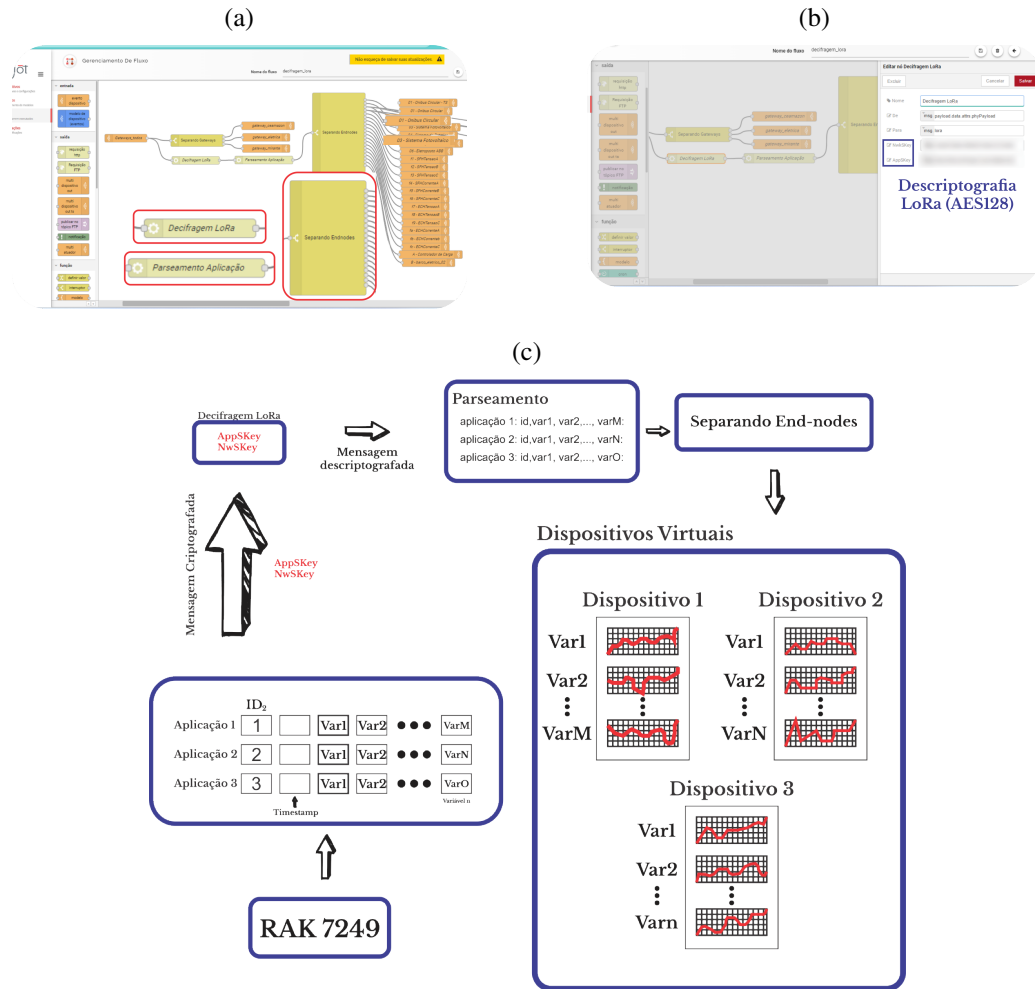
4.8 Configuração da Dojot

Como mencionado no tópico 3.7.1 a Dojot permite a criação de fluxos (*flowbroker*) e a criação de regras para montagem do sistema IoT. Na Fig. 50a é mostrado o gerenciamento de fluxo de todos os dispositivos e tecnologias envolvidos no projeto SIMA (LOBATO, 2022). Em destaque estão os principais blocos do *flowbroker*.

- **Decifragem LoRa:** Responsável pela descryptografia das informações enviadas por LoRa e portanto possui as chaves NwkSKey e AppSKey (Fig. 50b) que devem ser as mesmas que as configuradas no end-node;
- **Parseamento Aplicação:** Logo após os dados estarem descryptografados, este bloco é responsável por realizar o processo inverso da concatenação de variáveis, isto é, identificar os atributos de cada aplicação;
- **Separando End-Nodes:** responsável por enviar as variáveis de cada aplicação aos seus respectivos dispositivos virtuais na Dojot Fig. 50c.

Nesta Fig. 50c é possível observar detalhadamente como os dados são transmitidos aos dispositivos virtuais. Primeiramente os dados enviados do gateway já estão todos serializados (de acordo com 4.4.2.1). Estes precisam ser decifrados pelo bloco **Decifragem LoRa**, logo depois o parseamento identifica os atributos de cada aplicação pelo primeiro byte do pacote serializado e pela ordem que as variáveis foram concatenadas. Logo depois o parseamento envia as informações para os respectivos dispositivos virtuais.

Figura 50 – (a) *Flowbroker*; (b) Processamento de dados LoRa na Dojot; (c) Percurso dos dados até os dispositivos Virtuais na Dojot



Fonte: Elaborado pelo Autor

4.9 Síntese Sobre as 4 Aplicações

Foi visto que necessidades distintas exigem requisitos de implementação diferentes para cada aplicação. Para facilitar o entendimento ainda mais a Tabela 7 mostra as principais diferenças e semelhanças das 4 aplicações descritas neste capítulo.

É notório a diferença entre as aplicações estáticas e as móveis, haja vista que as estáticas utilizam um único dispositivo para captura de dados (As UACTs), porém necessita de mais um protocolo (MODBUS) para acessar as informações. enquanto isso, as outras utilizam mais de 2 dispositivos para acesso dos dados. Além disso o **Ônibus Intermunicipal** é a única aplicação em que seus dados não precisam ser processados no *flowbroker*, já que o protocolo MQTT garante que os dados sejam enviados diretamente para os dispositivos virtuais.

