

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

CAROLINE ROSA DA SILVA

**BLIP SOS: ACESSÓRIO IOT PARA ENVIO DE SOS COM GEOLOCALIZAÇÃO
EM SITUAÇÕES EMERGENCIAIS**

CURITIBA

2022

CAROLINE ROSA DA SILVA

**BLIP SOS: ACESSÓRIO IOT PARA ENVIO DE SOS COM GEOLOCALIZAÇÃO
EM SITUAÇÕES EMERGENCIAIS**

**Blip SOS: IOT accessory for sending SOS with geolocation in emergency
situations**

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia de Computação do Curso de Bacharelado em Engenharia de Computação da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Guilherme Luiz Moritz

Coorientador: Thalles Verçosa Palanca

CURITIBA

2022



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

CAROLINE ROSA DA SILVA

**BLIP SOS: ACESSÓRIO IOT PARA ENVIO DE SOS COM GEOLOCALIZAÇÃO
EM SITUAÇÕES EMERGENCIAIS**

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia de Computação do Curso de Bacharelado em Engenharia de Computação da Universidade Tecnológica Federal do Paraná.

Data de aprovação: 15/dezembro/2022

Guilherme Luiz Moritz
Doutorado
Universidade Tecnológica Federal do Paraná

Guilherme de Santi Peron
Doutorado
Universidade Tecnológica Federal do Paraná

Ana Cristina Barreiras Kochem Vendramin
Doutorado
Universidade Tecnológica Federal do Paraná

Marcos Eduardo Pivaro Monteiro
Doutorado
Universidade Tecnológica Federal do Paraná

CURITIBA
2022

Dedico este trabalho à minha família, pelo apoio e incentivo durante toda a minha jornada.

AGRADECIMENTOS

Agradeço grandemente à meu orientador Prof. Guilherme Mortiz pelo suporte e disponibilidade durante a jornada deste TCC, por me auxiliar nos assuntos em que tinha mais necessidade e me guiar durante toda a realização deste projeto.

Agradeço também ao meu coordenador Thalles Palanca, por me amparar no desenvolvimento deste trabalho e sempre me incentivar a ser uma profissional melhor.

Agradeço à UTFPR e a todos os docentes que fizeram parte da minha trajetória durante o curso, por terem prestado sua contribuição na construção de meus conhecimentos.

Agradeço amplamente à minha família, meu namorado e amigos próximos, por terem me acompanhado durante essa jornada, pelo suporte prestado e por acreditarem em mim, especialmente meu pai, que esteve disponível para me ajudar com os testes do projeto, e minha mãe, pelo apoio emocional.

Por fim, agradeço a Deus por me guiar e me dar forças para alcançar esta conquista.

RESUMO

Existem diversas situações de risco em que uma pessoa pode precisar pedir por socorro, como uma viagem de táxi/aplicativo suspeita, um sequestro ou mesmo um acidente. Normalmente o meio mais viável de solicitar ajuda é o celular, mas isso pode facilmente ser impossibilitado devido à ausência de internet ou mesmo incapacidade de acessar o aparelho para enviar mensagens ou fazer ligações. Pensando nisso, esse projeto propõe a ideia de um produto e o desenvolvimento de seu protótipo, o qual consiste em um dispositivo IoT (Internet das Coisas, do inglês *Internet of Things*), idealmente um acessório pequeno que pode ser mantido sempre em contato com o usuário, cuja simples ação de clicar um botão aciona o envio de uma mensagem contendo sua geolocalização e um pedido de ajuda. A comunicação é feita através de uma rede LoRaWAN (Rede LoRa de Longa Distância, do inglês *LoRa Wide Area Network*), que independe da disponibilidade do celular, e é capaz de repassar a mensagem de SOS a um servidor Amazon Web Services (AWS) disponível na nuvem, que a envia para uma lista de contatos (previamente selecionados por meio de um aplicativo móvel) via Whatsapp e SMS (Serviço de Mensagens Pequenas, do inglês *Short Message Service*). Desse modo, as pessoas selecionadas pelo usuário terão conhecimento de que ele está em perigo e onde está localizado.

Palavras-chave: sos; lorawan; geolocalização; iot; computação em nuvem.

ABSTRACT

There are several risk situations where someone may find themselves in need of asking for help, like a suspicious taxi ride, a kidnapping or even an accident. Usually the most viable way of seeking for help is the cellphone, but this could easily become unable due to no internet reception or even an inability of accessing the device in order to send messages or make calls. Thinking about that, this project proposes the idea of a product and the development of its project, which consists in an IoT (Internet of Things) device, ideally a small accessory that could be always kept close by the user, in which the simple act of pressing a button triggers the sending of a message containing its geolocation and a request for help. The communication happens through a LoRaWAN (LoRa Wide Area Network), which works regardless the availability of a cellphone, and is able to bypass the SOS message to an Amazon Web Services (AWS) cloud server, which sends it to a list of contacts (previously selected through a mobile app) via WhatsApp and SMS (Short Message Service). Thereby, the people selected by the user will be able to know that the user is in danger and where they are located.

Keywords: sos; lorawan; geolocation; iot; cloud computing.

LISTA DE FIGURAS

Figura 1 – Diagrama com os componentes do padrão Publisher-Subscriber	19
Figura 2 – Exemplo de comunicação com MQTT em um sistema IoT	19
Figura 3 – Topologia LoRaWAN	22
Figura 4 – Cobertura The Things Network no Brasil	23
Figura 5 – Pulseira e aplicativo invisawear	25
Figura 6 – Relógio KOSPET Prime 4G	26
Figura 7 – Relógio Claro Kids On	27
Figura 8 – Diagrama de contexto do sistema	29
Figura 9 – Pinout da placa TTGO T-Beam v1.1	31
Figura 10 – Diagrama de contexto do sistema com materiais e componentes definidos	35
Figura 11 – Diagrama de sequência do sistema	35
Figura 12 – Dispositivo final conectado à alimentação	38
Figura 13 – Diagrama de atividades do <i>firmware</i>	39
Figura 14 – Console da aplicação no The Things Stack	41
Figura 15 – Dispositivo cadastrado	42
Figura 16 – Interação AWS	43
Figura 17 – Integração The Things Stack e AWS	43
Figura 18 – Diagrama de componentes do projeto AWS	44
Figura 19 – Diagrama de tabelas do banco de dados	44
Figura 20 – Mensagem <i>uplink</i> recebida no tópico MQTT	45
Figura 21 – Diagrama de sequência do envio de um SOS	46
Figura 22 – Exemplo de mensagem SMS recebida	47
Figura 23 – Modelo de mensagem personalizado	47
Figura 24 – Exemplo de mensagens WhatsApp recebidas	49
Figura 25 – Tela de login	50
Figura 26 – Tela inicial	50
Figura 27 – Tela de dados do usuário	51
Figura 28 – Tela de dados do usuário com modal do id do dispositivo	51
Figura 29 – Tela da lista de contatos	52
Figura 30 – Tela do histórico de envios de SOS	52

Figura 31 – Mapa com locais e resultados dos testes de alcance 53

LISTA DE TABELAS

Tabela 1 – Legenda das cores utilizadas para classificar o resultado de cada teste	53
Tabela 2 – Custos totais	54

LISTAGEM DE CÓDIGOS FONTE

Listagem 1 – Rule handleSOS	44
Listagem 2 – Rule sendSMS e sendWhatsapp	46
Listagem 3 – Schema GraphQL	48

LISTA DE ABREVIATURAS E SIGLAS

Siglas

ABP	Activation By Personalization
AWS	Amazon Web Services
CSS	Espalhamento Espectral Chirp, do inglês <i>Chirp Spread Spectrum</i>
GPS	sistema de Posicionamento global, do inglês <i>Global Positioning System</i>
HTTP	Protocolo de Transferência de Hipertexto, do inglês <i>Hypertext Transfer Protocol</i>
IoT	Internet das Coisas, do inglês <i>Internet of Things</i>
LabSC	Laboratório de Sistemas de Comunicações
LMIC	LoraWAN-in-C
LoRa	Longo Alcance, do inglês <i>Long Range</i>
LoRaWAN	Rede LoRa de Longa Distância, do inglês <i>LoRa Wide Area Network</i>
LPWAN	Rede de Baixa Potência e Grande Distância, do inglês <i>Low Power Wide Area Network</i>
M2M	Machine to Machine
MQTT	Transporte de Filas de Mensagem de Telemetria, do inglês <i>Message Queuing Telemetry Transport</i>
OTAA	Over the Air Activation
SMS	Serviço de Mensagens Pequenas, do inglês <i>Short Message Service</i>
TTN	The Things Network
UTFPR	Universidade Tecnológica Federal do Paraná

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Objetivos	15
1.1.1	Objetivo geral	15
1.1.2	Objetivos específicos	15
1.2	Motivação	15
1.3	Estrutura do trabalho	16
2	REFERENCIAL TEÓRICO	17
2.1	Internet das Coisas	17
2.2	MQTT	18
2.3	Computação em nuvem	20
2.4	LPWAN	20
2.4.1	LoRa	21
2.4.2	LoRaWAN	22
2.5	GPS	24
3	PRODUTOS RELACIONADOS	25
3.1	invisaWear	25
3.2	SmartWatches	26
3.3	Claro Kids On	27
4	METODOLOGIA	28
4.1	Requisitos	28
4.2	Arquitetura	28
4.2.1	Comunicação	29
4.2.1.1	<u>LoRaWAN</u>	29
4.2.1.2	<u>The Things Network</u>	29
4.2.2	Plataforma	30
4.2.2.1	<u>TTGO T-Beam</u>	30
4.2.3	Servidor	32
4.2.3.1	<u>AWS</u>	32
4.2.4	Envio de mensagens	33
4.2.4.1	<u>WhatsApp Cloud API</u>	33

4.2.4.2	SMS	34
4.2.5	Aplicativo móvel	34
4.2.6	Definição dos componentes	34
4.3	Implementação e desenvolvimento	35
4.3.1	Firmware	35
4.3.2	The Things Network	36
4.3.3	Servidor AWS	36
4.3.4	Aplicativo móvel	37
4.3.5	Testes	37
5	RESULTADOS	38
5.1	Protótipo	38
5.2	Firmware	39
5.3	Aplicação The Things Network	40
5.4	Servidor AWS	41
5.5	Projeto WhatsApp Business	46
5.6	Aplicativo Móvel	47
5.7	Testes	49
5.7.1	Alcance	49
5.7.2	Sinal do GPS	52
5.8	Custos	54
5.9	Melhorias e evoluções	54
6	CONCLUSÃO	56
	REFERÊNCIAS	58

1 INTRODUÇÃO

Segurança pessoal é um assunto de interesse comum, independente de idade ou gênero. É bem provável que o cidadão comum brasileiro já tenha passado por alguma situação que lhe causou apreensão ou até mesmo resultou em riscos reais. Atualmente, a taxa de homicídio no país é de 27,5% ao ano a cada 100 mil habitantes, e tem-se 40% dos brasileiros já foram assaltados (PARALELO, 2022). Além disso, nos últimos 2 anos, o número de sequestros relâmpagos subiu em 35% no estado de São Paulo (CROQUER, 2022).

Falando especificamente sobre o público feminino, tem-se dados de que houve mais de 31 mil denúncias e quase 170 mil violações de violência doméstica ou familiar contra as mulheres no primeiro semestre de 2022, e esses números não chegam nem perto de contemplar o número total de vítimas, pois cerca de 70% das mulheres vítimas de feminicídio no Brasil nunca passaram pela rede de proteção (GOV.BR, 2022). A violência doméstica é um acontecimento tão frequente no país que diferentes métodos começaram a ser utilizados visando reduzir as ocorrências, como a utilização de tornozeleiras eletrônicas para monitoramento de agressores de forma a proteger as vítimas (CASTRO, 2019).

Fora de casa, também há um grande nível de insegurança, pois 7 a cada 10 mulheres já foram assediadas e 81% das brasileiras relatam ter sofrido algum tipo de violência durante seus deslocamentos em diferentes meios de transporte, sendo que os piores índices de insegurança pertencem ao transporte público (GALVÃO, 2021). Porém, nem mesmo na utilização de veículos particulares e aplicativos de carona há tranquilidade, dado que cada vez mais se houve falar de casos de assédio e abuso por parte de motoristas (VASCONCELLOS, 2022). Todos esses dados explicitam o alto nível de insegurança no cotidiano do cidadão brasileiro comum, especialmente para o gênero feminino.

Na maioria dessas situações, o meio mais imediato para pedir socorro acaba sendo o telefone celular. Porém, existem inúmeros fatores que podem impossibilitar essa ação, como a indisponibilidade de rede e sinal de telefone para a realização de ligações, a ausência de conexão com a internet para envio de mensagens por aplicativos e redes sociais, a falta de bateria ou até mesmo a impossibilidade de acessar o dispositivo devido à limitações físicas que possam existir em determinada situação de risco. Considerando essas limitações, constata-se que é arriscado depender inteiramente do celular como meio de solicitar socorro, porém quaisquer outros meios são escassos. Além de dispositivos inteligentes, como *smartwatches* e semelhantes, que são pouco utilizados pela população em geral, pouco se resta a fazer além de gritar por socorro, o que pode nem ser possível dependendo da situação.

Considerando o rápido avanço tecnológico atual, onde a cada ano surgem dispositivos e tecnologias de comunicação sem fio otimizados e cada vez mais democráticos, torna-se comum a maior disponibilidade de diferentes opções de aparelhos de comunicação cada vez menores e melhores. Celulares, relógios, acessórios inteligentes surgem com aplicações em diferentes

áreas, entre elas a segurança pessoal. Apesar da disponibilidade de inúmeros produtos e ideias que surgem a cada dia, ainda há muito espaço para inovação.

Levando em conta o contexto apresentado, este trabalho de conclusão de curso propõe a idealização e prototipação de um dispositivo focado em comunicação baseado no tema de segurança pessoal, visando oferecer uma alternativa para melhor suprir essa demanda constante da população por maior sensação de segurança e amparo.

1.1 Objetivos

1.1.1 Objetivo geral

O objetivo geral do presente trabalho é desenvolver um protótipo de um dispositivo capaz de, com um apertado de botão, e independente de conexão com um celular, enviar mensagens de SOS com geolocalização para uma lista de contatos pré-selecionada através de um aplicativo móvel.

1.1.2 Objetivos específicos

Os objetivos específicos deste trabalho são:

- Arquitetar um sistema de comunicação que seja independente da conexão com um telefone celular;
- Transmitir dados através de uma rede sem fio para uma aplicação na nuvem;
- Enviar mensagens personalizadas de forma automática para o celular de uma lista de telefones;
- Desenvolver um aplicativo móvel para agir como porta de entrada de informações do usuário;
- Analisar o desempenho de funcionamento do envio de mensagens do dispositivo;

1.2 Motivação

A ideia do projeto é motivada pela possibilidade de desenvolver um produto que poderia suprir uma necessidade de um público e melhorar a sensação de segurança da população em geral. O propósito, porém, não é criar algo inovador nunca antes visto, nem produzir uma mercadoria a ser comercializada. A inspiração em produtos semelhantes já existentes e a oportunidade de aperfeiçoamento aplicando tecnologias modernas, unidas ao interesse pessoal de

colocar em prática conhecimentos de diversas áreas adquiridos durante todos os anos de curso, fomentaram a decisão de escolher e desenvolver este projeto.

1.3 Estrutura do trabalho

Esta monografia é composta por 6 Capítulos. O primeiro Capítulo após a introdução apresenta o referencial teórico, explicando brevemente alguns conceitos e termos relevantes para a obtenção de uma base para o entendimento do projeto. Em seguida, o Capítulo 3 fala sobre alguns dos produtos usados como referência e inspiração para o desenvolvimento deste projeto. O quarto Capítulo é sobre a metodologia aplicada no projeto, discorrendo sobre as tomadas de decisão e processo de desenvolvimento. O quinto Capítulo apresenta o sistema resultando obtido, e por fim, o sexto Capítulo finaliza com uma conclusão sobre o trabalho realizado e inclui as referências utilizadas.

2 REFERENCIAL TEÓRICO

2.1 Internet das Coisas

A Internet das Coisas (Internet das Coisas, do inglês *Internet of Things* (IoT)) é um termo aplicado a sistemas que possuam dispositivos físicos que transeptam dados através de redes sem fio, geralmente de forma automatizada ou com pouca interação humana (Oracle, 2022). Comumente referidos como sistemas inteligentes, dispositivos IoT incluem desde versões tecnológicas de objetos comuns do dia a dia, como lâmpadas controladas por celular e câmeras de segurança para smart homes, até ferramentas e máquinas complexas em setores industriais, como aparelhos robóticos em linhas de montagem. O conceito de IoT já existe há algum tempo, mas suas aplicações tem evoluído cada vez mais nos últimos anos devido ao aperfeiçoamento na produção de sensores de baixo custo (ampliando seu acesso), melhorias nas tecnologias de conectividade e comunicação, e, principalmente, ao avanço e presença cada vez maior de plataformas de computação em nuvem, oferecendo cada vez mais escalabilidade e infraestrutura.

