



**UNIVERSIDADE
FEDERAL RURAL
DE PERNAMBUCO**



Beatriz Carvalho Falcão

Desenvolvimento de um aplicativo móvel para suporte ao monitoramento de ninhos de tartaruga marinha

Recife

Março de 2024

Beatriz Carvalho Falcão

Desenvolvimento de um aplicativo móvel para suporte ao monitoramento de ninhos de tartaruga marinha

Artigo apresentado ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Universidade Federal Rural de Pernambuco – UFRPE
Departamento de Estatística e Informática
Curso de Bacharelado em Sistemas de Informação

Orientador: Victor Wanderley Costa de Medeiros

Recife
Março de 2024

Beatriz Carvalho Falcão

Desenvolvimento de um aplicativo móvel para suporte ao monitoramento de ninhos de tartaruga marinha

Artigo apresentado ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Aprovada em: 08 de Março de 2024.

BANCA EXAMINADORA

Victor Wanderley Costa de Medeiros (Orientador)

Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

Cícero Garrozi

Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

Desenvolvimento de um aplicativo móvel para suporte ao monitoramento de ninhos de tartaruga marinha

Beatriz Carvalho Falcão¹, Victor Wanderley Costa de Medeiros¹

¹Departamento de Estatística e Informática – Universidade Federal Rural de Pernambuco
Rua Dom Manuel de Medeiros, s/n, - CEP: 52171-900 – Recife – PE – Brasil

beatriz.cfalcao@ufrpe.br, victor.wanderley@ufrpe.br

Resumo. *É notável que a degradação da vida marinha tem sido um tópico fortemente discutido nas últimas décadas, especialmente no que diz respeito às tartarugas marinhas. As ações humanas têm causado sérios danos a essas espécies em seu habitat natural. No entanto, é possível mitigar esses impactos negativos por meio do uso de tecnologias digitais, no âmbito de desenvolvimento de soluções que visem preservar essas espécies vulneráveis. Este projeto tem como objetivo a implementação de uma solução eficaz de suporte ao monitoramento de ninhos tartarugas marinhas, utilizando a aplicação móvel do Thingsboard, visando facilitar a atividade de monitoramento e a precisão dos processos envolvidos.*

Abstract. *It is notable that the degradation of marine life has been a heavily discussed topic in recent decades, especially in regard to sea turtles. Human actions have caused serious damage to these species in their natural habitat. However, it is possible to mitigate these negative impacts through the use of digital technologies, in the context of developing solutions that aim to preserve these vulnerable species. This project targets the implement of an effective solution that provides support in the monitoring of sea turtle nests, using the Thingsboard mobile application, aiming to facilitate the monitoring activity and the accuracy of the processes involved.*

1. Introdução

Nas últimas décadas se tornou fortemente evidente a degradação ambiental através das ações humanas. Apesar do tema sobre degradação ambiental remeter primeiramente a destruição das florestas, os oceanos também são vítimas dessa calamidade.

A vida marinha tem sido ameaçada constantemente por poluentes descartados in-consequentemente nos oceanos [Pereira 2023]. O oceano é o habitat das tartarugas marinhas, que desempenham um papel crucial de equilíbrio em seu ecossistema, além de desempenharem um papel social e econômico importante dentro do Brasil, através do turismo e das atividades de conservação [Sarmiento 2013].

Na atualidade, é um fato que a tecnologia possui uma grande influência em diversos âmbitos dentro da sociedade. Um dos papéis mais importantes da tecnologia é servir como ferramenta nos processos de solução de problemas, gerando impacto direto na forma em que as coisas são feitas, [Benakouche 1999]. Além disso, a tecnologia é conhecida por ser uma ferramenta flexível permitindo com que ela possa ser empregada em diversos cenários. Essa característica é muito relevante a qualquer desenvolvimento

de projeto, tendo em vista que soluções distintas com as mais específicas funcionalidades poderão ser desenvolvidas.