Tabela 7 – Síntese Sobre as 4 aplicações

Aplicação	Transmissão	Características	
Ônibus Circular	LoRa	Transmissão assíncrona	
		Estruturação	Serialização
		Possui GPS?	SIM
		Captura de Dados	Módulo GPS (Shield LoRa/GPS) 3 CFPs e sensor DHT11
		Utiliza Gateway e <i>flowbroker</i> ?	SIM
Ônibus Intermunicipal	GSM GPRS MQTT	Transmissão síncrona	
		Estruturação	JSON
		Possui GPS?	SIM
		Captura de Dados	Shield GPS DUINOPEAK 1 CFP e sensor DHT11
		Utiliza Gateway e <i>flowbroker</i> ?	NÃO
Sistema Fotovoltáico	LoRa	Transmissão assíncrona	
		Estruturação	Serialização
		Possui GPS?	NÃO
		Captura de Dados	UACT 3F
		Utiliza Gateway e <i>flowbroker</i> ?	SIM
Sistema de Baterias	LoRa	Transmissão assíncrona	
		Estruturação	Serialização
		Possui GPSw	NÃO
		Captura de Dados	UACT CC
		Utiliza Gateway e <i>flowbroker</i> ?	SIM

4.10 Conclusão do capítulo

Neste capítulo foram descritas as diferentes metodologias das aplicações do projeto SIMA. É possível perceber como as variadas tecnologias estão conectadas no *smart campus* da UFPA, bem como a complexidade que cada uma delas apresenta.

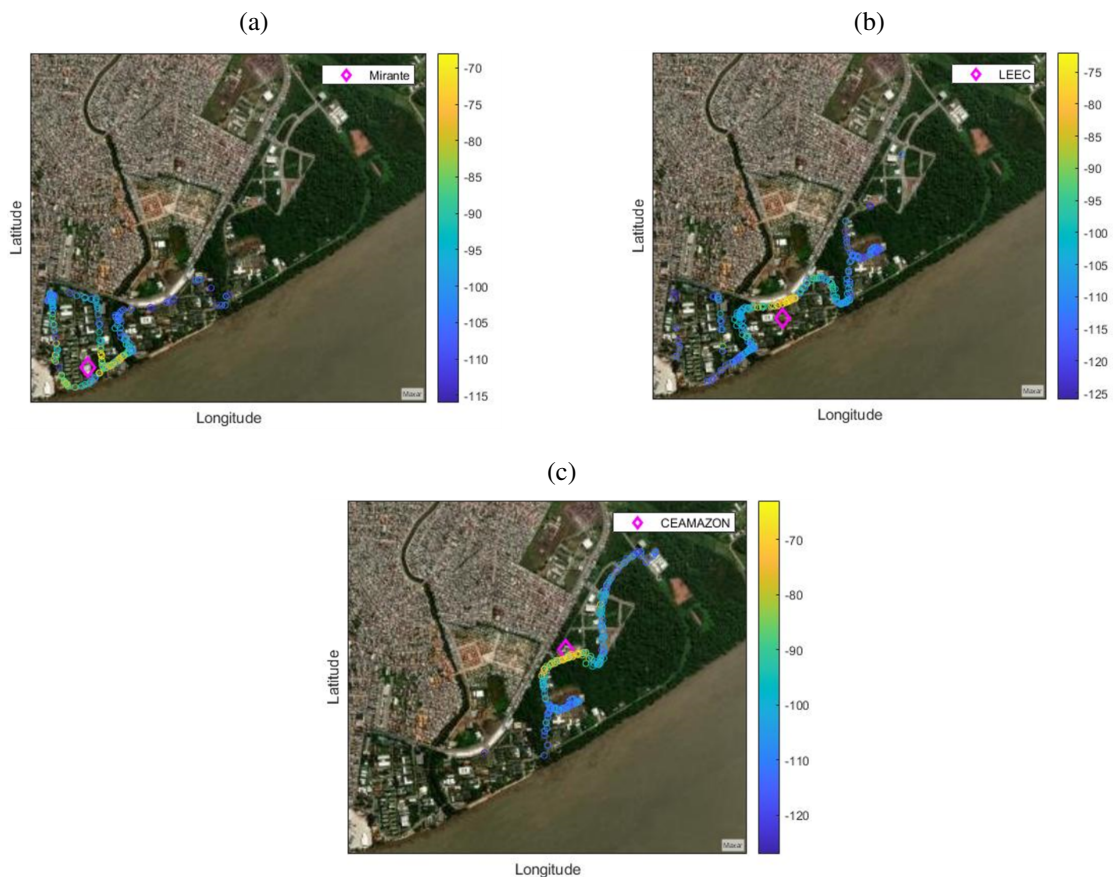
5 Resultados

5.1 Conteúdo do Capítulo

5.2 Análise de Cobertura dos Gateways

Com o objetivo de analisar a área de cobertura de cada um dos gateways foi realizado um teste em que um end-node no interior do Ônibus Circular transmitia informação de RSSI e geolocalização. E verificou-se como o RSSI se modificava a medida que o ônibus seguia sua rota no *smart campus*. As Figs. 51a, 51b e 51c se referem as amostras de RSSI coletadas dos 3 gateways.

Figura 51 – Instalação de gateways no *smart campus* da UFPA (a) Mirante; (b) Prédio do LEEB; (c) Prédio do CEAMAZON



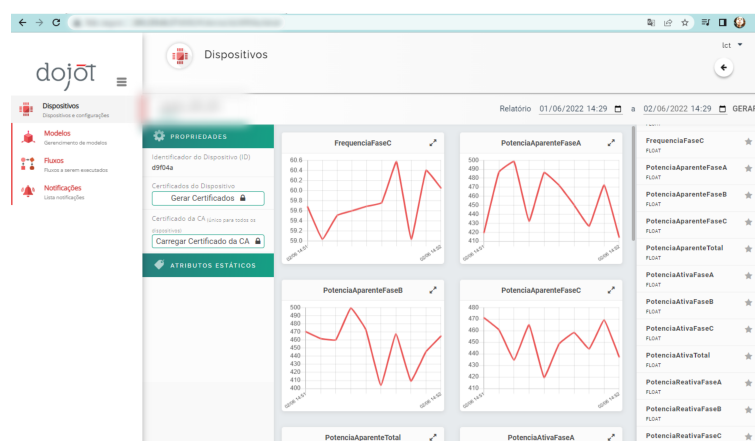
Fonte: Elaborado pelo Autor

A utilização dos 3 gateways aumenta o alcance da transmissão LoRa consideravelmente. Todavia há existência de 3 gateways no campus pode acarretar em redundâncias com mais de 1 gateway enviando a mesma informação para a Dojot em intervalos de tempo distintos. Porém é possível eliminar essa redundância, e mesmo que a mesma informação chegue em intervalos distintos na Dojot, cada uma delas possui o mesmo *timestamp*.