Falando especificamente do Brasil, a aplicação de serviços IoT, tanto na área empresarial quanto na governamental, tem tido grande avanço. O relatório Provider Lens: Internet das Coisas – Serviços e Soluções publicado em 2022 pela Information Services Group ISG (2022), empresa global de pesquisa e consultoria em tecnologia, destacou significativo crescimento e amadurecimento da utilização de tecnologias IoT no Brasil desde a publicação do Plano Nacional de IoT pelo (Comitê IoT – GS1, 2018), uma iniciativa do governo visando a elaboração de políticas públicas e planos de ação para maior aplicação da tecnologia IoT no país. Como resultado, houve um aumento da presença dessas tecnologias em diversos setores, como agro-negócio, da medicina, logística e manufatura. De acordo com o relatório, a estimativa é de que o Brasil pode chegar a ter até 27,1 bilhões de dispositivos conectados até 2025.

O relatório do ISG constata que, apesar de ainda estar em crescimento e estabelecimento de infraestrutura, o Brasil tem forte atuação no mercado IoT principalmente devido à forte presença de empresas de telecomunicação, apontando expectativas otimistas para o futuro. Além disso, ISG aponta a grande tendência da união das tecnologias IoT com o campo da Ciência de Dados e da Inteligência Artificial (IA), proporcionando expectativas promissoras para análises de dados e previsões de resultados, o que pode ser valioso para qualquer mercado. Por exemplo, dos principais desafios do país é com certeza sua grande extensão geográfica, o que acarreta uma grande quantidade de dispositivos participantes e dados coletados, portanto é fundamental saber as melhores estratégias para a localização dos sistemas e também para o processamento e aplicação dos dados resultantes.

2.2 MQTT

No tema de Internet das Coisas, é fato que a comunicação é uma chave crucial para a construção de sistemas consistentes e relevantes. Considerando a necessidade de protocolos de comunicação ágeis e eficientes para a realizar a troca de grandes quantidades de mensagens em um sistema composto por diversos dispositivos, surgiu nos anos 90, criado pela IBM, o protocolo Transporte de Filas de Mensagem de Telemetria, do inglês *Message Queuing Telemetry Transport* (MQTT). Apresentando uma alternativa ao Protocolo de Transferência de Hipertexto, do inglês *Hypertext Transfer Protocol* (HTTP), o MQTT também se utiliza da arquitetura cliente-servidor, possibilitando a troca de mensagens entre máquinas (Machine to Machine (M2M)) através de uma comunicação leve, confiável, escalável e eficiente (MQTT, 2022). Esse protocolo tem fácil implementação e pode ser executado em diversos protocolos de rede, dando destaque ao TCP/IP.

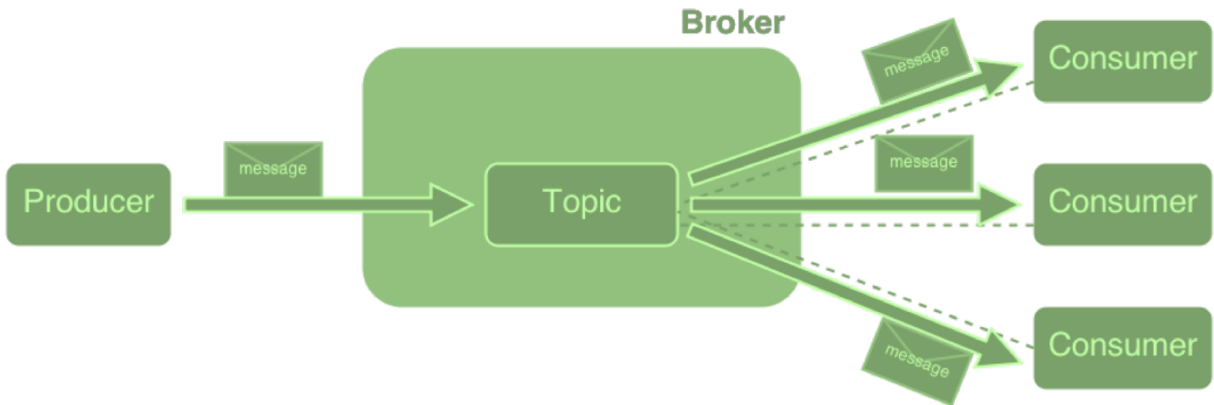
O MQTT utiliza o padrão *Publish/Subscribe*, um paradigma orientado a eventos que define o gerenciamento de mensagens através de papéis de escrita de dados e leitura por parte de interessados (COULOURIS *et al.*, 2013). Esse modelo permite que clientes se inscrevam para consumir uma determinada informação, e quando essa informação for publicada por algum agente, esses clientes serão notificados. Esse modelo de comunicação assíncrona é muito vantajoso para a otimização das transmissões e maior organização e confiabilidade dos dados.

Porém, essa organização só é possível devido à existência do conceito do *broker*, um servidor intermediário que tem a responsabilidade de receber, filtrar e redirecionar as mensagens publicadas aos clientes inscritos. Um meio muito comum para a filtragem e separação das mensagens é a utilização de tópicos. Basicamente, tópicos funcionam como recipientes para diferentes assuntos, de modo que um cliente pode publicar uma mensagem relacionada a um assunto em seu tópico, e todos os clientes interessados no mesmo assunto podem se inscrever nesse tópico para serem notificados e saberem onde encontrar as mensagens. Isso possibilita muita independência e confiabilidade ao gerenciamento de mensagens. Os componentes do padrão pub-sub e seu funcionamento são ilustrados na Figura 1.

Essa arquitetura é muito bem aplicada a sistemas IoT, permitindo que diferentes dispositivos, como sensores ou qualquer tipo de hardware, publiquem dados em um *broker* centralizado, que gerencia essas informações e as disponibilize quando houver necessidade, como um servidor na nuvem ou até mesmo os próprios dispositivos. Dessa forma, é estabelecida uma rede de comunicação leve, bi-direcional e escalável. Além disso, o MQTT é aberto e preparado para a implementação de mecanismos de encriptação e autenticação, garantindo uma boa camada de segurança.

A transmissão dos dados no protocolo MQTT define um *header* que pode variar de 2 a 5 bytes, contendo informações sobre o tipo da mensagem, bits de validação de duplicação e de qualidade da mensagem. Os eventos de conexão, envio e recebimento de dados definem algumas mensagens padrão, que são usadas como resposta e confirmação de diferentes cenários.

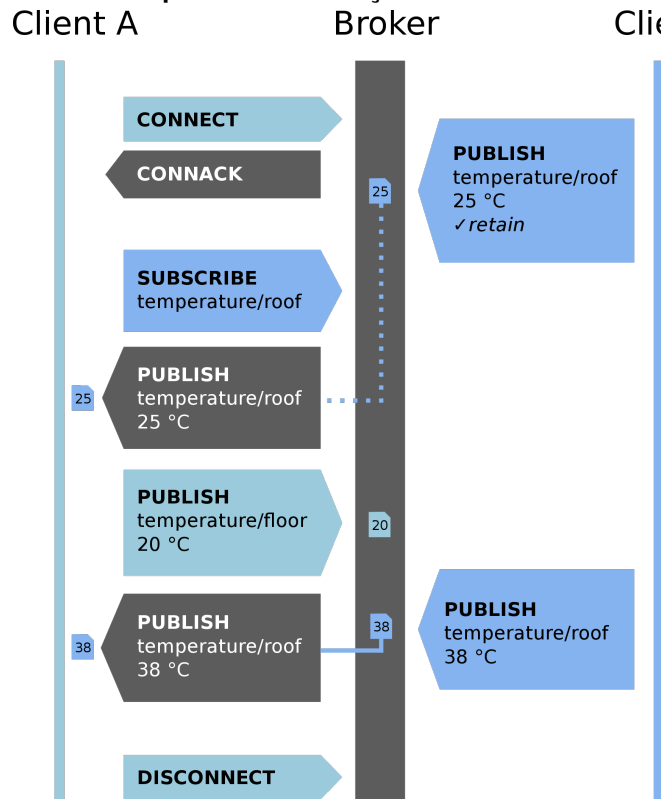
Figura 1 – Diagrama com os componentes do padrão Publisher-Subscriber



Fonte: The New Stack (2022).

A Figura 2 ilustra um exemplo de comunicação entre dois clientes e um broker de um sistema IoT hipotético.

Figura 2 – Exemplo de comunicação com MQTT em um sistema IoT



Fonte: Wikipedia Commons (2018).

2.3 Computação em nuvem

Computação em nuvem, também conhecida pela tradução em inglês *Cloud Computing*, é o termo utilizado para nomear a grande disponibilidade de servidores e recursos relacionados à computação, caracterizados pela presença de grandes centros de servidores geralmente distribuídos em diversas regiões ao redor do mundo, acessíveis sob demanda através da internet (Amazon Web Services, 2022d). Esse termo começou a se popularizar na década passada, quando grandes empresas como Amazon, Google e Microsoft iniciaram uma ofensiva de aperfeiçoamento e inovação desses serviços. As principais características que tornam a computação em nuvem vantajosa incluem a alta escalabilidade e elasticidade de recursos, acompanhados de alta disponibilidade, e a oferta de serviços sob demanda e otimizados, orientados ao consumo. Atualmente, a maioria dos grandes provedores conta com uma infinidade de serviços e ferramentas, desde a implantação e gerenciamento de infraestrutura de softwares até armazenamento e engenharia de dados com inteligência artificial, sendo adotados cada vez mais por empresas e organizações.

No contexto brasileiro, esse tema tem apresentado grande crescimento nos últimos anos. De acordo com relatórios divulgados este ano pela Associação Brasileira das Empresas de Tecnologia da Informação e Comunicação (Brasscom, 2022), a computação em nuvem tem grande destaque ao apresentar previsão de impulso de 24% ao ano, acompanhada de previsões de uma receita de R\$ 181,1 bilhões nos próximos três anos. Essa tecnologia vem sempre acompanhada também do crescimento de tecnologias de *Big Data* e Análise de Dados, seguidos pela área de telecomunicações que é fortemente associada a tecnologias IoT. Dessa forma, notam-se boas previsões de crescimento desse mercado no Brasil, abrindo espaço para aplicação de tecnologias inovadoras e de ponta.

2.4 LPWAN

A comunicação é um dos pilares do IoT, independente do tamanho dos dados e da distância para sua transmissão. Para sistemas pequenos que funcionam pela proximidade, muito se utiliza opções como Bluetooth e Wi-Fi, que são capazes de transmitir uma quantidade razoável de dados por curtas distâncias, na grandeza de metros. Para sistemas que necessitam de transmissão para maiores distâncias, opções conhecidas são sinais de rádio e redes de telefonia, que por ainda dar conta de propagar uma certa quantidade de dados, acarretam maiores custos de infraestrutura e manutenção. Neste cenário de aplicações IoT, que em geral não se utilizam de grande volume de dados mas sim de transmissões de grande alcance, como é o caso deste trabalho, se popularizou uma categoria de protocolos de comunicação chamada Rede de Baixa Potência e Grande Distância, do inglês *Low Power Wide Area Network* (LPWAN).

Redes de baixo consumo e para grandes áreas, LPWANs são redes de comunicação que vem sendo muito utilizadas no mercado IoT devido à sua capacidade de transmissão de da-

dos pequenos à grandes distâncias com baixo consumo de bateria, proporcionando maior autonomia de energia e menos custo (RAZA; KULKARNI; SOORIYABANDARA, 2017). Podendo alcançar até 10km em campo aberto, as redes dessa categoria são capazes de transmitir um grande volume de dados de pequeno tamanho com baixa latência, permitindo a conexão em uma mesma rede de milhares de dispositivos, que por utilizarem módulos de comunicação otimizados com baixo consumo de energia e alta durabilidade, diminuem a necessidade de manutenção, acarretando diretamente a redução dos custos de implantação e de infraestrutura, sendo essa também uma das grandes vantagens das LPWANs.

Tecnologias LPWAN tipicamente utilizam a topologia em estrela, de modo a centralizar o acesso dos dispositivos conectados a um ponto de acesso, comumente nomeado *gateway*. O *gateway*, por funcionar como estação base, pode estar conectado à outras redes, portanto é responsável por interceptar as mensagens enviadas pelos dispositivos finais e realizar a tradução para os protocolos desejados, como o TCP/IP para integração com servidores na nuvem. Em geral, essa arquitetura também inclui a camada dos servidores, tendo inclusive a separação do nível de rede do nível de aplicação, disponibilizando servidores separados para estes dois, de modo a proporcionar isolamento para melhor escalabilidade, provisionamento e confiabilidade. Esse modelo de arquitetura é também um dos fatores que proporciona a redução dos custos de operação, pois a infraestrutura pode ser estrategicamente planejada com o posicionamento dos *gateways* e instanciamento dos servidores.

Dentro da categoria LPWAN, encontram-se diversas tecnologias, cada uma tendo seus diferenciais quanto ao alcance, capacidade de transmissão, faixas licenciadas ou não, cobertura pública ou privada. Os nomes mais populares no tema são LoRa, NB-IoT e SigFox, todos crescendo cada vez mais a nível global.

2.4.1 LoRa

Uma das tecnologias integrantes das LPWAN mais conhecidas, Longo Alcance, do inglês *Long Range* (LoRa) é uma tecnologia de rádio frequência utilizada na camada física que faz uso a técnica de modulação do espectro de espalhamento aplicando o formato Espalhamento Espectral Chirp, do inglês *Chirp Spread Spectrum* (CSS). Essa técnica de espalhamento espectral, como discutido em (REYNDERS; POLLIN, 2016), utiliza pulsos chirp de frequência linear de banda larga para codificar dados, propiciando a transmissão de mensagens com baixo consumo de energia, mas também com baixa taxa de dados. Sua operação acontece em faixas de rádio frequência não licenciadas na grandeza de sub-gigahertz, como por exemplo 915MHz, a faixa da Austrália e América do Norte, mas também utilizada no Brasil (The Things Network, 2022b).

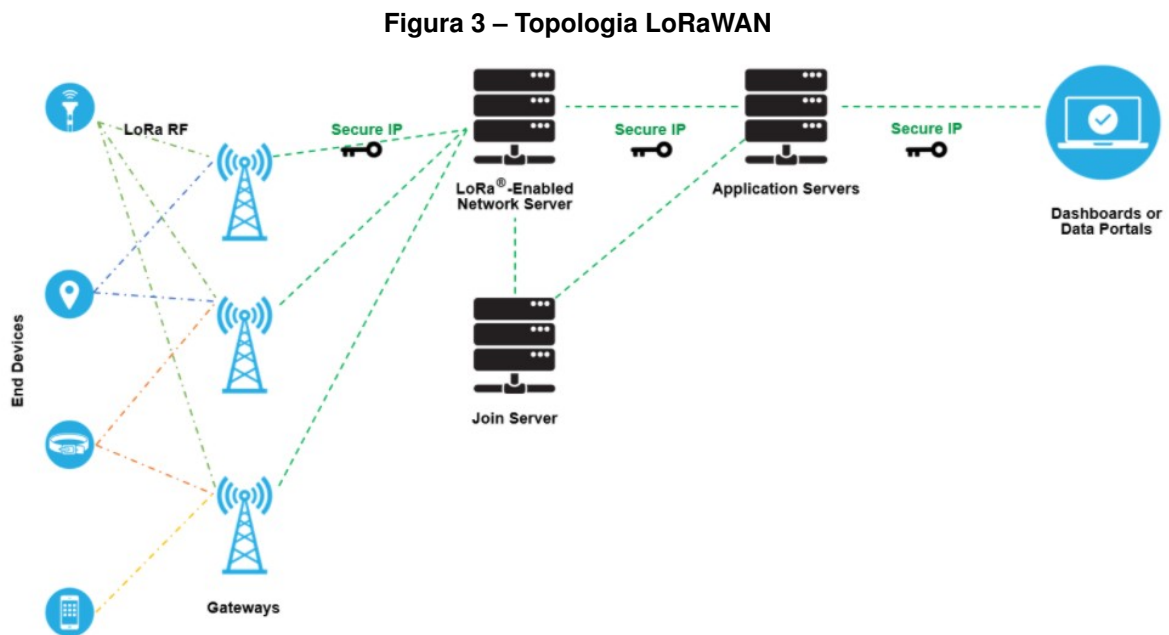
Suas principais vantagens, além da transmissão de grande alcance, são o baixo custo de operação devido à praticidade da infraestrutura e do baixo consumo de energia, oferecendo um bom compromisso entre consumo e alcance. Além disso, também é incluída em seus benefícios

a manutenção da transmissão de dados através da conexão com dispositivos em movimento, o que proporciona uma infinidade de oportunidades de aplicações (Semtech, 2022a). Outro ponto de destaque para essa tecnologia é obtenção de dados de geolocalização através da combinação de *Received Signal Strength Indication* (RSSI) e multilateração *Time Difference of Arrival* (TDoA).