Com base no contexto acima, é nítido evidenciar que há uma necessidade em desenvolver soluções que auxiliem a amenizar o impacto das ações humanas no meio ambiente. A conservação parte não apenas do cuidado com o habitat das tartarugas marinhas, mas também com o seu ciclo natural de vida. Este projeto concentra-se no ciclo reprodutivo, em especial a desova e a proteção dos ninhos de tartaruga marinha. Mais especificamente, este projeto consiste na implementação de um aplicativo móvel de suporte a instalação e manutenção de sensores utilizados no monitoramento da atividade de ninhos de tartarugas marinhas com a finalidade de apoiar a atividade de preservação.

1.1. Contexto

O monitoramento das áreas de desova e eclosão dos ovos de tartaruga marinha é crucial para garantir a preservação dessas espécies. Através desse acompanhamento, é possível coletar dados essenciais sobre o número de ninhos, o sucesso reprodutivo, as ameaças enfrentadas pelas tartarugas e seus filhotes, entre outros aspectos. Essas informações são fundamentais para orientar a implementação de medidas de conservação e manejo adequadas, visando proteger as populações de tartarugas marinhas e garantir a continuidade de seus ciclos de vida. Além disso, o monitoramento contribui para aumentar a conscientização da comunidade local e de visitantes sobre a importância da conservação dessas espécies e dos ambientes costeiros onde elas se reproduzem.

A tecnologia digital desempenha um papel crucial no apoio a este monitoramento, especialmente considerando que estas atividades são predominantemente realizadas manualmente. Dispositivos sensores que permitam uma coleta mais eficiente e precisa de informações sobre a atividade dos ninhos tartarugas proporcionam uma visão mais abrangente e detalhada do processo reprodutivo das tartarugas marinhas auxiliando no desenvolvimento de estratégias de conservação mais eficazes. Ao integrar a tecnologia digital ao monitoramento tradicional, é possível otimizar os esforços de conservação e garantir a proteção dessas espécies vulneráveis e dos ecossistemas costeiros onde habitam.

1.2. Motivação

O presente estudo é motivado pela necessidade de melhorar a eficiência do monitoramento de ninhos de tartarugas marinhas e parte do pressuposto de que o emprego de suporte tecnológico pode colaborar neste sentido aumentando, inclusive, a capacidade de monitoramento.

1.3. Objetivos

Esse artigo tem como objetivo principal o desenvolvimento de uma aplicação móvel de apoio a otimização do monitoramento de ninhos de tartarugas marinhas. A abordagem do trabalho visa reduzir significativamente a necessidade de deslocamentos extensos diários por parte dos responsáveis pelo cuidado dos ninhos, eliminando a exigência de verificações repetitivas.

Referente aos objetivos específicos, o presente artigo busca:

1. Explorar o aplicativo móvel open-source do Thingsboard para manipulação de uma nova aplicação de suporte às ferramentas existentes da plataforma.

2. Permitir fácil acesso aos dados retornados pelos sensores através da aplicação móvel.
3. Desenvolver uma lógica de device claiming compatível com o cenário dos usuários.
4. Desenvolver uma função de retorno da localização do telefone móvel para atribuição de localização aos ninhos.
5. Criação de uma lógica de alarmes que envia notificações móveis.

2. Referencial teórico

Nesta seção serão abordados os principais conceitos técnicos com o propósito de apoiar a compreensão do presente artigo, que são: conservação das tartarugas marinhas e o papel da tecnologia; o que é internet das coisas (IoT); protocolo MQTT; LoRa; open source; dashboards; e *Widgets*.

2.1. Conservação das tartarugas marinhas e o papel da tecnologia

Existem cinco espécies de tartarugas marinhas que ocorrem no Brasil (Carreta carretta, *Chelonia mydas*, *Eretmochelys imbricata*, *Lepidochelys olivacea* e *Dermochelys coriacea*) e quase todas elas estão listadas com algum grau de risco de extinção [Martins and Molina 2008]. Apenas a Tartaruga-verde (*Chelonia mydas*) que segundo o ICMBio¹ (2018) foi classificada como quase ameaçada, saindo assim do grupo de categorias que expressam risco de extinção.