5.3 Teste de Validação da Serialização

A validação descrita em ?? foi realizada utilizando um end-node apenas com o shield LoRa de forma que o pacote enviado tinha o mesmo formato descrito na serialização: código de aplicação, timestamp e variáveis de interesse. As variáveis de interesse foram números aleatórios gerados utilizando a função **rand** no Arduino. A Fig. 52 mostra os resultados deste teste.

Figura 52 – Resultados do teste de validação da serialização tendo como exemplo o **Sistema Fotovoltáico**



Fonte: Elaborado pelo Autor

5.4 Ônibus Circular

5.4.1 Montagem do Transmissor

A montagem do transmissor desenvolvida por Silva (2022) do **Ônibus Circular** pode ser observado na Fig. 53. Nele pode-ser observar os seus componentes separadamente e a montagem completa. O RS485 é por onde a comunicação com o contador de pessoas (CFP 100) será realizada.

Figura 53 – Dispositivo Transmissor do **Ônibus Circular**

Fonte: Elaborado pelo Autor

5.4.2 Instalação do Transmissor

O transmissor foi instalado com êxito no interior do **Ônibus Circular** (Fig. 54). O dispositivo contador de pessoas (CFP 100) ainda não foi introduzido porém testes realizados por Silva (2022) mostram a geolocalização que pode ser acompanhada em tempo real pelos projetistas.

Figura 54 – Instalação do End-node do transmissor do **Ônibus Circular**

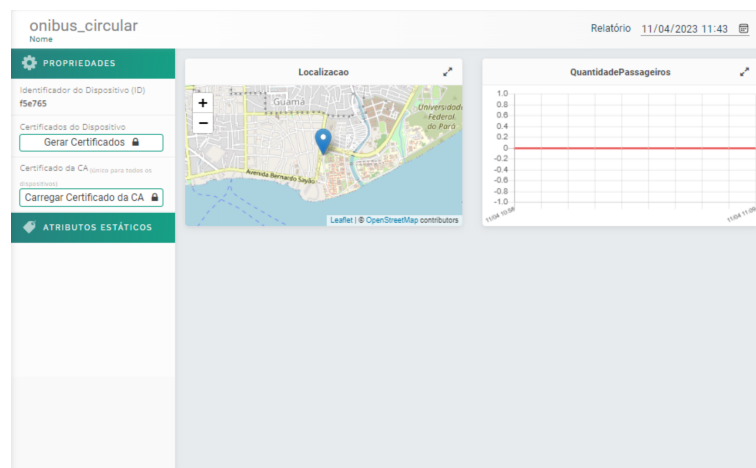


Fonte: Elaborado pelo Autor

5.4.3 Resultados da Transmissão do **Ônibus Circular**

Os resultados mostrados a seguir estão relacionados a transmissão ocorrida em 24 de fevereiro às 17h48min com o end-node já instalado e funcionando. Na Fig. 55 são mostrados apenas a geolocalização e a temperatura, já que o dispositivo contador de pessoas ainda não foi instalado.

Figura 55 – Resultados da Transmissão do **Ônibus Circular**



Fonte: Elaborado pelo Autor

Além disso foi realizada a extração da geolocalização da Dojot através da API *openStreetMap* para acrescentar esta informação no site do Laboratório de Computação e Telecomunicações (LCT), como pode ser observado na Fig. 56. Desta forma as pessoas que frequentemente utilizam o circular para se locomover pelo campus podem acompanhar em tempo real o trajeto do ônibus.

Figura 56 – Geolocalização do **Ônibus Circular** visto no site do LCT

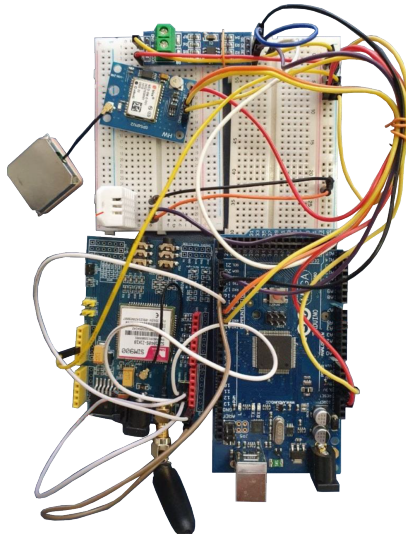


Fonte: Elaborado pelo Autor

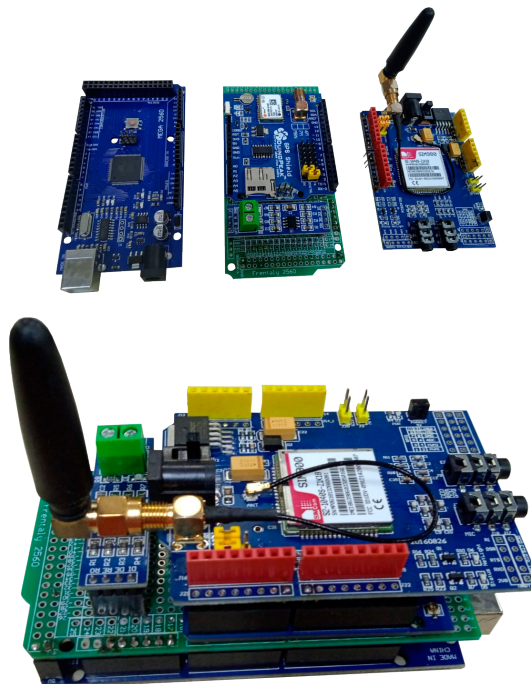
5.5 Ônibus Intermunicipal

5.5.1 Aprimoramento do Transmissor do Ônibus Intermunicipal

Foi desenvolvida melhorias em relação ao design do transmissor, haja vista que Luz (2022) realizou testes utilizando um protótipo com vários jumpers e os componentes do circuito separados uns dos outros. A Fig. 57 mostra o transmissor desenvolvido por Luz (2022) e a Fig. 58 o end-node aprimorado.