LoRa é uma tecnologia proprietária, pertencente à empresa Semtech (Semtech, 2022b), que define um protocolo apenas de camada física. Para sua utilização, foi necessário desenvolver protocolos para definir as camadas restantes, e com isso surge o LoRaWAN.

2.4.2 LoRaWAN

De nome semelhante, Rede LoRa de Longa Distância, do inglês *LoRa Wide Area Network* (LoRaWAN) é o protocolo que implementa uma arquitetura de rede utilizando a modulação LoRa. LoRaWAN é uma camada de software, mais especificamente um protocolo de camada de controle de acesso ao meio, que define como os dispositivos podem utilizar um módulo LoRa e se responsabiliza pela transmissão e formatação das mensagens (Semtech, 2022a). Desenvolvido pela empresa Lora Alliance (LoRa Alliance, 2022), uma organização sem fins lucrativos fundada pela Semtech em 2015, esse protocolo, como uma LPWAN, utiliza uma arquitetura em estrela, composta pelos dispositivos finais, *gateways* e servidores de rede e de aplicação. Inclusive, alguns autores preferem a nomenclatura "estrela de estrelas" para sua topologia, devido à utilização dos *gateways*. A topologia LoRaWAN e seus componentes podem ser visualizados abaixo na Figura 3.



Fonte: (Semtech, 2022b).

Um dos principais pontos que acarretaram sua popularidade é a possibilidade de implementação de redes privadas e a facilidade de integrá-la com plataformas de diferentes provedores. Além disso, LoRaWAN também possui segurança de ponta a ponta, através da utilização de criptografia AES-128. Desse modo, essa rede de comunicação sem fio se difundiu no mercado mundial, estando presente nos setores de agricultura, logística, indústria, cidades inteligentes, e etc. Aliás, a LoRa Alliance anunciou em 2021 que LoRaWAN foi aprovada pela a Internacional Telecommunication Union (International Telecommunication Union (ITU), 2021) para se tornar o padrão internacional para LPWANs.

A especificação LoRaWAN define três tipos de dispositivos: Classe A, B e C. A classe A deve ser implementada por todos os dispositivos finais

Atualmente, LoRaWAN possui cobertura em 173 países através da LoRa Alliance, e esse número se expande cada vez mais. No Brasil, a cobertura LoRaWAN é majoritariamente provida pela empresa American Tower (American Tower, 2021), que chegou ao país em 2018 e, após 4 anos construindo mais de 23 mil torres e desenvolvendo sua infraestrutura para mais de 130 fornecedores, já alcançou mais de 280 cidades. Além disso, no Brasil há também a presença do projeto The Things Network (The Things Network, 2022), uma iniciativa colaborativa global que busca expandir a rede de cobertura e utilização de LoRaWAN no mundo através de *crowdfunding* e parcerias. Presente no Brasil desde 2015, a The Things Network possui atualmente 54 *gateways* conectados ao redor do país, todos mantidos pela comunidade, proporcionando uma alternativa mais acessível para trabalhos com essa rede. A Figura 4 ilustra um mapa localizando todos os *gateways* ativos da The Things Network (TTN) no Brasil.

Figura 4 – Cobertura The Things Network no Brasil



Fonte: The Things Network (2022).

2.5 GPS

O sistema de posicionamento global, tradução de sistema de Posicionamento global, do inglês *Global Positioning System* (GPS), é um sistema de navegação por satélite capaz de obter a localização de um receptor em qualquer lugar do mundo, desde que este esteja no alcance de no mínimo três satélites, para possibilitar o cruzamento por triangulação. Atualmente, existem 24 satélites orbitando o planeta divididos em seis planos orbitais, de modo que cada plano possui pelo menos quatro satélites, o que proporciona maior precisão nas medições.

Além de informações sobre localização, com coordenadas de latitude e longitude, o sinal GPS também é capaz de obter dados sobre a velocidade do receptor, altitude em relação ao mar e também dados de data e horário. Esse sistema possui um alto grau de precisão, tendo margem de erro da grandeza de metros na localização, e atraso no horário na margem de 1 segundo a cada 100 mil anos (TechTudo, 2011) .

3 PRODUTOS RELACIONADOS

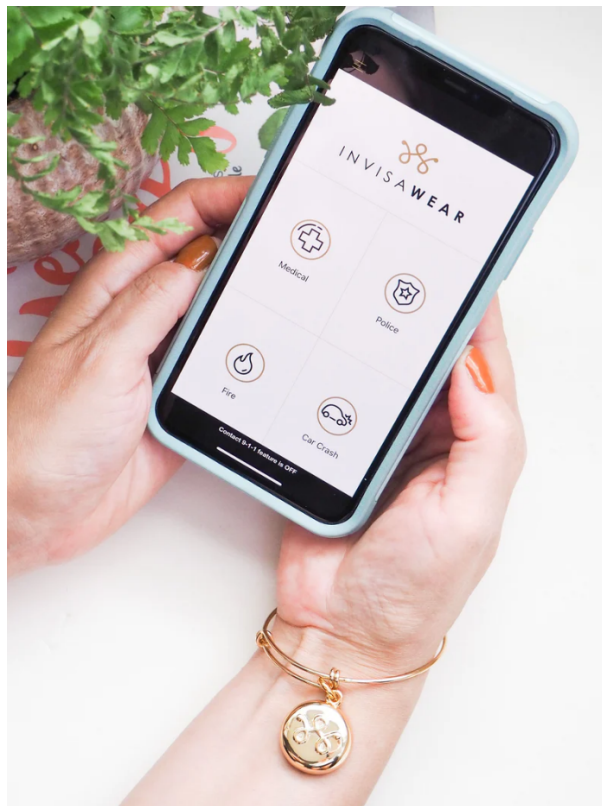
O protótipo de produto descrito nesse projeto, ainda que propondo apresentar uma iniciativa com de tecnologias modernas, teve seu conceito inspirado em produtos semelhantes já existentes, alguns dos quais serão referenciados neste capítulo.

3.1 invisawear

A maior inspiração para este projeto, invisawear (invisawear, 2022) é uma empresa estadunidense que comercializa jóias e acessórios que funcionam como um dispositivo de segurança. O dispositivo se passa por uma jóia normal e discreta, mas na realidade contém um botão que, ao ser pressionado, envia para até cinco contatos de emergência pré-selecionados um link para a observação da geolocalização do usuário, de modo a informá-los que este precisa de ajuda. Também há a possibilidade de ativar a funcionalidade de se conectar com a ADT, um serviço de monitoramento e segurança (ADT, 2022), e com a polícia.

A Figura 5 mostra um exemplo de jóia invisawear, nesse caso uma pulseira dourada, e o aplicativo de celular.

Figura 5 – Pulseira e aplicativo invisawear



Fonte: (invisawear, 2022).

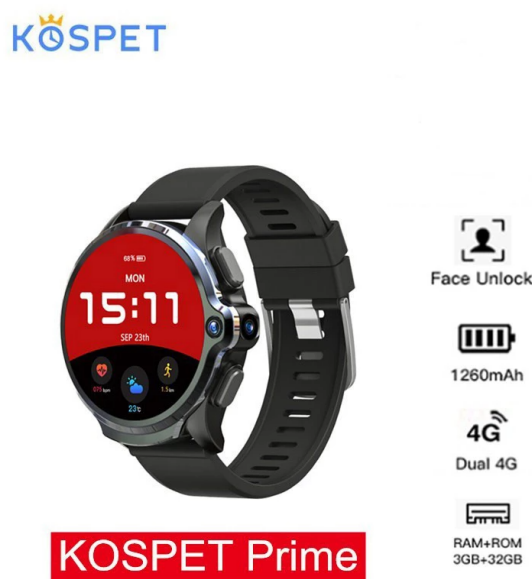
O funcionamento do produto é completamente dependente da disponibilidade de um telefone celular. O dispositivo se conecta com o celular através da rede Bluetooth, e essa conexão

é necessária para o envio do SOS, sendo que a jóia serve apenas como ativador. A obtenção dos dados de localização e construção da mensagem acontecem inteiramente no aplicativo no celular. Considerando essa arquitetura, identifica-se uma possível brecha para problemas, pois não são improváveis os cenários de o telefone do usuário estar indisponível, seja por falta de bateria ou até mesmo por não estar próximo o suficiente para estabelecer conexão, e como o propósito do produto é ser uma fonte de segurança em situações de emergência, essa brecha pode ser preocupante.

3.2 SmartWatches

Amplamente conhecidos, SmartWatches são relógios inteligentes que podem chegar a ser quase um celular a ser utilizado no pulso. A maioria dos modelos incluem conexão com o celular para a utilização da rede e proporcionar o uso de diversos apps, inclusive de mensagens. Porém modelos mais avançados, como o KOSPET Prime 4G SmartWatch, que pode ser visto na Figura 6, incluem até comunicação por rede de telefonia (o que implica na necessidade de cartão SIM).

Figura 6 – Relógio KOSPET Prime 4G



Fonte: (AliExpress, 2022).

Esses dispositivos podem ser um meio bem mais prático que o celular para envio de uma notificação em uma situação de perigo, mas ainda acaba existindo a dependência de disponibilidade e proximidade com o celular, e os modelos independentes acabam sendo mais caros e inacessíveis. Além disso, por terem uma estética digital e já amplamente conhecida, são acessórios que acabam chamando atenção de assaltantes e pessoas mal intencionadas, possivelmente causando insegurança em seu uso.

3.3 Claro Kids On

Seguindo a mesma proposta de notificação em situações de emergência, mas dessa vez focado no público infantil, o relógio/celular Kids On (Claro, 2022) promete tranquilidade aos pais através de um dispositivo a ser utilizado no pulso das crianças. O produto oferece funcionalidades de envio de localização com o aperto de um botão SOS, notificação de movimento com GPS entre zonas de segurança pré-definidas, e até a realização de ligações através da rede de telefonia, tudo por meio de um design simples e de fácil entendimento para crianças, o qual pode ser conferido na Figura 7.

Figura 7 – Relógio Claro Kids On



Fonte: (Claro, 2022).

O produto tem a vantagem de se utilizar da rede de telefonia, o que confere maior autonomia de seu uso. Porém, por ser um adicional ofertado exclusivamente pela operadora Claro, sua utilização deve ser necessariamente associada ao consumo de um plano, o que é um limitador para possíveis usuários que não tenham interesse em serem clientes da operadora.

A apresentação desses produtos visa apresentar exemplos de produtos que já existem hoje no mercado, expondo seus benefícios mas também analisando possíveis pontos de melhoria, e é destes últimos que este projeto buscou inspiração. Dando ênfase no desejo de construir um acessório que possa ser utilizado de forma discreta no dia a dia que ofereça maior sensação de segurança ao usuário através da utilização de meios de comunicação menos suscetíveis a falhas e dependência de ações do usuário, esse trabalho culminou no método apresentado no Capítulo 4.

4 METODOLOGIA

Neste capítulo, é descrito o processo de planejamento e desenvolvimento do projeto, iniciando com a descrição dos requisitos estabelecidos e a definição do sistema capaz de cumprí-los, seguindo para exploração dos materiais e ferramentas disponíveis para utilização e a justificativa para suas escolhas, finalizando com explicações e relatos sobre como foi o desenvolvimento do projeto.

4.1 Requisitos

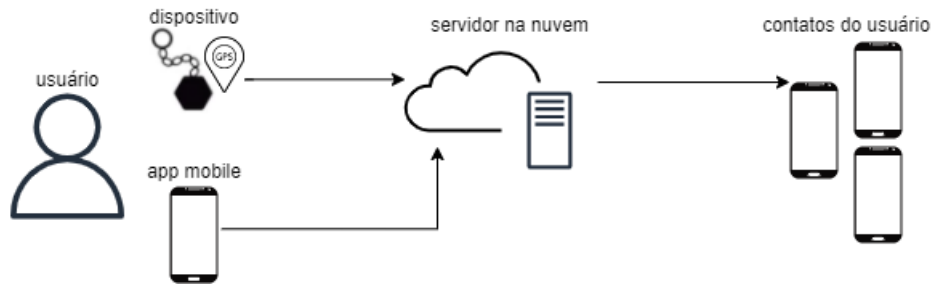
Para cumprir com os objetivos almejados, foram definidos os seguintes requisitos para este projeto:

- RF1: O sistema deve realizar o envio de uma mensagem de SOS apenas com o apertado de um botão.
- RF2: O sistema deve realizar o envio de uma mensagem de SOS sem a necessidade de interação com algum telefone celular.
- RF3: O sistema deve incluir na mensagem de SOS as coordenadas de localização do usuário no momento do envio.
- RF4: O sistema deve realizar o envio da mensagem de SOS para todos os números incluídos na lista de contatos do usuário.
- RF5: O sistema deve permitir que o usuário personalize sua lista de contatos para receber suas mensagens de SOS.
- RF6: O sistema deve permitir que o usuário associe seu dispositivo IoT à sua conta no aplicativo.
- RF7: O sistema deve permitir que o usuário visualize seu histórico de envio de mensagens de SOS.
- RNF1: O sistema deve possuir um dispositivo de tamanho reduzido (na dimensão de centímetros) que permita ao usuário transportá-lo com facilidade.

4.2 Arquitetura

Para atender aos requisitos definidos, foi feito o planejamento da arquitetura do sistema considerando os componentes existentes e os meios de comunicação entre eles. O diagrama abaixo ilustrado na Figura 8 representa o sistema a nível de contexto, contemplando os componentes existentes e como interagem entre si.

Figura 8 – Diagrama de contexto do sistema



Fonte: Autoria própria (2022).

Nesse diagrama são ilustrados os 2 fluxos de interação do usuário no sistema: o envio da mensagem de SOS e a edição de suas informações através do aplicativo móvel.

Após ter a clareza da estrutura esperada, foi possível realizar um estudo das ferramentas e tecnologias disponíveis para obter uma análise de quais seriam as melhores escolhas. Como este trabalho se trata de um sistema IoT, pode-se dividir sua composição em: uma plataforma, um protocolo de comunicação e um servidor. Além disso, a composição do projeto também inclui o aplicativo e os provedores do envio da mensagem.

4.2.1 Comunicação

Considerando haver o objetivo de enviar mensagens remotamente sem a necessidade de conexão do dispositivo com a internet ou com um telefone celular, surge uma limitação nas opções de protocolos de rede sem fio, mas a necessidade de envio de dados a longas distâncias direciona as pesquisas para LPWANs. Foram realizadas análises com algumas candidatas, como LoRa, NB-IoT e SigFox, considerando os fatores de custo, capacidade, cobertura e principalmente disponibilidade no local de desenvolvimento, e a escolhida foi a tecnologia LoRa e sua aplicação com o protocolo LoRaWAN.

4.2.1.1 LoRaWAN

Os principais pontos que propiciaram a escolha dessa rede foram o fato de LoRaWAN ser ideal para a transmissão de pacotes de dados de pequeno tamanho, possuir longo alcance de transmissão (até 3km em áreas urbanas e 10km em áreas rurais), ter baixo consumo de energia e ser capaz de realizar a transmissão de dados enquanto o transmissor está em movimento, além de ser *open source*.

4.2.1.2 The Things Network

Foram realizadas pesquisas de provedores LoRaWAN capazes de oferecer os recursos necessários para o projeto, considerando principalmente custos e cobertura na cidade de

Curitiba. De modo a reduzir ao máximo os custos desde projeto, o escolhido foi o provedor The Things Network, um ecossistema colaborativo com uma comunidade relevante utilizado no mundo todo, contando com alguns *gateways* ativos em Curitiba, um deles sendo de posse do Laboratório de Sistemas de Comunicações (LabSC) da Universidade Tecnológica Federal do Paraná (UTFPR). A The Things Network possui o The Things Stack, um servidor de rede LoRaWAN que permite a criação de aplicações, configuração de *gateways* e de dispositivos finais, além de integrações externas, tudo disponibilizado dentro de um limite gratuito de utilização dos recursos.