Dado esse contexto, é notável que a conservação das tartarugas marinhas é de suma importância no Brasil. São diversas as abordagens utilizadas para a conservação da espécie, sendo elas por exemplo: leis com finalidade de promoção a proteção da fauna brasileira, pesquisas e desenvolvimento de tecnologias especializadas.

Nessas circunstâncias a tecnologia pode assumir o papel de ser uma das ferramentas chaves na conservação à vida das tartarugas marinhas. Na atualidade diversas tecnologias já são utilizadas, como por exemplo satélites que fazem o rastreamento das tartarugas marinhas para estudos comportamentais [Timko and Kolz 1982].

2.2. Internet das Coisas (IoT)

Internet of Things (IoT) que em tradução literal significa Internet das Coisas é um conceito que vem sendo bastante pesquisado e discutido nas últimas décadas. Segundo [McEwen and Cassimally 2014] Internet das Coisas é quando a internet é embutida em objetos do dia-a-dia, sendo esses objetos considerados *things* de IoT. A partir do momento que o objeto tem acesso a internet, ele pode receber e enviar informações ou se comunicar com algo (outros objetos ou alguma rede), trazendo conectividade ao objeto. De forma simplificada, internet das coisas é quando tecnologia é embutida em algum objeto, como por exemplo um ventilador, e esse ventilador recebe a habilidade de saber qual a temperatura atual do cômodo, também coleta e envia para um servidor informações sobre quantas vezes ele é ligado, desligado, quais horários ele fica ligado e entre outros dados. A partir desses dados é possível criar um sistema inteligente que por exemplo liga o ventilador nos horários mais quentes e que pessoas estão na casa.

¹<https://www.icmbio.gov.br/centrotamar/ultimas-noticias/53-nota-de-esclarecimento-do-centro-tamar-icmbio-sobre-a-nova-classificacao-do-status-de-conservacao-da-tartaruga-verde-chelonia-mydas-no-brasil>

2.3. Protocolo MQTT

Message Queuing Telemetry Transport (MQTT), segundo [Hillar 2017] “É um protocolo *machine to machine* (M2M) e de conectividade baseada em Internet das Coisas. O esquema de troca de mensagens é fundamentado no modelo Publicador-Assinante (*Publisher-Subscriber*), extremamente simples e leve que roda em cima do *Transmission Control Protocol/ Internet Protocol* (TCP/IP)”, ou seja, MQTT é um protocolo de troca/envio de mensagens.

Neste projeto o protocolo MQTT cumpre o papel de envio das leituras feitas pelo dispositivo que contém os sensores. Essas leituras são os dados de temperatura, movimento e umidade, e através do MQTT elas são enviadas para o servidor Thingsboard Cloud.

2.4. LoRa

LoRa é a abreviação de Long Range e é uma técnica de modulação de espectro espalhado desenvolvida pela empresa SEMTECH². Segundo a SEMTECH “LoRa é uma plataforma sem fio de longo alcance e baixo consumo de energia” [Semtech 2024]. Através das informações supracitadas LoRa foi utilizada neste projeto como fonte de comunicação entre o *hardware* que contém os sensores e o servidor que recebe e armazena as telemetrias.

2.5. Open Source

Open Source possui diversas formas de denominação, podendo ser conhecido como software livre, software de código aberto ou fonte aberta em tradução direta. Em 1998, Christine Peterson inventou o termo *Open Source* com o propósito de sobrepor o modo como comumente era conhecido anteriormente, *Free Software*, pois quando as pessoas ouviam ou pensavam em software grátis elas pensavam em um software privado que não havia custos para adquirir, porém esse não é o conceito de *Open Source*.