Figura 57 – Protótipo do transmissor do **Ônibus Intermunicipal**

Fonte: Adaptado de (LUZ, 2022)

Figura 58 – Transmissor do **Ônibus Intermunicipal** aprimorado

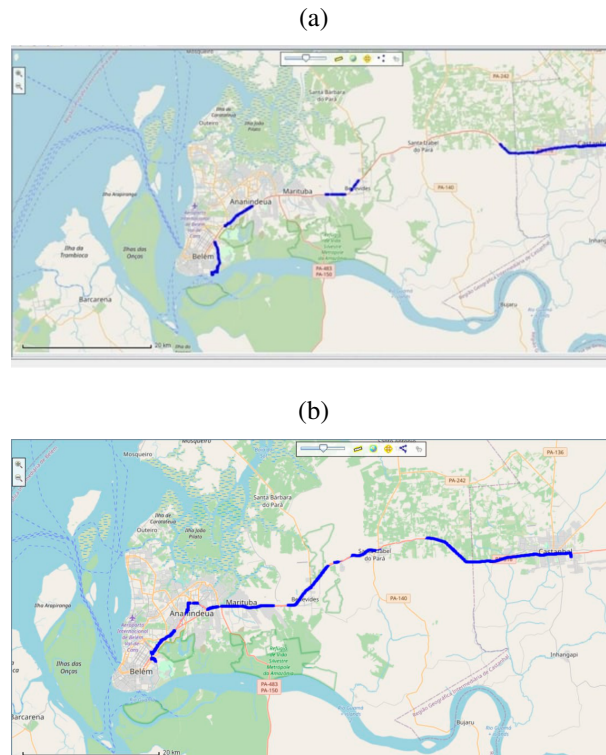
Fonte: Elaborado pelo Autor

5.5.2 Teste em Campo

Para verificação do transmissor antes e depois da melhoria da reconexão foram realizados dois testes em campo: um com o transmissor sem a reconexão e o outro com a reconexão em uma viagem de Belém para Castanhal. Os resultados deste teste foram obtidos extraindo as geolocalizações da Dojot e são mostrados nas Figs. 59a e 59b.

O primeiro teste foi realizado com o código sem reconexão, de forma que nos períodos

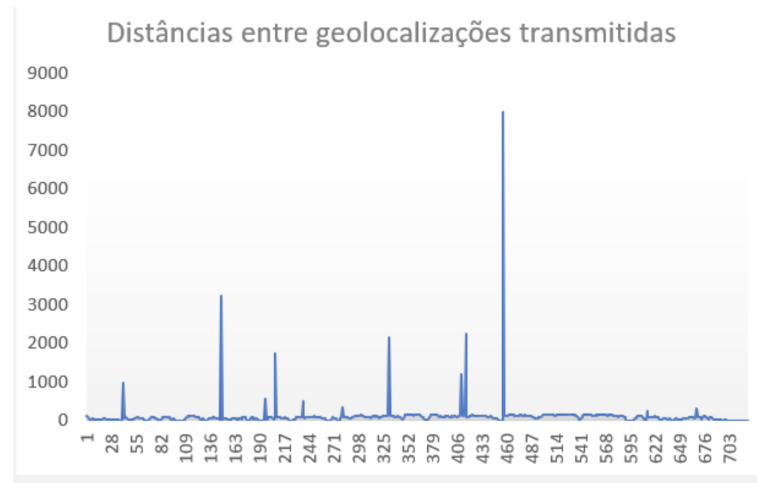
Figura 59 – Testes realizados em campo com o transmissor do *Intermunicipal*. (a) Sem a reconexão; (b) Com a reconexão



Fonte: Elaborado pelo Autor

em que o shield perdia a conexão GSM, esperava-se um tempo arbitrário para verificar se a reconexão ocorreria, e posteriormente o transmissor era reiniciado manualmente. E por isso podem ser observados algumas regiões que o **Ônibus Intermunicipal** deixou de transmitir. A Fig. 60 mostra o gráfico que mostra distâncias entre as capturas de GPS durante a rota de Belém-Castanhal. Por exemplo, a primeira distância se refere a distância entre a primeira e a segunda amostra de GPS que foram capturadas durante a rota. As distâncias com desvio padrão alto são consideradas regiões de sombra que totaliza 21Km. A maior dessas regiões é o perímetro próximo a Santa Isabel, que pode ser observado em 59b, cujo valor é de aproximadamente 8Km.

Figura 60 – Distâncias entre pontos os pontos de GPS capturados durante a rota



Fonte: Elaborado pelo Autor

5.6 Sistema Fotovoltáico

5.6.1 Setup do Transmissor

O design deste dispositivo foi desenvolvido utilizando uma placa perfurada para circuito impresso a fim de auxiliar no encaixe com o Arduino. Sob esta placa foi inserido o RS485, para transmissão de dados com o UACT 3F, e born de 3 entradas (A, B e GND) relacionadas as portas do RS485. Este setup pode ser visto na Fig. 61.