A propósito, um item de grande relevância para o projeto é a possibilidade de integrações externas, pois é necessário encaminhar a mensagem recebida pela The Things Network para um servidor na nuvem. Nesse tópico, a TTN possui diversas de opções disponíveis para integrações IoT considerando diferentes protocolos e serviços. Dessas opções, foi escolhido o protocolo MQTT, pois com sua eficiência e redução do tamanho na transmissão de mensagens, alta escalabilidade e confiança de entrega e utilização de comunicação assíncrona, todas as mensagens enviadas pela TTN podem ser recebidas e armazenadas até que possam ser consumidas por um servidor, garantindo maior integridade e confiabilidade ao projeto.

4.2.2 Plataforma

De forma a atender o protocolo de comunicação MQTT, foram realizadas pesquisas para avaliar qual seria o hardware mais adequado para ser implementado como o dispositivo do projeto. Os requisitos incluíam: comunicação LoRaWAN, a capacidade de obter sua geolocalização, e ter um tamanho reduzido. Considerando o escopo do projeto, foi decidido descartar a alternativa de obter módulos individuais e construir uma placa personalizada, sendo preferido obter um hardware já pronto com todos esses recursos. Após avaliar quesitos de preço, qualidade e complexidade de utilização, a escolhida foi a placa TTGO T-Beam (LilyGO, 2020).

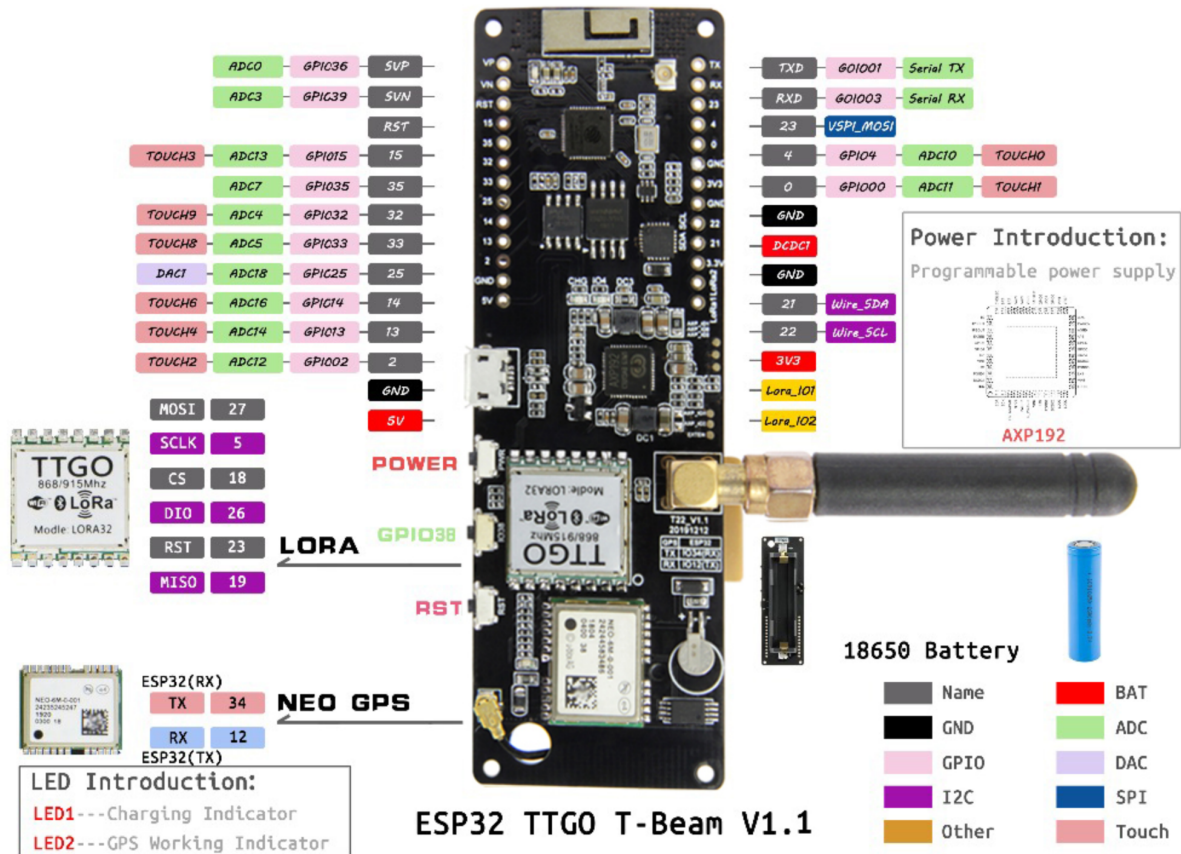
4.2.2.1 TTGO T-Beam

TTGO T-Beam é uma placa da família ESP32, e seus principais atributos conforme a documentação (RIOT, 2022) são a disposição de um transceptor LoRa SemTech SX1276 e um receptor GPS U-Blox Neo-6, além de seu tamanho pequeno, com 10cm de comprimento e 3cm de largura. Foi adquirida a versão v1.1 (mais atual) no modelo com opção de frequência de comunicação na faixa 915MHz-928MHz, o plano de frequência utilizado para LoRaWAN no Brasil (The Things Network, 2022b).

A Figura 9 mostra uma ilustração da placa TTGO T-Beam acompanhada de sua pinagem.

Uma das principais vantagens da placa escolhida, além de atender os requisitos necessários, foi a familiaridade com a programação de placas ESP. A programação do *firmware*

Figura 9 – Pinout da placa TTGO T-Beam v1.1



Fonte: (LilyGO, 2022).

foi feita com a linguagem C++ através de IDE Arduino, utilizando a biblioteca LoraWAN-in-C (LMIC) (mcci-catena, 2022) para conexão com o servidor TTN via LoRaWAN utilizando ativação Over the Air Activation (OTAA) (The Things Network, 2022a), e a biblioteca TinyGPS++ (mikalhart, 2022) para a leitura do sinal GPS. Para testes externos, foi utilizado como alimentação um *power bank* comum, dado que supre a necessidade de locomoção e praticidade.

Como meio de ativação do dispositivo com o servidor TTN, foi optado pelo OTAA ao invés do Activation By Personalization (ABP) pois o primeiro, que utiliza a atribuição de um endereço dinâmico com a identificação do dispositivo através de chaves de segurança, é o método mais seguro e recomendado comparado ao segundo, que necessita ter o endereço do dispositivo fixo no código.

Sobre o interesse em ter um dispositivo pequeno, na dimensão de ser utilizado como jóia ou chaveiro, é reconhecido que o hardware escolhido não atende aos critérios no nível desejado. Porém é importante ressaltar que o projeto se trata de um protótipo com escopo de experimentação, e certamente em evoluções futuras será avaliada a possibilidade de criação de um hardware personalizado e projetado para ter tamanho reduzido.

4.2.3 Servidor

O servidor representa a aplicação que é responsável por armazenar os dados referentes aos usuários e realizar a coordenação das mensagens enviadas pela TTN de modo a serem enviadas para os telefones corretos. Essa aplicação pode ser qualquer tipo de *web server* que esteja disponível na nuvem. Considerando o escopo do projeto e que o objetivo a ser atingido independe de implementações de baixo nível, foi favorecida a opção de utilizar um provedor de serviços de nuvem à opção de implementar uma aplicação do zero, pois o esforço de arquitetar e desenvolver um software e sua infraestrutura implicam em grande gasto de tempo comparado à alternativa de utilizar serviços prontos a baixo ou nenhum custo para os recursos exigidos.

Portanto, foi feita uma pesquisa dos provedores de serviços em nuvem disponíveis, e tanto pela quantidade de soluções ofertadas, quanto pelo interesse pessoal em aprofundar conhecimentos e práticas neste provedor, o escolhido foi Amazon Web Services (AWS).

4.2.3.1 AWS

Amazon Web Services (AWS) é uma das maiores plataformas de serviços de computação em nuvem, tendo servidores hospedados em vários locais distribuídos pelo mundo, com uma infinidade de categorias que vão desde computação e infraestrutura até engenharia de dados e inteligência artificial, ofertando diversas faixas de preços conforme o consumo.

Para construir um servidor capaz de suprir as necessidades do projeto, foram utilizados os seguintes serviços (Amazon Web Services, 2022a):

- DynamoDB: banco de dados NoSQL performático e escalável, sendo utilizado para armazenar os dados dos usuários e das mensagens de SOS;
- Lambda: funções *serverless* escaláveis e com infraestrutura de alta disponibilidade, sendo utilizadas para integrar os serviços configurados;
- IoT Core: serviços de integração e comunicação entre dispositivos IoT e a cloud, sendo utilizados para hospedar o broker MQTT;
 - *Rules*: formas de manipular mensagens recebidas no broker MQTT de forma a configurar interações com outros serviços AWS já pré-disponíveis;
 - *Actions*: ações a serem configuradas dentro de cada Rule, podendo definir gatilhos de ativação de outros serviços e repasses de mensagens;
- *Cloud Formation*: serviço de provisionamento e configuração de *stacks* de recursos a serem gerenciados como uma unidade, sendo utilizado para a criação de diversas Lambdas, *rules* MQTT, criptografia e *roles* de segurança, grupos de *logs* para a integração com a The Things Network;

- *Simple Notification Service* (SNS): serviço que permite o envio de notificações para usuários finais, sendo utilizado para o envio de SMS com a mensagem de SOS;
- AppSync: serviço GraphQL que permite integrar bases de dados a uma API disponibilizada para serviços externos, sendo utilizado como API para leitura/escrita de dados por parte do aplicativo móvel;
- *Identity and Access Management* (IAM): serviço de segurança para controle e gerenciamento de acessos e permissões entre serviços internos e externos. Considerando que o projeto se restringiu a um escopo de protótipo, o IAM foi utilizado de forma básica para permitir integrações entre os serviços, pois não houve necessidade de grande complexidade de permissionamento;
- *CloudWatch*: serviços de monitoramento confiáveis e escaláveis, sendo utilizados para depuramento e acompanhamento dos testes do sistema durante o desenvolvimento;

Cada um desses recursos possui sua própria especificação de precificação, que pode levar em conta a quantidade de dados utilizada, o tempo de conexão ativa, uma relação entre quantidade de recursos por período de tempo e até mesmo a região de hospedagem dos servidores utilizados. Neste projeto, foi considerado que os objetivos de prototipagem e testes poderiam ser alcançados dentro da utilização dos planos do nível gratuito ofertados pela AWS (Amazon Web Services, 2022e). Uma grande vantagem desse modelo de consumo é a facilidade de escalar os recursos conforme necessário para determinado volume de usuários, o que seria muito conveniente no caso de aplicação real do produto. Porém, essa utilização deve ser planejada e analisada com cautela para levar em conta os gastos decorrentes de maior consumo, que pode se tornar uma operação bem cara.

4.2.4 Envio de mensagens

4.2.4.1 WhatsApp Cloud API

Por ser de longe o aplicativo de mensagens mais utilizado no Brasil (PURZ, 2022), foi fácil escolher o WhatsApp como provedor para envio de mensagens SOS no projeto. A plataforma WhatsApp Business (Meta, 2022a) possui uma API de nuvem que oferta serviços de mensageria para empresas, com planos de cobrança conforme número de conversas. Para cumprir com os requisitos do projeto e possibilitar a demonstração, criar um aplicativo na plataforma da Meta Developers (Meta, 2022c) no modo desenvolvimento foi suficiente, o que permitiu criar um *template* de mensagem personalizado e enviá-lo através da API do WhatsApp para números de telefone cadastrados e autorizados individualmente.

4.2.4.2 SMS

Apesar de não ser largamente utilizado atualmente, o Serviço de Mensagens Pequenas, do inglês *Short Message Service* (SMS) foi também incluído no projeto por, diferente dos aplicativos de mensagem, não utilizar a internet para envio das mensagens. Desse modo, caso os contatos do usuário não estejam com seus celulares conectados na internet, a mensagem de SOS ainda poderia ser entregue através da rede de telefonia móvel, que costuma ter grande cobertura.

O envio de mensagens SMS foi configurado através do serviço SNS ofertado pela AWS, que utiliza integrações com as redes de telefonia brasileiras para a transmissão das mensagens, o que inclui a cobrança de taxas (Amazon Web Services, 2022c).

4.2.5 Aplicativo móvel

Por fim, de modo a oferecer ao usuário uma interface para o gerenciamento de seus dados e configuração do sistema, foi decidido desenvolver um aplicativo de celular. Considerando as ferramentas disponíveis para desenvolvimento (um celular Android) e os conhecimentos de experiências prévias, optou-se pela utilização do React Native (Meta, 2022b), uma biblioteca Javascript *open source*, criada pela Meta, para desenvolvimento híbrido de aplicativos Android, iOS e outros sistemas operacionais. Essa biblioteca utiliza conceitos de programação funcional junto com estados e *hooks*, de forma a criar componentes de interface de usuário robustas e performáticas.

Além disso, para consumir a API GraphQL (The GraphQL Foundation, 2022) do servidor AWS, foi utilizada a biblioteca Amplify (Amazon Web Services, 2022b) uma ferramenta disponibilizada para facilitar a integração de aplicativos com os serviços da nuvem.

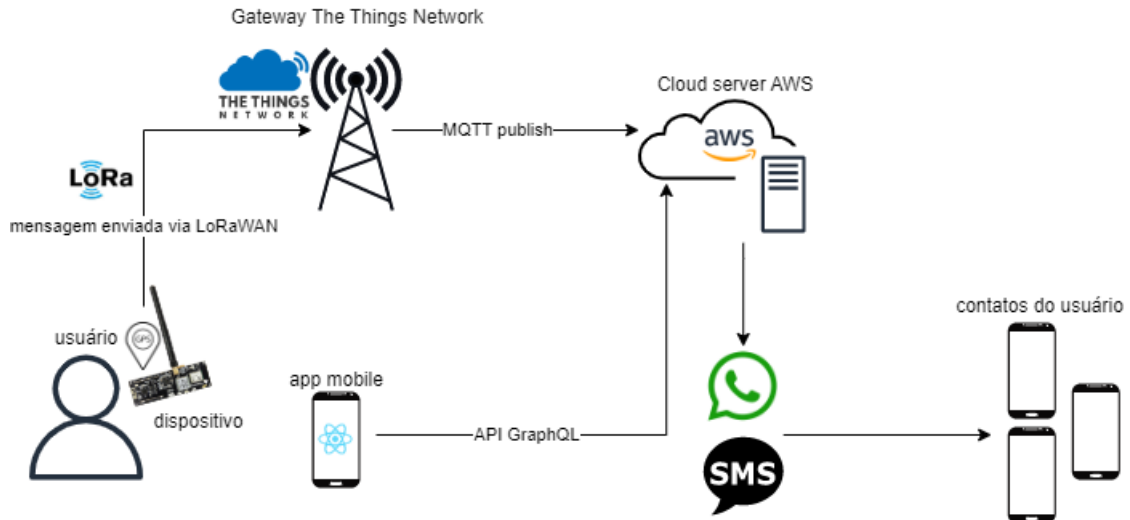
É importante reforçar que o aplicativo é utilizado somente para a configuração das informações do usuário em casos de edição, não configurando qualquer dependência no processo de envio de mensagens de SOS.

4.2.6 Definição dos componentes

Tendo-se decidido todos os materiais e ferramentas, foi possível atualizar o diagrama de contexto do sistema para contemplar todos os componentes agora definidos, como pode ser visto na Figura 10.

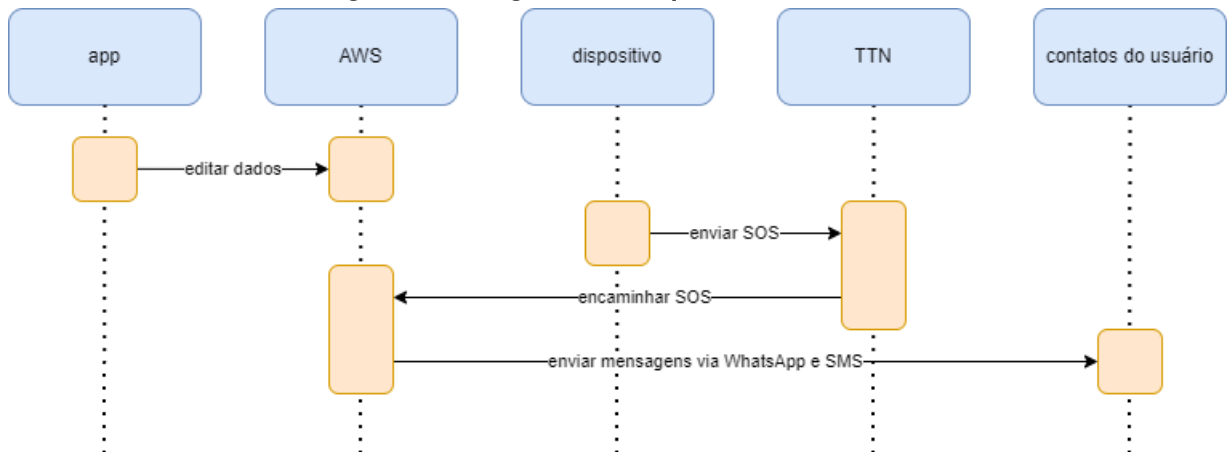
Os fluxos podem ser melhor visualizados no diagrama de sequência ilustrado na Figura 11.