Figura 61 – Transmissor do **Sistema Fotovoltáico**

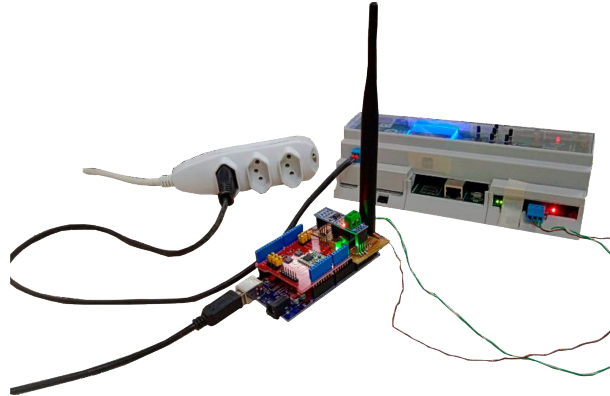
Fonte: Elaborado pelo Autor

5.6.2 Testes em Laboratório

Para validar a metodologia de transmissão do **Sistema Fotovoltáico** foi utilizado um UACT 3F de teste, em que foi possível analisar a transmissão sem o UACT 3F estar conectado a aplicação. Foi instalado um *firmware* no dispositivo de tal forma que dados fixos foram inseridos

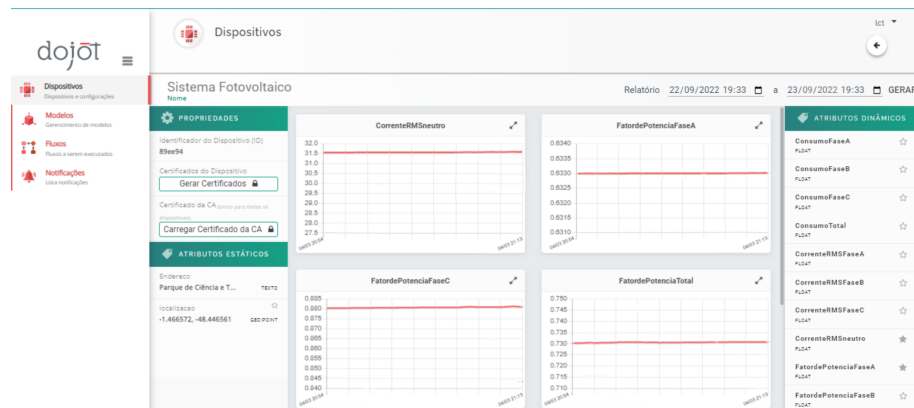
nos registradores tal como na Tabela 5. O teste em laboratório e os resultados na Dojot podem ser visualizados, respectivamente, nas Figs. 62 e 63.

Figura 62 – End-node do Sistema Fotovoltáico



Fonte: Elaborado pelo Autor

Figura 63 – Resultado do Teste realizado em laboratório visualizados na Dojot



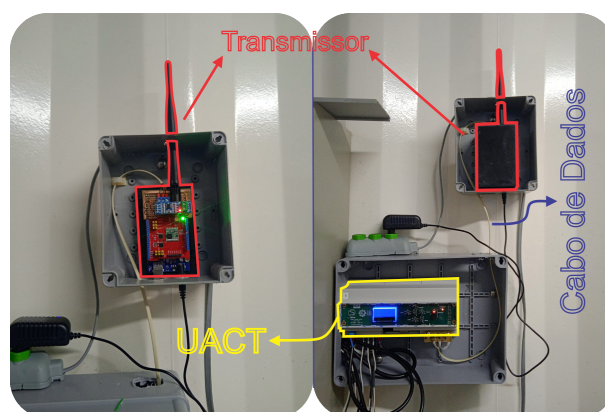
Fonte: Elaborado pelo Autor

A análise concluiu que as informações observadas na Dojot coincidem com as variáveis fixas armazenadas no firmware do UACT 3F.

5.6.3 Instalação do Transmissor e Teste em Campo

A instalação do UACT 3F foi realizada ao lado do quadro de energia da aplicação **Sistema Fotovoltáico**, e dessa forma optou-se por instalar o transmissor próximo ao medidor (Fig. 64). O quadro de energia que disponibiliza os dados pode ser observado na Fig. 65.

Figura 64 – Transmissor do **Sistema Fotovoltáico** instalado ao lado do UACT 3F



Fonte: Elaborado pelo Autor

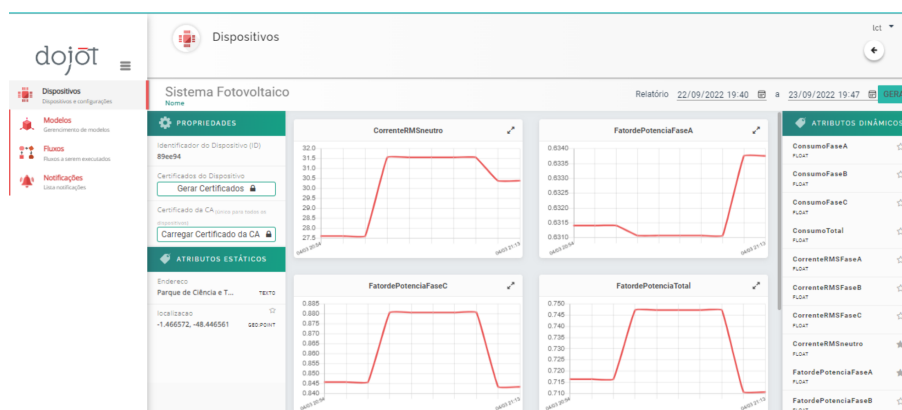
Figura 65 – Quadro de energia que disponibiliza as informações ao UACT 3F



Fonte: Elaborado pelo Autor

A transmissão desta aplicação tiveram êxito, como pode ser observado na Fig. 66. Souza (2023) analisou as amostras de corrente RMS A, B e C durante 24h, tomando como intervalo os 10min. Está análise consistiu na comparação entre as amostras que chegam a Dojot por meio do UACT 3F (transmissão LoRa, um dos focos deste trabalho) e a informação medida pelo software PQA-HiVIEW. Souza (2023) concluiu que o erro médio entre as amostras, considerando as duas metodologias, não escedeu 1%.

Figura 66 – Quadro de energia que disponibiliza as informações ao UACT 3F



Fonte: Elaborado pelo Autor

5.7 Sistema de Baterias

5.7.1 Montagem e Instalação do Transmissor

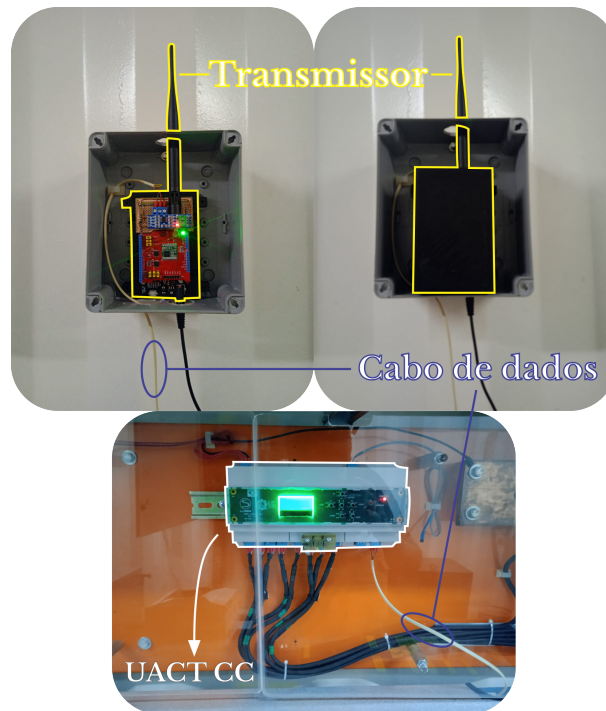
Como a arquitetura do **Sistema de Baterias** é análoga ao **Sistema Fotovoltáico**, o setup do transmissor é o mesmo observado na Fig. 61. respectivamente. Além disso, foi realizada a instalação do transmissor próxima ao UACT CC (Fig. 67), para acesso as variáveis da aplicação, porém ainda não foram realizado testes em campo, embora testes em laboratório tenham ocorrido em função da validação da metodologia.

5.7.2 Testes Realizados no Laboratório

O teste consistiu em um setup análogo a 63 armazenando variáveis fixas nos registradores tal como na Tabela 6. Em que substituiu-se o UACT 3F pelo UACT CC, com mesmo algoritmo descrito em 4.5.4.2. O resultado pode ser verificado na Fig. 68.

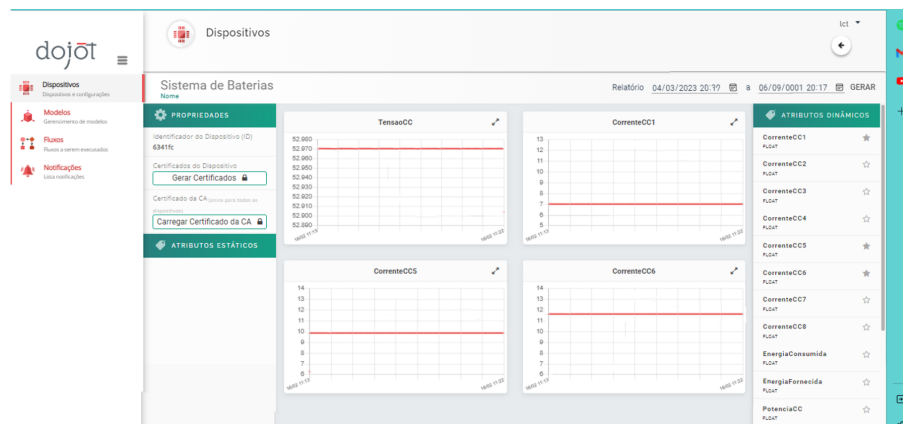
As amostras obtidas na Dojot também coincidiram com os valores fixos armazenados na UACT CC, comprovando dessa forma a metodologia proposta.

Figura 67 – Instalação do Transmissor do Sistema de Baterias



Fonte: Elaborado pelo Autor

Figura 68 – Resultado do teste em laboratório referente ao Sistema de Baterias



Fonte: Elaborado pelo Autor

5.8 Análise do *time-on-air* para as Aplicações LoRa

Como mencionado em 4.4.2.1 a serialização proporciona diminuição do tempo de transmissão do end-node para o gateway devido a existência de um único preâmbulo. Dessa forma, a Tabela 8 apresenta a comparação dos tempos de transmissões com e sem serialização. Para análise do *time-on-air* considerou-se que a transmissão sem serialização seria com o envio individual de cada variável (desconsiderando o ID_2). Foi verificado em TTN (2023) qual o tempo de transmissão de uma variável de 4 bytes em SF8 e SF9 que resultou em 92,7 e 164,9 ms, respectivamente; Cada um desses valores foi multiplicado pelo número de variáveis que

aplicação contém, obtendo o *time-on-air* sem serialização. Para as variáveis com serialização, apenas verificou-se o tempo de transmissão de todas as variáveis concatenadas. Os resultados mostraram considerável diminuição no tempo de transmissão do pacote LoRa em todas as aplicações.

Tabela 8 – Tempo no ar das 3 aplicações sem e com serialização

Aplicação	Forma de Envio		Redução do <i>time-on-air</i>
	Sem Serialização	Com Serialização	
Ônibus Circular (SF9)	164,9 ms x 4 = 659,6 ms	205,8 ms	68,8%
Sistema Fotovoltáico (SF8)	92,7 ms x 30 = 2873,7 ms	399,9 ms	86,08%
Sistema de Baterias (SF9)	164,9 ms x 13 = 2143,7 ms	390,1	81,8%

Fonte: Elaborado pelo Autor

5.9 Conclusão do Capítulo

Neste capítulo foram descritos diversos resultados a respeito das aplicações **Ônibus Circular**, **Ônibus Intermunicipal**, **Sistema Fotovoltáico** e **Sistema de Baterias**. Foi possível concluir a eficácia das múltiplas metodologias empregadas, bem como alguns problemas encontrados nas transmissões e etapas que ainda faltam ser concluídas.

Conclusão

De forma geral este trabalho demonstrou grande importância para pesquisas que envolvem Internet das Coisas, haja vista engloba várias tecnologias, de forma a descrever como cada uma delas está interconectada com a Dojot. Vale ressaltar a importância da tecnologia LoRa, que recentemente vêm sendo bastante usada em aplicações IoT.