Figura 10 – Diagrama de contexto do sistema com materiais e componentes definidos



Fonte: Autoria própria (2022).

Figura 11 – Diagrama de sequência do sistema



Fonte: Autoria própria (2022).

4.3 Implementação e desenvolvimento

O desenvolvimento de cada um dos componentes do sistema foi realizado sequencialmente acompanhado de testes constantes.

4.3.1 Firmware

O desenvolvimento do projeto foi iniciado pelo *firmware* para priorizar os testes de *hardware* o quanto antes, de forma a ter tempo para tomar ações caso algo estivesse com defeito. Ambos os testes do módulo GPS e LoRa se basearam no código do projeto T-Beam Tracker for The Things Network (kizniche, 2022).

O GPS apresentou problemas para captar sinal inicialmente, mas após um *reset* de suas configurações utilizando a biblioteca *SparkFun's Ublox Arduino* (eriktheV-king, 2020), foi possível ler os dados de geolocalização com sucesso.

Os testes do módulo LoRa foram feitos já validando a conexão com a the Things Network utilizando o gateway do LabSC. Foi preciso definir a biblioteca LMIC (compilada localmente) para utilizar a sub-banda correta para a faixa de frequência AU915 (utilizada no Brasil).

Como o local de desenvolvimento se localizava fora do raio de alcance de algum *gateway* TTN ativo, o LabSC se dispôs a emprestar o *gateway* da universidade de forma a facilitar e agilizar o desenvolvimento do projeto remotamente.

4.3.2 The Things Network

Para realizar testes da comunicação LoRaWAN, foi preciso criar uma aplicação no console da The Things Stack e cadastrar a placa TTGO T-Beam como um dispositivo na aplicação, de forma a obter as credenciais que permitem com que o dispositivo ingresse na rede. Após verificar que ambos o *gateway* e a aplicação estavam utilizando a mesma instância de servidores da TTN, foi possível verificar se as mensagens enviadas pela placa estavam conseguindo alcançar a aplicação TTN com sucesso.

Para a integração MQTT, uma vantagem ofertada pela TTN foi a existência de uma integração com o AWS IoT Core, disponibilizando um *template* para a criação de todos os serviços necessários na AWS já prontos para serem conectados com a aplicação TTN. Dessa forma, a própria TTN garantiu a possibilidade da criação de uma *stack* no Cloud Formation para gerenciar toda a comunicação com o *broker* MQTT, incluindo o tratamento de mensagens *uplink* e *downlink* e o gerenciamento de dispositivos.

No contexto deste projeto, onde existe a disponibilidade de somente um dispositivo, sua configuração do console da TTN foi feita manualmente, mas pensando em evoluções futuras e até ponto de vista de produto, seria necessário buscar por automações ou APIs que facilitassem o gerenciamento e a realização em massa desse processo.

4.3.3 Servidor AWS

O desenvolvimento foi iniciado pela construção da API GraphQL com o Appsync e consequentemente a estruturação das entidades e da base de dados, de forma a ter disponível a entidade do usuário e seus dados para as etapas seguintes.

Na sequência, foram realizados testes do envio de SMS utilizando tópicos com o serviço SNS (disponibilizado pela AWS), e também testes do envio de mensagens WhatsApp através da API do WhatsApp Business Cloud. Para isso, foi preciso criar um projeto na plataforma do

Facebook: Meta for Developers, além da criação de um modelo de mensagem personalizado para o SOS.

Por fim, foi feita a configuração do IoT Core para consumir corretamente as mensagens dos tópicos de mensagens da TTN. Foi definido que seria necessário realizar três ações ao receber uma mensagem: registrar o dado do SOS no banco de dados, enviar a mensagem SOS para Whatsapp e enviar a mensagem SOS via SMS. Cada uma dessas ações foi implementada em uma função Lambda, e para aplicar esse fluxo foram utilizadas Rules do IoT Core.

4.3.4 Aplicativo móvel

A criação do aplicativo React Native e seu desenvolvimento contaram com testes diretamente no celular Android pessoal disponível. O *design* das interfaces seguiu protótipos criados previamente com o software Figma (Figma, 2022), visando obter uma interface de usuário de qualidade e boa usabilidade. A integração com a API GraphQL com a biblioteca Amplify já foi validada logo no início, de forma que a construção das telas contou com os dados reais, sem a necessidade de criar dados fictícios de teste.

4.3.5 Testes

Testes e validações foram realizados constantemente durante todo o processo de desenvolvimento, de modo a garantir que cada módulo estivesse funcional como uma unidade, antes de prosseguir para o desenvolvimento de um próximo item. Por exemplo, foi validada a execução do *firmware* exclusivamente na placa, e em seguida foi testada a comunicação do *firmware* com o servidor The Things Network através da rede LoRaWAN. Da mesma forma, durante a construção do projeto AWS, cada serviço foi testado individualmente, progredindo para validação das interações entre os serviços, como entre a Api GraphQL e o banco de dados, e as integrações entre o *broker* no IoT Core e funções Lambda. Na sequência foi possível validar a comunicação entre a aplicação TTN e o servidor AWS através do MQTT, além da integração com a API do WhatsApp.

Dessa forma, com a finalização do projeto, foi obtido um sistema consistente e de confiança, sendo possível focar apenas em testes de desempenho e de qualidade para possibilitar análises dos resultados obtidos, discorridos a seguir no Capítulo 5.

5 RESULTADOS

Neste Capítulo será apresentado o sistema obtido como resultado deste trabalho. As seções irão discorrer sobre cada um dos módulos do sistema, apresentar os resultados dos testes realizados, e falar brevemente sobre melhorias identificadas e possíveis evoluções.

5.1 Protótipo

Considerando o escopo do projeto e o objetivo de apresentar um protótipo funcional, a validação foi feita com uma placa de desenvolvimento e uma fonte portátil de energia. Como trabalho futuro sugere-se a criação de uma placa de circuito impresso que utilize componentes com dimensões reduzidas, mais adequadas para a integração em acessórios como os mencionados no Capítulo 3.

A versão final do dispositivo (a placa TTGO T-Beam), conectado a um *power bank* para alimentação, pode ser vista na Figura 12.

Figura 12 – Dispositivo final conectado à alimentação

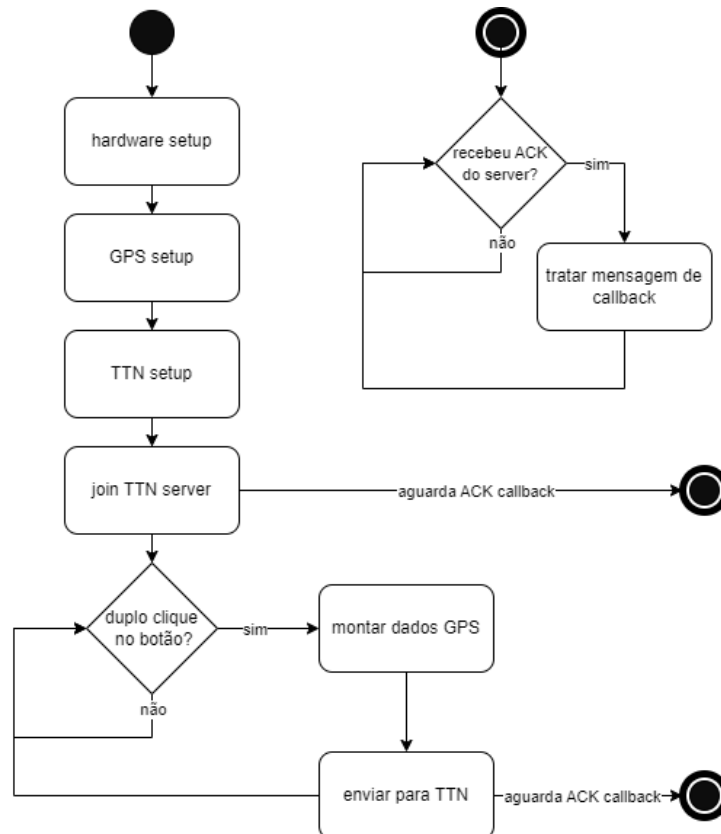


Fonte: Autoria própria (2022).

5.2 Firmware

O código carregado no *hardware* tem a responsabilidade de obter a geolocalização de onde se encontra e gerenciar o envio dos dados para o servidor The Things Network. Foi desenvolvido um *script* para implementar a utilização de duplo clique no botão da placa como meio de ativação por parte do usuário para o envio da mensagem. A execução do *firmware* pode visualizada na Figura 13 através de um diagrama de atividades.

Figura 13 – Diagrama de atividades do *firmware*



Fonte: Autoria própria (2022).

O código de inicialização realiza diversas configurações básicas, como a inicialização do sistema de gerenciamento de energia e a comunicação entre o GPS e o microcontrolador. Em seguida, é inicializado o Serial a ser utilizado para leitura do módulo GPS através dos pinos RX e TX. Por fim, é realizada a inicialização do Serial para a comunicação com o módulo LoRa e das credenciais para conexão com a aplicação TTN.

Em seguida, é realizado o processo de *join* com o servidor TTN. Isso envolve basicamente a seleção da sub-banda a ser utilizada pela biblioteca LMIC, a configuração da função de *callback* a ser chamada pela biblioteca ao receber uma resposta do *server*, e a chamada de função LMIC que realiza o *join* de fato. Se tudo ocorre corretamente, uma mensagem do tipo EV_JOINED é recebida pela função de *callback*, sinalizando que a tentativa de *join* do dispositivo foi recebida e aceita pela TTN.

Após esses primeiros passos, o código entra na etapa de *loop*, na qual o dispositivo fica no aguardo do aperto do botão. Caso o botão seja pressionado duas vezes no intervalo de 1 segundo, é acionado o código de envio da mensagem.

O primeiro passo é montar a mensagem a ser enviada contendo as informações da geolocalização. É realizada a leitura do Serial GPS para obter dados da latitude e longitude, aplicando algumas manipulações de bits para construir um *buffer* contendo todos os dados codificados em base binária. Isso é feito para otimizar o tamanho da mensagem a ser enviada via LoRaWAN.

Por fim, é realizado o envio da mensagem através de funções da biblioteca LMIC. Novamente, a função de *callback* fica no aguardo de uma confirmação de recebimento da mensagem no servidor TTN. Caso o envio tenha acontecido com sucesso, é recebida uma mensagem do tipo `EV_TXCOMPLETE` no *callback*.

É relevante destacar que, neste projeto, as mensagens recebidas no *callback* foram utilizadas apenas com objetivos de testes e depuração, mas possíveis evoluções poderiam se aproveitar desses dados para implementar mecanismos de tratamento de erros e reenvios automáticos.

5.3 Aplicação The Things Network

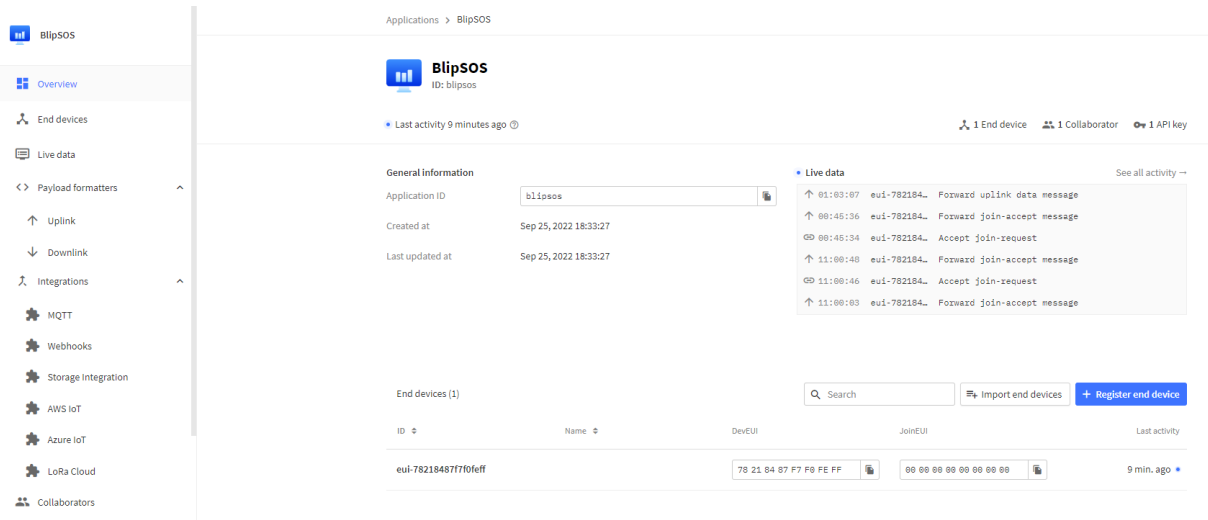
A aplicação criada no The Things Stack tem o objetivo de receber, formatar e redirecionar mensagens enviadas pelo dispositivo através da rede LoRaWAN do The Things Network. Como pode ser visto na Figura 14, a aplicação BlipSOS possui um dispositivo cadastrado (sendo este a placa TTGO T-Beam), cuja comunicação com o servidor TTN pode ser acompanhada pelo painel *Live data*. Nesse painel, é possível observar alguns exemplos de mensagens recebidas com solicitação para *join* no servidor e recebimento de mensagens *uplink* com os dados da geolocalização.

A Figura 15 mostra a tela do dispositivo cadastrado, incluindo alguns dados da configuração do dispositivo e chaves secretas, e novamente um painel *Live data* exibindo os *logs* de mensagens.

Como o *firmware* realizou a codificação dos dados, a aplicação TTN deve ser capaz de decodificá-los. Dessa forma, na seção *Payload formatters*, foi implementado um formatador de payload customizado com Javascript para dados de *uplink*, visando formatar as mensagens enviadas pelo dispositivo de forma a deixá-las no formato correto a ser publicado no tópico MQTT. O código para esse formatador foi disponibilizado pelo projeto TTGO T-Beam Tracker por kizniche (2022).

Quanto à integração MQTT, na seção *Integrations* foi configurada uma integração do tipo AWS IoT, como pode ser visto na Figura 16. Como a principal parte dessa integração é realizada na própria AWS, o trabalho a ser feito na aplicação TTN é bastante simples, se resumindo

Figura 14 – Console da aplicação no The Things Stack



Fonte: Autoria própria (2022).

à inserção do nome da stack no *Cloud Formation* da AWS e sua *role ARN*. Além disso, foi necessário criar uma *API Key* a ser utilizada na configuração da *stack* no *Cloud Formation*.

A arquitetura dessa integração, cujo *template* para criação dos recursos AWS é disponibilizado pela TTN, pode ser visualizada na Figura 17.

5.4 Servidor AWS

A aplicação AWS resultante consiste na interação e integração de diversos recursos. Para facilitar a visualização desse sistema resultante, foi desenhado o diagrama de componentes e comunicação da Figura 18, descrevendo ambos os fluxos do envio da mensagem SOS e da interação do aplicativo móvel.

Na Figura 19 pode ser observada a estrutura das tabelas do banco de dados. A tabela *UserTable* armazena os usuários do sistema com todos os dados pessoais, o id do dispositivo associado e uma lista de objetos contendo o nome e o telefone da lista de contatos de emergência. A tabela *DeviceTable* armazena a relação entre um *deviceld* como chave primária e um email, sendo necessária para a estruturação das rules no MQTT. Por fim, a tabela *SOSTable* armazena todos os SOS gerados, contendo o id do dispositivo, o email do usuário correspondente, a data e hora do disparo e um objeto contendo a latitude e a longitude da localização no momento do disparo.