A metodologia de serialização de dados, composta por código da aplicação, timestamp e as variáveis da aplicação, foi fundamental para garantir a integridade e a segurança dos dados transmitidos. E As informações do *Ônibus Circular*, **Sistema de Baterias** e **Sistema Fotovoltáico** podem ser monitoradas.

O **Ônibus Circular** está em funcionamento atualmente os alunos da UFPA e a comidade de forma geral podem acessar a sua geolocalização no site do LCT. Quando o CFP for instalado, muitas pesquisas podem ser desenvolvidas em relação ao fluxo de pessoas no ônibus, haja vista que os projetistas tem acesso ao banco de dados para possíveis análises.

Quanto ao **Ônibus Intermunicipal** foi possível verificar a eficácia na transmissão de suas variáveis em transmissões entre cidades, e como a rede GSM se adequou ao sistema IoT. Seus resultados demonstram que o dispositivo foi capaz de se reconectar automaticamente após passar pelas regiões de sombra onde a rede GSM não funcionava.

A transmissão do **Sistema Fotovoltáico** está atualmente em funcionamento, e suas variáveis podem ser monitoradas na Dojot, o que demonstra sucesso na aquisição de dados MODBUS, bem como na transmissão de dados. Desta forma, pesquisas sobre os dados da aplicação podem ser desenvolvidos.

Trabalhos Futuros

Como trabalhos futuros propõe-se:

- Analisar possíveis perdas de pacote para transmissões LoRa;
- Analisar perda de pacote, bem como qualidade do sinal GSM na rota Belém-Castanhal;
- Realizar transmissão de dados do **Ônibus Intermunicipal**, utilizando outras tecnologias como LTE e comparando-as;
- Acrescentar outras variáveis às aplicações do **Ônibus Circular** e **Intermunicipal** como: sensor de fogo para prevenir possíveis incêndios, identificando aumento de temperatura repentino; sensor de luminosidade para controle da iluminação interna do veículo;

- Desenvolvimento de um sistema capaz de contar pessoas e dessa forma diminuir o custo do projeto;
- Realizar a transmissão de dados das aplicações estáticas utilizando Wi-Fi, e comparar com a transmissão LoRa.

Referências

ADELANTADO, F. et al. Understanding the limits of lorawan. *IEEE Communications magazine*, IEEE, v. 55, n. 9, p. 34–40, 2017.

ANATEL. Lei nº 14448, de 4 de dezembro de 2017. estabelece os requisitos técnicos dos equipamentos de radiocomunicação de radiação restrita, conforme previsto no art. 10 do regulamento sobre equipamentos de radiocomunicação de radiação restrita, aprovado pela resolução nº 680, de 27 de junho de 2017. *Agência Nacional de Telecomunicações*, Brasília, DF, 2012. ISSN 1677-7042. Disponível em: <informacoes.anatel.gov.br/legislacao/atos-de-certificacao-de-produtos/2017/1139-ato-14448>.

ATZORI, L.; IERA, A.; MORABITO, G. The internet of things: A survey. *Computer networks*, Elsevier, v. 54, n. 15, p. 2787–2805, 2010.

BORGIA, E. The internet of things vision: Key features, applications and open issues. *Computer Communications*, Elsevier, v. 54, p. 1–31, 2014.

CAMPOS, M. de P. *AMPLIFICADOR COM ENTRADAS E SAÍDAS DIFERENCIAIS INTEGRADO EM TECNOLOGIA CMOS*. Tese (Doutorado) — Universidade Estadual de Campinas, 2002.

CARVALHO, H.; SÁ, J.; FARIAS, F. Implantação de uma arquitetura de software para monitoramento de dados ambientais em um cenário de smart campus. In: *SBC. Anais do XII Workshop de Computação Aplicada à Gestão do Meio Ambiente e Recursos Naturais*. [S.l.], 2021. p. 167–170.

CISCO, U. Cisco annual internet report (2018–2023) white paper. *Cisco: San Jose, CA, USA*, v. 10, n. 1, p. 1–35, 2020.

COMPANY, M. . Unlocking the potential of the internet of things. 2015. Disponível em: <<https://www.mckinsey.com/~media/McKinsey/Business%20Functions/McKinsey%20Digital/Our%20Insights/The%20Internet%20of%20Things%20The%20value%20of%20digitizing%20the%20physical%20world/Unlocking%20the%20potential%20of%20the%20Internet%20of%20Things.ashx>>.

CPQD. *Você Conhece a Dojot?* [S.l.], 2017. Disponível em: <<https://dojot.com.br/sobre-a-dojot-iot/>>.

DEVALAL, S.; KARTHIKEYAN, A. Lora technology - an overview. In: *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*. [S.l.: s.n.], 2018. p. 284–290.

DUNN, J. We put siri, alexa, google assistant, and cortana through a marathon of tests to see who’s winning the virtual assistant race—here’s what we found. *Business Insider*, 2016.

EMMANUEL, A. L. et al. Optimization of spreading factor distribution in high density lora networks. In: *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. [S.l.: s.n.], 2020. p. 1–5.