Como pode-se observar na Figura 18, o broker MQTT, item 1 do diagrama, atua como principal orquestrador do fluxo. Ao receber os dados da mensagem da TTN, uma *rule* (item 2 do diagrama) foi configurada para obter o email correspondente a esse *deviceld* na tabela *DeviceTable*. Essa rule utiliza uma função SQL *built-in* capaz de obter um registro de uma tabela com base em um valor de chave primária, de modo a juntar essa informação aos dados já

Figura 15 – Dispositivo cadastrado

Applications > BlipSOS > End devices > eui-78218487f7f0feff

eui-78218487f7f0feff

ID: eui-78218487f7f0feff

↑ 2,575 ↓ n/a • Last activity 10 minutes ago

Overview
Live data
Messaging
Location
Payload formatters
Claiming
General settings

General information

End device ID: eui-78218487f7f0feff

Frequency plan: Australia 915-928 MHz, FSB 2 (used by TTN)

LoRaWAN version: LoRaWAN Specification 1.0.2

Regional Parameters version: RP001 Regional Parameters 1.0.2

Created at: Oct 5, 2022 23:21:14

Activation information

AppEUI: 00 00 00 00 00 00 00 00

DevEUI: 78 21 84 87 F7 F0 FE FF

AppKey:

Session information

Session start: Nov 30, 2022 01:03:07

Device address: 26 0D 5F 21

NwkSKey:

SNwkSIntKey:

NwkSEncKey:

AppSKey:

MAC data

Download MAC data

Live data See all activity →

- ↓ 01:03:07 Schedule data downlink for transmission on Gateway Server De
- ↑ 01:03:07 Forward uplink data message DevAddr: 26 0D 5F 21 <> Pa
- ↑ 01:03:07 Successfully processed data message DevAddr: 26 0D 5F 21 <>
- ↑ 00:45:36 Forward join-accept message

Location Change location settings →

No location information available

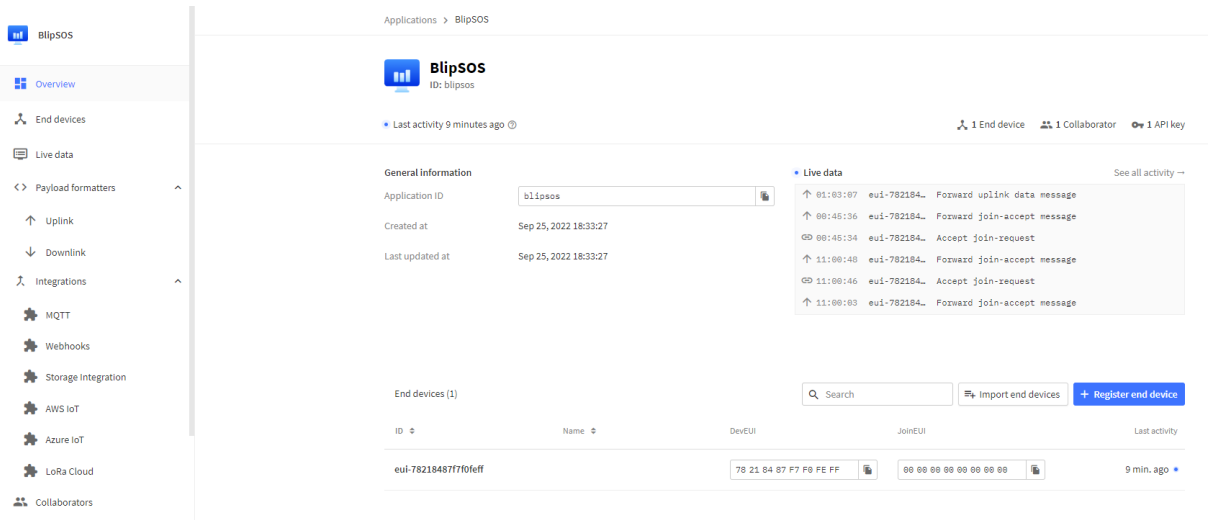
Fonte: Autoria própria (2022).

recebidos na leitura do tópico *lorawan/+/uplink*. Um exemplo de mensagem de *uplink* recebida no tópico é mostrada na Figura 20, e a consulta SQL utilizada nessa *rule* pode ser vista na Listagem 1.

Com isso, foram configuradas duas *actions* para essa *rule*: uma para chamar a função Lambda (item 5 do diagrama) responsável por tratar esses dados para a criação de um registro na tabela *SOSTable* no banco de dados (item 8 do diagrama), e outra para realizar um *republish* da mensagem no broker, dessa vez no tópico *sos*. Esse *republish* é necessário pois o formato da mensagem dessa *rule* de persistência do SOS é diferente do formato necessário para as *rules* de envio do SOS.

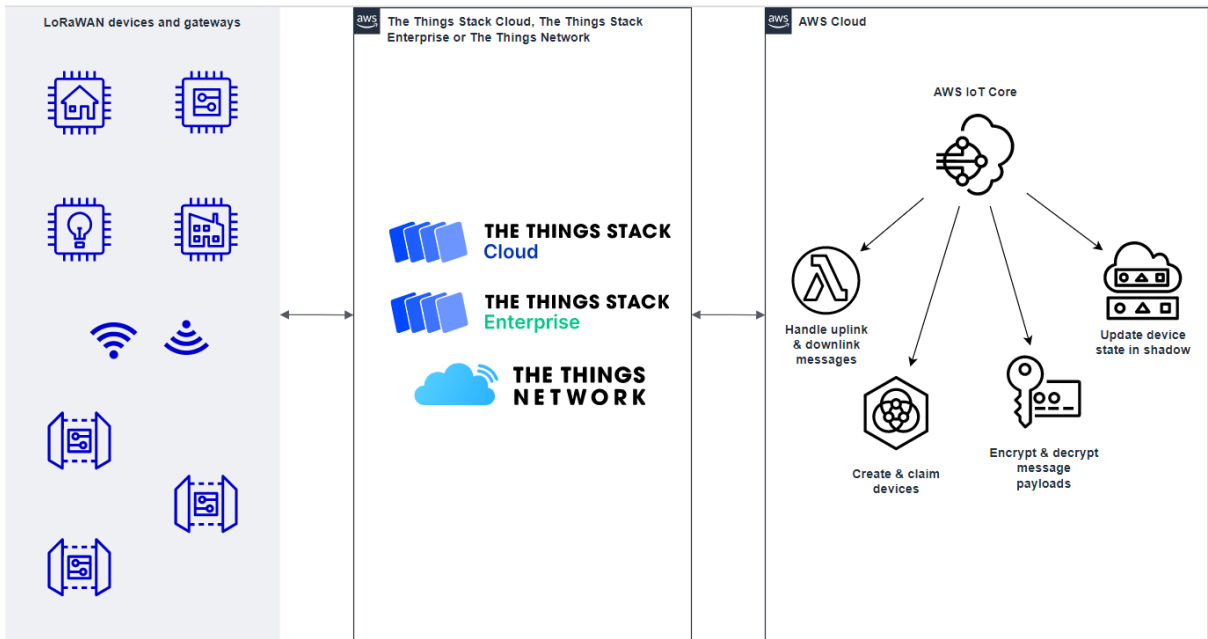
Em seguida, duas outras *rules*, os itens 3 e 4 do diagrama, são acionadas pela publicação no tópico *sos*, ambas realizando a consulta apresentada na Listagem 2, cada uma configurando *actions* que encaminham os dados do usuário e das coordenadas para duas funções Lambdas. A primeira (item 6) é responsável por formatar os dados na mensagem SOS e publicá-

Figura 16 – Interação AWS



Fonte: The Things Stack (2022).

Figura 17 – Integração The Things Stack e AWS

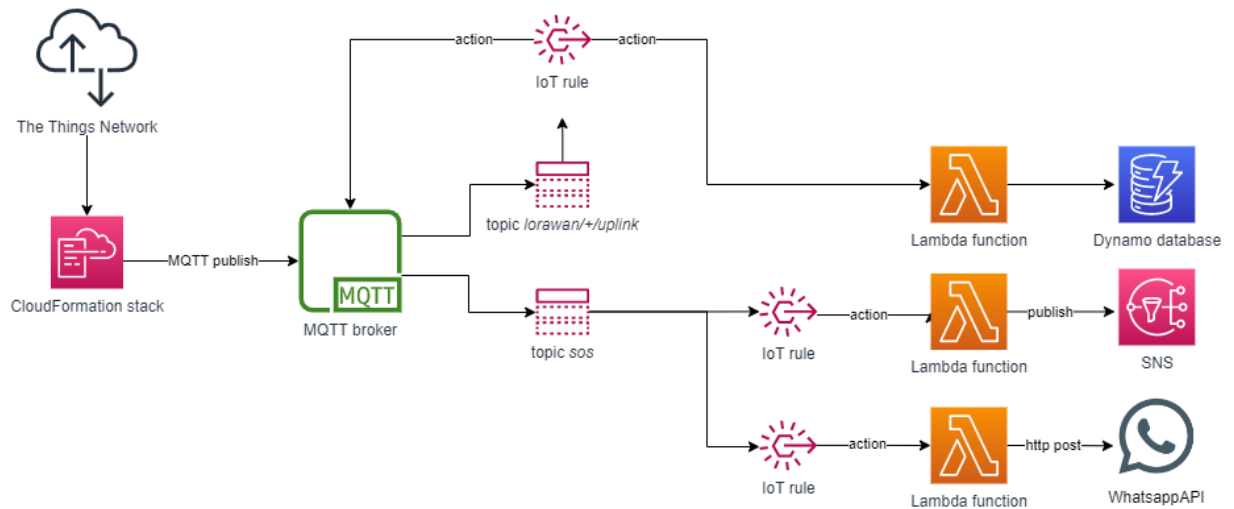


Fonte: The Things Network (2022).

la no serviço de mensageria SNS (item 9) no tópico correspondente ao usuário, de modo a enviar uma mensagem de texto via SMS para todos os números de telefones configurados para este usuário. A segunda (item 7) é responsável por, para cada telefone da lista de contatos, formatar os dados da mensagem SOS no payload de duas requisições HTTP POST para a API do WhatsApp (item 10): uma para enviar o texto de SOS e outra para enviar a localização.

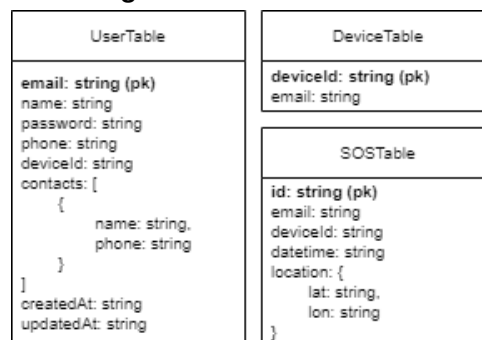
Todo esse fluxo do envio de uma mensagem SOS e as comunicações realizadas entre os serviços podem ser vistos no diagrama de sequência na Figura 21, iniciando com o gatilho

Figura 18 – Diagrama de componentes do projeto AWS



Fonte: Autoria própria (2022).

Figura 19 – Diagrama de tabelas do banco de dados



Fonte: Autoria própria (2022).

Listagem 1 – Rule handleSOS

```

1 SELECT
2 end_device_ids.device_id AS deviceId ,
3 uplink_message.decoded_payload.bytes.latitude AS lat ,
4 uplink_message.decoded_payload.bytes.longitude AS lon ,
5 get_dynamodb("DeviceTable", "deviceId", end_device_ids.device_id ,
6     "arn:aws:iam::{id}:role/IoTtest").email AS email
7 FROM "lorawan/+uplink "
```

Fonte: Autoria própria (2022).

na stack do Cloud Formation acionado pela The Things Network e finalizando com o envio das mensagens para os celulares da lista de contatos por SMS e pela API externa do WhatsApp.

Em relação ao envio do SMS, a mensagem foi definida na função Lambda, que a publica no tópico SNS. Como existem algumas limitações, como o número de caracteres máximo e restrições no uso de caracteres especiais, o texto elaborado precisou ser conciso e direto, informando o nome do usuário que enviou o SOS e as coordenadas de sua localização. Um exemplo de mensagem SOS pode ser visualizado na Figura 22.

Figura 20 – Mensagem *uplink* recebida no tópico MQTT

Subscriptions	lorawan/+/uplink
lorawan/+/uplink ♥ ✕	<div style="border: 1px solid #ccc; padding: 5px;"> <p>▼ lorawan/78218487F7F0FEFF/uplink</p> <pre> { "end_device_ids": { "device_id": "eui-78218487f7f0feff", "application_ids": { "application_id": "blipsos" } }, "dev_eui": "78218487F7F0FEFF", "join_eui": "0000000000000000", "dev_addr": "260D0420" }, "correlation_ids": ["as:up:01GK3DTCVZQ99NF2782AMWS1PJ", "gs:conn:01GJZ15J6G4XMMZKX7TQ5FTKER", "gs:up:host:01GJZ15JEWXTMEJY1N2AF68DA6", "gs:uplink:01GK3DTCNEYDHFJ2HH56WH3754", "ns:uplink:01GK3DTCNE9XZQFJV075FZ6N3A", "rpc:/ttn.lorawan.v3.GsNs/HandleUplink:01GK3DTCNEQ4Z9MBS9JJTXZAM1", "rpc:/ttn.lorawan.v3.NsAs/HandleUplink:01GK3DTCVYJQVY8RFT6QXHQRVH"], "received_at": "2022-11-30T04:48:16.254810584Z", "uplink_message": { "session_key_id": "AYTG3JoAW669xI+XtR/hxg==", "f_port": 10, "f_cnt": 2575, "frm_payload": "f///f///AADnAA==", "decoded_payload": { "bytes": { "altitude": 0, "hdop": 23.1, "latitude": -0.00000536441834242396, "longitude": -0.00001072883668484792, "sats": 0 } } }, "rx_metadata": [{ </pre> </div>

Fonte: Autoria própria (2022).

Na Listagem 3 pode ser visto o *schema* (código que descreve todos os dados disponíveis a serem consumidos e sua estrutura) escrito para a API GraphQL, consumida pelo aplicativo móvel e configurada no AppSync (item 11 do diagrama da Figura 18). Esse schema define os tipos User e SOS e as consultas encarregadas de possibilitar a leitura e escrita do banco de dados através de *resolvers*. *Resolvers* são a maneira de definir uma consulta ao DynamoDB com parâmetros personalizados recebidos pela consulta à API GraphQL. Dessa forma, é possível realizar leituras e escritas de dados no banco.

Listagem 2 – Rule sendSMS e sendWhatsapp

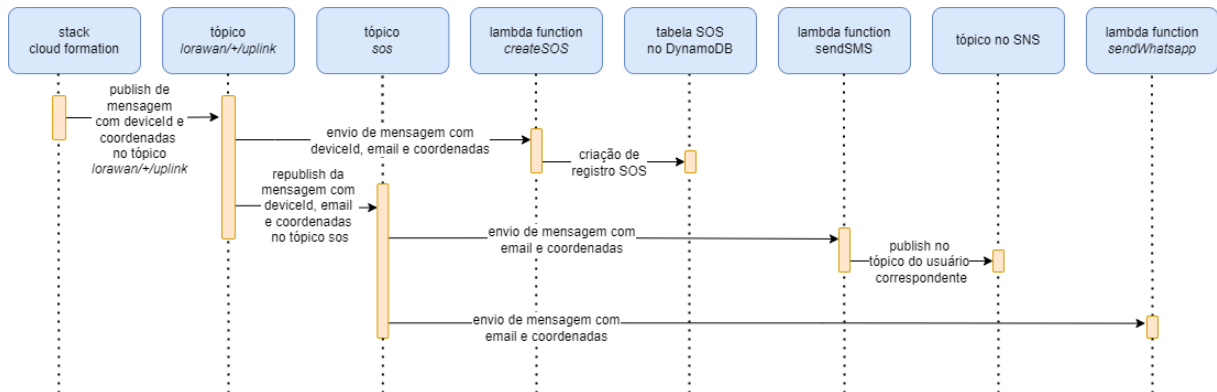
```

1 SELECT
2 deviceId AS deviceId ,
3 lat AS lat ,
4 lon AS lon ,
5 get_dynamodb("UserTable", "email", email ,
6     "arn:aws:iam::{id}:role/IoTtest") AS user
7 FROM "sos"

```

Fonte: Autoria própria (2022).

Figura 21 – Diagrama de sequência do envio de um SOS



Fonte: Autoria própria (2022).

5.5 Projeto WhatsApp Business

Para utilizar a API do WhatsApp Business para o envio de mensagens a números de telefone reais, foi preciso criar um projeto na plataforma Meta for Developers. Esse projeto permite a criação de um número de teste para realizar o envio das mensagens, e concede o cadastro de até 5 números de telefone para serem destinatários das mensagens de forma gratuita. Como o projeto é no modo de desenvolvimento, o cadastro dos destinatários exige uma confirmação por parte do proprietário do número, de modo a evitar *spam*. Além disso, o *token* de autenticação para as chamadas de API é temporário, sendo necessário atualizá-lo a cada 23h.

Para evoluir a aplicação do modo de desenvolvimento para o modo *live*, é necessário solicitar a análise do aplicativo, o que significa passar por um processo de verificação e testes da aplicação por parte de uma equipe da Meta, o que envolve uma certa burocracia de criação de termos de uso e políticas de privacidade e pode levar um certo período de tempo para passar pela avaliação. Como foi entendido que não havia a necessidade de passar por esse processo para cumprir com os requisitos do projeto, foi mantida a utilização do modo de desenvolvimento.

Para a configuração da mensagem a ser enviada, foi criado um modelo de mensagem personalizado, que pode ser visto na Figura 23. Dessa forma, a mensagem sempre irá conter o nome do usuário, o que é importante para o conhecimento dos integrantes da sua lista de contatos. Como as opções de personalização de um modelo de mensagens são limitadas, não

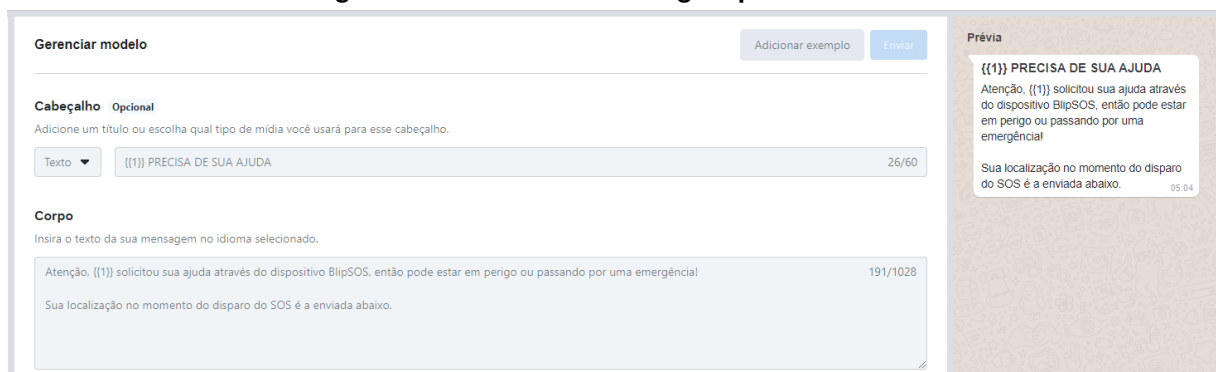
Figura 22 – Exemplo de mensagem SMS recebida



Fonte: Autoria própria (2022).

foi possível incluir a funcionalidade de localização no modelo. Portanto, a solução encontrada foi fazer duas requisições diferentes para a API do WhatsApp: uma passando o parâmetro do nome do usuário para o envio do modelo de mensagem personalizado, e outra passando os parâmetros de latitude e longitude para o envio do mapa identificando o ponto referente a essas coordenadas. Um exemplo de mensagens recebidas pelos telefones da lista de contatos pode ser visto na Figura 24.

Figura 23 – Modelo de mensagem personalizado



Fonte: Autoria própria (2022).

5.6 Aplicativo Móvel

O aplicativo, desenvolvido e testado em um dispositivo Android Samsung A72, resultou em uma aplicação com interfaces intuitivas de fácil utilização, cujas telas serão exibidas e explicadas a seguir.

Listagem 3 – Schema GraphQL

```

1 type User {
2   name: String!
3   email: String!
4   phone: String!
5   password: String!
6   deviceId: String
7   contacts: [Contact]
8   createdAt: String
9   updatedAt: String
10 }
11 type Contact {
12   name: String!
13   phone: String!
14 }
15 type SOS {
16   id: ID!
17   deviceId: String!
18   email: String!
19   location: String!
20   datetime: String!
21 }
22 type Query {
23   getUser(email: String!): User
24   getSosHistory(email: String!): [SOS]
25 }
26 type Mutation {
27   updateUser(email: String!, name: String!,
28   phone: String!, password: String!): User
29   updateUserContacts(email: String!, contacts: [ContactInput]): User
30 }

```

Fonte: Autoria própria (2022).

Na Figura 25 é exibida a tela de *login*, com os campos de email e senha. Como não era necessário para o cumprimento dos objetivos propostos, o aplicativo não implementa uma funcionalidade de *login* robusta, utilizando recursos próprios como AWS Cognito, por exemplo. Em vez disso, é feita uma simples consulta à API para obter o usuário correspondente ao email digitado.

A Figura 26 apresenta a tela inicial, que inclui instruções de uso de cada tela do aplicativo.

Na Figura 27 é apresentada a tela de dados do usuário, onde este pode editar seu nome, senha, telefone e id do dispositivo. Este último é realizado através de uma modal, dando destaque à importância dessa informação e incluindo instruções de como adquiri-la (caso fosse um produto real), como pode ser visto na Figura 28.

Figura 24 – Exemplo de mensagens WhatsApp recebidas



Fonte: Autoria própria (2022).

Em seguida, a Figura 29 exibe a tela da lista de contatos de emergência, onde o usuário pode adicionar, excluir e editar as pessoas e seus respectivos números de telefone para as quais o usuário deseja enviar suas mensagens de SOS.

Por fim, a tela de histórico de envios de SOS mostrada na Figura 30 contém uma lista com todas as mensagens de SOS enviadas por este usuário, contendo informações da data e horário e um pequeno mapa interativo indicando o local em que o disparo foi realizado.

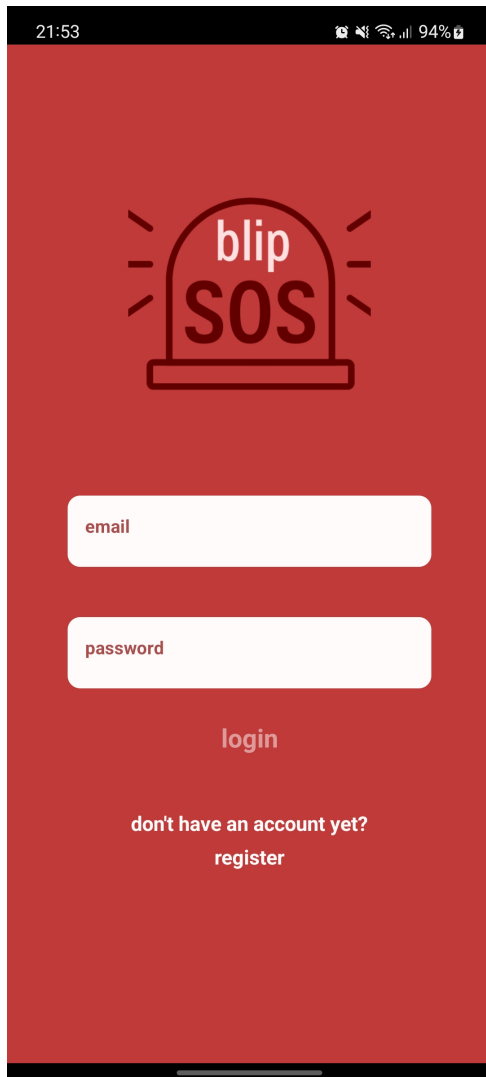
5.7 Testes

Para validar o funcionamento e os resultados obtidos com o sistema final, foram realizados diferentes testes, a serem discutidos a seguir.

5.7.1 Alcance

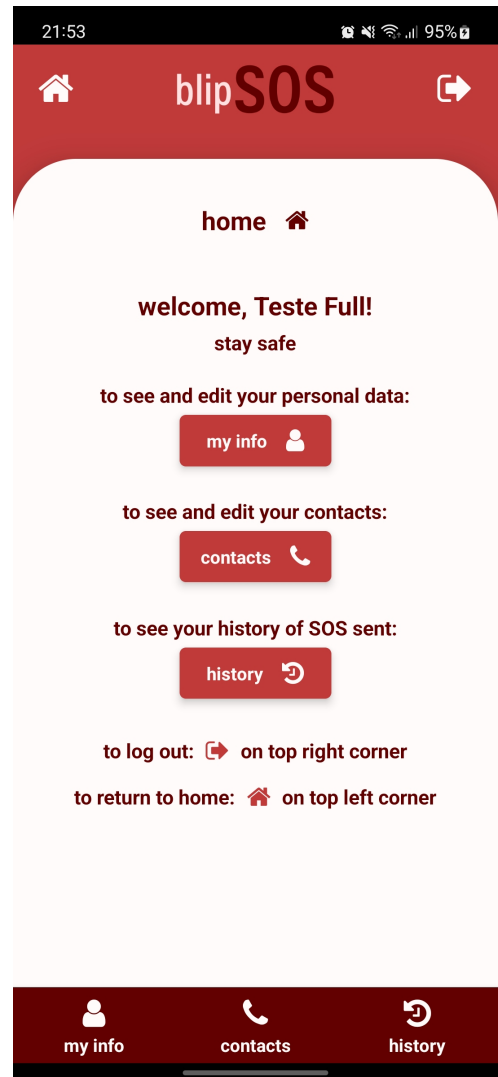
Para verificar a capacidade de comunicação do dispositivo com a rede LoRaWAN em termos de alcance, foram realizadas diversas medições em diferentes locais ao redor da localização do *gateway*, fornecido pelo LabSC e equipado com uma antena com 6dBi de ganho. Os dados registrados durante as medições incluíram: a validação de *joins* bem sucedidos com o *server* TTN e o envio de mensagens *uplink*, o local em que o dispositivo se encontrava, e o tempo gasto entre o apertado do botão e o recebimento da notificação WhatsApp/SMS no celular dos contatos cadastrados (no caso de sucesso). Os resultados foram compilados em um mapa, personalizado através do site My Maps (Google Maps, 2022), que pode ser visto na Figura 31. O mapa marca todos os locais em que foram realizados testes e utiliza cores para ilustrar o nível de sucesso conforme a legenda na Tabela 1:

Figura 25 – Tela de login



Fonte: Autoria própria (2022).

Figura 26 – Tela inicial



Fonte: Autoria própria (2022).

Analisando este mapa, nota-se que a capacidade do alcance de captação do *gateway* é relativamente distribuído ao seu redor. Apesar disso, houve certos locais que não obtiveram sucesso mesmo estando mais próximos que outros locais de sucesso. Isso foi atribuído à possibilidade de alguns meios materiais estarem obstruindo a transmissão, como árvores e construções. Além disso, um fator identificado que pareceu influenciar consideravelmente os resultados foi a altitude. Percebeu-se que regiões de teste abaixo do nível do *gateway* apresentaram baixo alcance em relação à distância direta. Por outro lado, regiões com altitude acima do nível do *gateway* apresentaram alcance expressivamente maior. Considerando todos esses pontos de teste, foi registrado que a maior distância com a qual se obteve sucesso no envio das mensagens SOS foi 862 metros.

Figura 27 – Tela de dados do usuário

21:54 95%

blipSOS

my info

email: email@gmail.com

name: Teste Full

phone: 41111111111

password:

device id: eui-78218487f7f0feff

save

my info contacts history

Fonte: Autoria própria (2022).

Figura 28 – Tela de dados do usuário com modal do id do dispositivo

21:54 95%

blipSOS

my info

email: email@gmail.com

name: Teste Full

edit device id

type in here the id number from your device's box:

device id

ok

save

my info contacts history

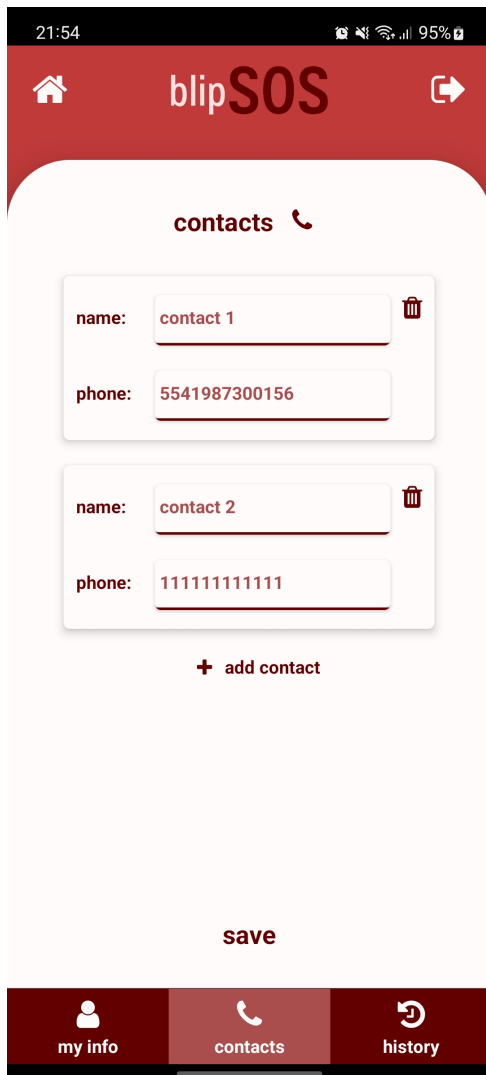
Fonte: Autoria própria (2022).

Calculando-se as taxas de erro e acerto, encontra-se que dos 43 testes realizados, 57% obtiveram sucesso, enquanto 30% falharam provavelmente devido à distância ou obstrução na transmissão.

É importante ressaltar que esse resultado obtido deu-se com a utilização de apenas um *gateway*. Caso a cobertura LoRaWAN fosse melhorada com a inserção de mais *gateway*, certamente seria obtido mais sucesso no envio de mensagens, mas essa validação está fora do escopo do projeto.

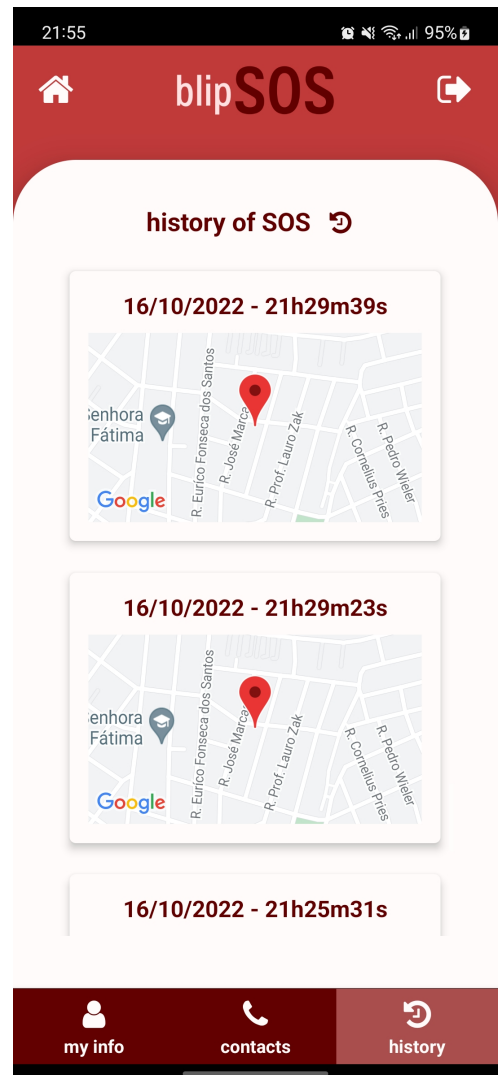
Em relação ao tempo gasto entre o apertado do botão no dispositivo e o recebimento das mensagens, todos os pontos de teste obtiveram um valor próximo, se mantendo em uma média de 6 segundos. O fato de o tempo resultante ser semelhante em todos os pontos independente

Figura 29 – Tela da lista de contatos



Fonte: Autoria própria (2022).

Figura 30 – Tela do histórico de envios de SOS



Fonte: Autoria própria (2022).

da distância é um bom sinal, pois entende-se que a comunicação da rede LoRaWAN é bem estabelecida.

5.7.2 Sinal do GPS

Os resultados obtidos com a utilização GPS NEO-6M, incluído no hardware TTGO T-Beam, apresentaram considerável limitação de locais que capacitam a obtenção dos dados, além de lentidão para sincronização do sinal. Infelizmente, somente foi obtido sucesso com a captação de dados de GPS em áreas sob céu aberto, excluindo espaços cobertos, o que pode ser considerada uma limitação relevante levando em conta os objetivos do projeto.