- FACINA RAUL G. TANCREDO, M. S. P. F. e. G. F. A. R. Comunicação lora na zona rural: Análise baseada em uma campanha de medição no noroeste paulista. *Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*, SBRT, n. 39, Setembro 2021.
- JIA, H.-j.; GUO, Z.-h. Research on the technology of rs485 over ethernet. In: *2010 International Conference on E-Product E-Service and E-Entertainment*. [S.l.: s.n.], 2010. p. 1–3.
- JIN, J. et al. Security and privacy in internet of things (iots): Models, architectures and applications. *IEEE Communications Magazine*, IEEE, v. 55, n. 1, p. 26–33, 2017.
- JÚNIOR, U. C. P. *Metodologia para Desenvolvimento e Gestão de Comunidades de Energia, com Estudo de Caso na Universidade Federal do Pará*. Tese (Doutorado) — Programa de Pós Graduação em Engenharia Elétrica - Universidade Federal do Pará, 2022.
- KAHAN, W. Ieee standard 754 for binary floating-point arithmetic. *Lecture Notes on the Status of IEEE*, v. 754, n. 94720-1776, p. 11, 1996.
- LOBATO, A. C. D. B. Elen Priscila de S. et al. Smart city: application of the abnt nbr iso 37122: 2020 standard in the university city of ufpa. In: IEEE. *2021 14th IEEE International Conference on Industry Applications (INDUSCON)*. [S.l.], 2021. p. 1258–1265.
- LOBATO, E. P. de S. *Implantação de um middleware IoT escalável para aplicações de Mobilidade Elétrica Multimodal*. Dissertação (Mestrado) — Universidade Federal do Pará. Programa de Pós Graduação em Engenharia Elétrica, outubro 2022.
- LOPES, A. V. R. *Análise Estatística e Modelagem Empírica para Campus Densamente Arborizado Utilizando a Tecnologia LoRa em 915 MHz*. Dissertação (Mestrado) — Universidade Federal do Pará. Programa de Pós Graduação em Engenharia Elétrica, 2021.
- LORA ALLIANCE. *What is LoRaWAN® Specification*. [S.l.], 2015. Disponível em: <lora-alliance.org/about-lorawan>.
- LUZ, J. V. A. *Um Protótipo para Mobilidade Urbana: IoT, GSM, Temperatura, Geolocalização E CFP 100 no Controle de Fluxo de Pessoas*. Dissertação (Trabalho de Conclusão de Curso) — Faculdade de Computação e Telecomunicações Universidade Federal do Pará, Julho 2022.
- MANUEL, A. V. et al. Fiee smart campus iot real-time bus tracking system and web app using lorawan. In: *2022 IEEE International Smart Cities Conference (ISC2)*. [S.l.: s.n.], 2022. p. 1–7.
- MATOS, M. de L. *Relatório técnico de Proposta de Reestruturação da Infraestrutura da Universidade de Brasília Baseada no Conceito de Smart Campus*. Dissertação (Mestrado) — Universidade de Brasília. Programa de Pós Graduação em Propriedade Intelectual e Transferência de Tecnologia para Inovação, Maio 2021.
- MODBUS. *MODBUS Application Protocol Specification*. [S.l.], 2012. Disponível em: <modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf>.
- MONICO, J. F. G. *Posicionamento pelo GNSS: descrição, fundamentos e aplicações*. [S.l.]: Editora Unesp, 2007.
- NARDO, M. D.; YU, H. *Special issue “industry 5.0: The prelude to the sixth industrial revolution”*. [S.l.]: MDPI, 2021. 45 p.

- OASIS. *MQTT Version 5.0*. [S.l.], 2019. Disponível em: <docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.pdf>.
- PAGLIARO, F. et al. A roadmap toward the development of sapienza smart campus. In: *2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC)*. [S.l.: s.n.], 2016. p. 1–6.
- PANITZ, M. A. *Dicionário técnico: português-inglês*. [S.l.]: EDIPUCRS, 2003.
- PINHÃO, C. M. B. *Estratégias de Otimização da Interface ABIS em Redes de Telefonia Celular GSM*. Dissertação (Mestrado) — Programa de Pós Graduação em Engenharia da Universidade Federal do Rio de Janeiro, Março 2006.
- PIROTTI, R.; ZUCCOLOTTI, M. Transmissão de dados através de telefonia celular: arquitetura das redes gsm e gprs. *Revista Liberato*, v. 10, n. 13, p. 81–90, 2009.
- RAMLI, N. et al. An open source lora based vehicle tracking system. *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, v. 7, n. 2, p. 221–228, 2019.
- S, T. A.; J, W. D. *Computer networks fifth edition*. [S.l.]: pearson, 2011.
- SÁNCHEZ-TORRES, B. et al. Smart campus: Trends in cybersecurity and future development. *Revista Facultad de Ingeniería, Universidad Pedagógica y Tecnológica de Colombia*, v. 27, n. 47, p. 104–112, 2018.
- SCHILLER, J. H. *Mobile communications*. [S.l.]: Pearson education, 2003.
- SEMTECH. *Single Channel LoRa IoT Kit v2 User Manual: Document Version: 1.0.1*. [S.l.], 2019. Disponível em: <www.dragino.com/downloads/downloads/LoRa_IoT_Kit/v2-Kit/Single%20Channel%20LoRa%20IoT%20Kit%20v2%20User%20Manual_v1.0.5.pdf>.
- SEMTECH. *AN1200.22 LoRa® Modulation Basics*. [S.l.], 2022.
- SILVA, M. C. D. P. da. *Sistema IoT/LoRa para Controle de Fluxo de Pessoas no Ônibus Circular da UFPA*. Dissertação (Trabalho de Conclusão de Curso) — Faculdade de Computação e Telecomunicações Universidade Federal do Pará, Julho 2022.
- SIMCOM. *SIM900 Hardware Design V2.05*. [S.l.], 2013. Disponível em: <simcom.ee/documents/SIM900/SIM900_Hardware%20Design_V2.05.pdf>.
- SOUZA, G. M. G. de. *Unidade de Aquisição de Corrente E Tensão Trifásica (UACT-3F)*. Dissertação (Trabalho de Conclusão de Curso) — Faculdade de Computação e Telecomunicações Universidade Federal do Pará, 2023.
- THE THINGS NETWORK. *LoRaWAN airtime calculator*. [S.l.], 2023.
- VEJLGAARD, B. et al. Coverage and capacity analysis of sigfox, lora, gprs, and nb-iot. In: *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*. [S.l.: s.n.], 2017. p. 1–5.
- WIXTED, A. J. et al. Evaluation of lora and lorawan for wireless sensor networks. In: *IEEE. 2016 IEEE SENSORS*. [S.l.], 2016. p. 1–3.
- XU, X. et al. Industry 4.0 and industry 5.0—inception, conception and perception. *Journal of Manufacturing Systems*, Elsevier, v. 61, p. 530–535, 2021.