Além disso, notou-se que após a inicialização do módulo GPS, é necessário um tempo considerável para a obtenção dos dados de localização. Foram realizados testes em diferentes

É natural constatar que o tempo gasto esperando a sincronização do sinal GPS para localização é exageradamente alto, o que destaca o meio escolhido como interceptação da geolocalização do dispositivo, um item bastante relevante no projeto, como um ponto de atenção para trabalhos futuros.

5.8 Custos

Levando em conta que este projeto buscou ao máximo utilizar recursos *open source* e gratuitos, o gasto total resultante foi considerado baixo. A única compra realizada foi a da placa TTGO T-Beam, custando na época R\$163,55.

Quanto à AWS, contudo, apesar de ter sido mencionado que todos os recursos utilizados estariam dentro do nível *free tier* disponibilizado, houve imprevistos em relação ao envio de SMS. Devido aos destinatários das mensagens SMS serem de fora dos Estados Unidos, o envio envolve tramitação com companhias de telecomunicação brasileiras, o que acarreta em custos. A configuração padrão do SNS estabelece, por questões de segurança, um limite de U\$1.00 mensal para gastos com envio de SMS, resultando na liberação do envio de algumas dezenas de mensagens somente. Considerando o período de 4 meses em que esta funcionalidade foi implementada e testada, foi gasto um total de U\$4.00 (mais taxas), convertidos em R\$24,21.

A compilação de todos os custos pode ser conferida na Tabela 2.

Tabela 2 – Custos totais

Despesa	Valor
TTGO T-Beam	R\$163,55
AWS SNS	R\$24,21
Total	R\$187,76

Fonte: Autoria própria (2022).

5.9 Melhorias e evoluções

Esta seção foi incluída para discorrer brevemente sobre alguns pontos de melhoria que foram identificados, além de possíveis evoluções e incrementos dos quais tem-se ciência de que poderiam acrescentar bastante ao projeto em caso de trabalhos futuros.

O principal ponto de melhoria destacado é em relação à geolocalização. Os resultados obtidos com o módulo GPS NEO-6M e a antena acoplada à placa foram abaixo do esperado, especialmente pela necessidade de se estar a céu aberto para possibilitar a inicialização do módulo e a captação de sinal, processo este que demandou consideráveis minutos. Pensando na aplicação real do projeto, essas condições limitariam muito sua utilização, portanto seria necessário avaliar melhores opções, desde considerar testar diferentes modelos de antena até

estudar a possibilidade de outros métodos para obter geolocalização, como recursos da própria tecnologia LoRa.

Em relação às evoluções pretendidas, encontram-se pontos de aperfeiçoamento tanto para o projeto apresentado quanto para a ideia de aplicação do produto, todos listados a seguir:

- Implementar funcionalidade robusta de login e telas de cadastro de novo usuário no aplicativo;
- Evoluir recursos de interface do aplicativo, como *loadings* e mensagens de resposta para interações do usuário;
- Automatizar a criação de tópicos SNS com o cadastro de um novo usuário e atualização de telefones inscritos com a edição de contatos;
- Incluir tratamento de mensagens de downlink no firmware para possibilitar a indicação de resposta ao envio de SOS;
- Implementar um *stream* de transmissão de mensagens com a geolocalização do dispositivo para permitir rastreamento em tempo real;
- Adaptar o *hardware* para funcionar como um transceptor da rede LoRaWAN, de modo a utilizar os dispositivos como nós de propagação do sinal e consequentemente aumentando o raio de alcance da rede;
- Incluir um motor de vibração no *hardware* de modo a dar um *feedback* do reconhecimento do duplo clique para o usuário;
- Incluir a funcionalidade de conectar o dispositivo no celular via bluetooth para utilizá-lo como segunda opção de meio de envio de SOS;

6 CONCLUSÃO

Segurança pessoal é um tema que tem cada vez mais relevância e é de interesse geral da grande maioria das pessoas. A união desse tema com tecnologias modernas que evoluem a cada dia mais, como a Internet das Coisas (IoT), proporciona a criação de uma infinidade de aplicações. Dispositivos de monitoramento, *smart homes*, *wearables*, a lista de produtos que se utilizam de meios comunicações sem fio e integrações com serviços de computação em nuvem só cresce.

Incorporando-se a essa lista, este projeto apresentou uma proposta de desenvolvimento da idealização de um produto que busca oferecer maior sensação de segurança para seus usuários. Configurando algum tipo de acessório pequeno, discreto e passível de uso diário, esse produto possibilita uma forma prática de seu usuário pedir socorro caso se encontre em uma situação de perigo, garantindo que esse pedido de ajuda chegue a uma lista de contatos de emergência de forma que possam tomar alguma ação.

Os meios e tecnologias escolhidas para construir o protótipo desse produto visaram apresentar diferenciais quando comparado a outros semelhantes já existentes. A principal vantagem que esse projeto apresenta é o meio de comunicação utilizado, implementando a transmissão de dados de geolocalização via LoRaWAN, uma rede LPWAN, de forma a possibilitar o envio de mensagens através de um meio que não fosse dependente de um telefone celular ou da rede de telefonia. A possibilidade de integração dessa rede com um servidor AWS na nuvem viabilizou a construção de um serviço escalável e capaz de orquestrar todas as comunicações, gerenciamento de dados e envio de mensagens. A escolha dos dois provedores de envio de mensagens, WhatsApp e SMS, garante um bom nível de confiança de que as pessoas selecionadas receberão a notificação de SOS como esperado, pois o primeiro é um aplicativo de mensagens utilizado por quase todos os brasileiros, e o segundo possibilita a recepção da mensagem sem a necessidade de conexão com a internet.

O sistema final obtido e os resultados atingidos atenderam aos objetivos propostos de forma satisfatória. Foi possível realizar o envio de mensagens SOS em diferentes locais, verificando que todo o fluxo de comunicação e integração é funcional e bem desenvolvido. As mensagens de SOS recebidas correspondem ao planejado, principalmente as mensagens WhatsApp, por terem o benefício de exibir um mapa indicando a localização das coordenadas do local de envio do SOS. Além disso, o aplicativo mobile desenvolvido satisfaz a necessidade de uma interface conveniente que possibilitasse a edição dos dados por parte do usuário e a visualização do histórico de envios de SOS.

O único ponto de atenção que acabou não atingindo um nível aceitável de satisfação foi a captação da geolocalização através do módulo GPS. Infelizmente, o longo tempo necessário para sua inicialização e captação dos dados de localização deixaram a desejar, fazendo dessa questão o item prioritário para melhoria na continuidade do trabalho.

Em resumo, os resultados obtidos com este trabalho de conclusão de curso foram considerados satisfatórios e os objetivos definidos foram atingidos. Apesar disso, foi possível listar inúmeras oportunidades de melhorias e evoluções para o projeto, que poderiam agregar muito ao resultado final.

REFERÊNCIAS

- ADT. **ADT**. 2022. Disponível em: <https://www.adt.com.br/>. Acesso em: 23 nov. 2022.
- AliExpress. **Kospet prime 4g relógio inteligente**. 2022. Disponível em: <https://pt.aliexpress.com/item/4000419552908.html>. Acesso em: 23 nov. 2022.
- Amazon Web Services. **AWS Documentation**. 2022. Disponível em: <https://docs.aws.amazon.com/>. Acesso em: 14 ago. 2022.
- Amazon Web Services. **Documentação Amplify**. 2022. Disponível em: <https://docs.amplify.aws/>. Acesso em: 7 out. 2022.
- Amazon Web Services. **Mensagens de texto em dispositivos móveis (SMS)**. 2022. Disponível em: https://docs.aws.amazon.com/pt_br/sns/latest/dg/sns-mobile-phone-number-as-subscriber.html. Acesso em: 11 set. 2022.
- Amazon Web Services. **O que é computação em nuvem**. 2022. Disponível em: <https://aws.amazon.com/pt/what-is-cloud-computing/>. Acesso em: 16 dez. 2022.
- Amazon Web Services. **Perguntas frequentes sobre o nível de uso gratuito da AWS**. 2022. Disponível em: <https://aws.amazon.com/pt/free/faqs/>. Acesso em: 14 ago. 2022.
- American Tower. **Rede ATC LoRaWAN**. 2021. Disponível em: <https://americantower.com.br/pt/solu%C3%A7%C3%B5es/rede-neutra-loRaWAN.html>. Acesso em: 5 ago. 2022.
- Brasscom. **Segurança da Informação e Segurança Cibernética**. [S.l.], 2022. Disponível em: <https://brasscom.org.br/pdfs/seguranca-da-informacao-e-seguranca-cibernetica/>. Acesso em: 29 jul. 2022.
- CASTRO, B. Em cinco anos, mais de 700 mulheres vítimas de violência são protegidas por meio de monitoramento eletrônico. **G1 Globo**, 2019. Disponível em: <https://bityli.com/zhsAJkgAk>. Acesso em: 21 nov. 2022.
- Claro. **Kids On**. 2022. Disponível em: <https://www.claro.com.br/servicos/kidson>. Acesso em: 23 nov. 2022.
- MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES (MCTIC). **Internet das Coisas: Um Plano de Ação para o Brasil**. Disponível em: <https://www.gs1br.org/conteudo/materiais-tecnicos/Industria40/IoT%20Plano%20de%20Acao%20para%20o%20Brasil.pdf>. Acesso em: 17 ago. 2022.
- COULOURIS, G. *et al.* **Sistemas Distribuídos: conceitos e projeto**. Bookman Editora, 2013. ISBN 9788582600542. Disponível em: <https://books.google.com.br/books?id=6WU3AgAAQBAJ>.
- CROQUER, G. Sequestros-relâmpago crescem 35% em SP com chegada do Pix. **R7**, 2022. Disponível em: <https://www.brasilparalelo.com.br/artigos/indices-criminalidade-brasil>. Acesso em: 21 nov. 2022.
- eriktheV-king. **TTGO T-Beam GPS reset**. 2020. Disponível em: https://github.com/eriktheV-king/TTGO_T-beam_GPS-reset. Acesso em: 16 ago. 2022.
- Figma. **Documentação Figma**. 2022. Disponível em: <https://www.figma.com/>. Acesso em: 29 set. 2022.

GALVÃO, I. P. 81% das mulheres já sofreram violência em seus deslocamentos pela cidade. 2021. Disponível em: <https://dossies.agenciapatriciagalvao.org.br/violencia-em-dados/81-das-mulheres-ja-sofreram-violencia-em-seus-deslocamentos-pela-cidade/>. Acesso em: 21 nov. 2022.

Google Maps. **My Maps**. 2022. Disponível em: <https://www.google.com/intl/pt-BR/maps/about/mymaps/>. Acesso em: 20 nov. 2022.

GOV.BR. Brasil tem mais de 31 mil denúncias de violência doméstica ou familiar contra as mulheres até julho de 202. **Gov.br**, 2022. Disponível em: <https://bityli.com/VRWbhOI>. Acesso em: 21 nov. 2022.

International Telecommunication Union (ITU). **LoRaWAN® Formally Recognized as ITU International Standard for Low Power Wide Area Networking**. [S.l.], 2021. Disponível em: <https://lora-alliance.org/lora-alliance-press-release/lorawan-formally-recognized-as-itu-international-standard-for-low-power-wide-area-networking/>. Acesso em: 21 out. 2022.

invisaWear. **invisaWear**. 2022. Disponível em: <https://www.invisawear.com/>. Acesso em: 8 jun. 2022.

ISG. **ISG Provider Lens Internet of Things — Services and Solutions 2022**. [S.l.], 2022. Disponível em: <https://www.tgt.com.br/blog/2022/08/10/isg-provider-lens-internet-of-things-services-and-solutions-brazil-2022/>.

kizniche. **TTGO T-Beam Tracker for The Things Network**. 2022. Disponível em: <https://github.com/kizniche/ttgo-tbeam-ttn-tracker>. Acesso em: 25 jul. 2022.

LilyGO. **LILYGO® TTGO T-Beam V1.1**. 2020. Disponível em: http://www.lilygo.cn/prod_view.aspx?TypeId=50060&Id=1317. Acesso em: 16 jun. 2022.

LilyGO. **Pinagem TTGO-T-Beam v1.1**. 2022. Disponível em: <https://github.com/LilyGO/TTGO-T-Beam>. Acesso em: 7 jul. 2022.

LoRa Alliance. **What is LoRaWAN® Specification**. 2022. Disponível em: <https://lora-alliance.org/about-lorawan/>. Acesso em: 15 jul. 2022.

mcci-catena. **Biblioteca Arduino LMIC**. 2022. Disponível em: <https://github.com/mcci-catena/arduino-lmic>. Acesso em: 15 jul. 2022.

Meta. **API de Nuvem da Plataforma do WhatsApp Business**. 2022. Disponível em: <https://developers.facebook.com/docs/whatsapp/cloud-api>. Acesso em: 18 set. 2022.

Meta. **Documentação React Native**. 2022. Disponível em: <https://reactnative.dev/>. Acesso em: 2 out. 2022.

Meta. **Meta for Developers**. 2022. Disponível em: <https://developers.facebook.com/>. Acesso em: 16 dez. 2022.

mikalhart. **Biblioteca TinyGPSPlus**. 2022. Disponível em: <https://github.com/mikalhart/TinyGPSPlus>. Acesso em: 15 jul. 2022.

MQTT. **MQTT**. 2022. Disponível em: <https://mqtt.org/>. Acesso em: 4 nov. 2022.

Oracle. **O que é IoT**. 2022. Disponível em: <https://www.oracle.com/br/internet-of-things/what-is-iot/>. Acesso em: 16 dez. 2022.

PARALELO, B. Índices chocantes de Criminalidade no Brasil. **Brasil Paralelo**, 2022. Disponível em: <https://www.brasilparalelo.com.br/artigos/indices-criminalidade-brasil>. Acesso em: 21 nov. 2022.

PURZ, M. WhatsApp no Brasil: números atuais e as oportunidades comerciais do app. **MessengerPeople**, 2022. Disponível em: <https://www.messengerpeople.com/pt-br/whatsapp-no-brasil/>. Acesso em: 29 nov. 2022.

RAZA, U.; KULKARNI, P.; SOORIYABANDARA, M. Low Power Wide Area Networks: An Overview. **IEEE Communications Surveys Tutorials**, 2017.

REYNDERS, B.; POLLIN, S. Chirp spread spectrum as a modulation technique for long range communication. *In: 2016 Symposium on Communications and Vehicular Technologies (SCVT)*. [S.l.: s.n.], 2016.

RIOT. **Documentation: Support for TTGO T-Beam boards**. [S.l.], 2022. Disponível em: https://doc.riot-os.org/group__boards__esp32__ttgo-t-beam.html. Acesso em: 13 jul. 2022.

Semtech. **Documentation LoRa and LoRaWAN**. 2022. Disponível em: <https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan>. Acesso em: 12 jul. 2022.

Semtech. **Semtech**. 2022. Disponível em: <https://www.semtech.com/>. Acesso em: 12 jul. 2022.

TechTudo. **Saiba o que é GPS e como funciona**. 2011. Disponível em: <https://www.techtudo.com.br/noticias/2011/12/como-funciona-o-gps.ghtml>. Acesso em: 29 nov. 2022.

The GraphQL Foundation. **Documentação GraphQL**. 2022. Disponível em: <https://graphql.org/>. Acesso em: 7 out. 2022.

The New Stack. **Entenda o MQTT**. 2022. Disponível em: <https://thenewstack.io/mqtt-protocol-iot/>. Acesso em: 17 nov. 2022.

The Things Network. **The Things Network**. 2022. Disponível em: <https://www.thethingsnetwork.org/country/brazil/>. Acesso em: 11 jun. 2022.

The Things Network. **End Device Activation**. 2022. Disponível em: <https://www.thethingsnetwork.org/docs/lorawan/end-device-activation/>. Acesso em: 20 nov. 2022.

The Things Network. **Frequency Plans by Country**. 2022. Disponível em: <https://www.thethingsnetwork.org/docs/lorawan/frequencies-by-country/>. Acesso em: 5 nov. 2022.

The Things Stack. **The Things Stack Console**. 2022. Disponível em: <https://au1.cloud.thethings.network/console/>. Acesso em: 6 ago. 2022.

VASCONCELLOS, H. Relatos de dopagem em Uber e 99 viralizam nas redes. **Tilt UOL**, 2022. Disponível em: <https://www.uol.com.br/tilt/noticias/redacao/2022/06/06/como-se-proteger-do-golpe-da-dopagem-por-gas-ou-spray.htm>. Acesso em: 21 nov. 2022.

Wikipedia Commons. **MQTT protocol example without QoS**. 2018. Disponível em: https://commons.wikimedia.org/wiki/File:MQTT_protocol_example_without_QoS.svg. Acesso em: 17 nov. 2022.