Portanto, mesmo sendo citado vagamente no passado, em 1983 Richard Stallman criou a *Free Software Foundation* (FSF) e foi creditado pela criação do termo *Free Software* equivalente ao *Open Source* da atualidade. No site da FSF³, *Free Software* é descrito como “software gratuito em que os usuários têm a liberdade de executar, editar, contribuir e compartilhar o software. Assim, o *Free Software* é uma questão de liberdade, não de preço.” [free Software Foundation 2022]

Utilizando o conceito de *Open Source* mencionado anteriormente, foi elaborada uma aplicação que se baseia no código-fonte aberto do *ThingsBoard Professional Edition Mobile App*. Este código possui uma licença BSD de 3 cláusulas, que permite a modificação e redistribuição do código-fonte, desde que o aviso de direitos autorais seja mantido e que o nome dos autores e colaboradores do código não sejam usados para promoção de produtos derivados da aplicação original.

2.6. Dashboards

Os *dashboards* são interfaces que reúnem um conjunto de informações em um único lugar com o propósito de facilitar a visualização dessas informações.

²<https://www.semtech.com/lora/what-is-lora>

³<https://www.fsf.org/>

Dashboards são comumente utilizados por empresas para criar visões de indicadores e métricas que auxiliam na tomada de decisões dentro da organização. É importante ressaltar que a forma como os dados são organizados dentro do *dashboard* é um quesito crucial para que a visualização seja flúida e o *dashboard* cumpra o propósito de facilitação. A Figura 1 ilustra um dos dashboards desenvolvidos neste trabalho demonstrando o seu funcionamento porém, utilizando dados fictícios.

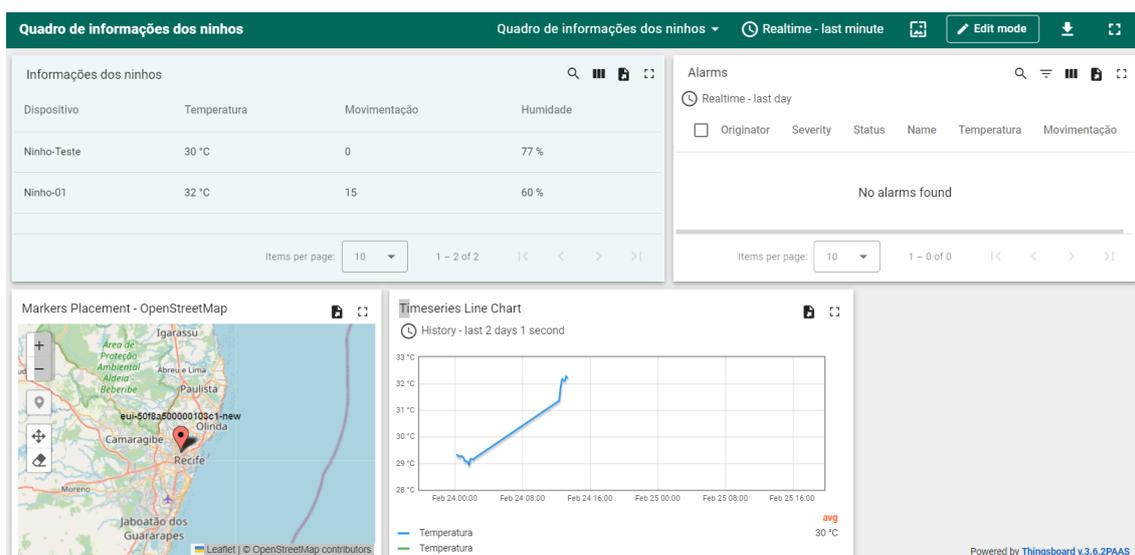


Figura 1. Dashboard desenvolvido para o projeto. Fonte: Autora.

A Figura 1 ilustra também a utilidade de um *dashboard*, através do agrupamento de informações sobre as telemetrias recebidas pelo dispositivo, alarmes, localização do dispositivo e um gráfico contendo a variância de temperatura em uma única tela.

2.7. Widgets

Widget vem da junção de duas palavras inglesas *window* e *gadget*. *Widgets* funcionam como janelas de aplicação que são responsíveis, ou seja, podem conter botões e outras funcionalidades além de poder conter um pequeno conjunto de informações (Figura 2). Geralmente os *widgets* possuem tamanho reduzido, justamente para atuarem como pequenas janelas funcionais.

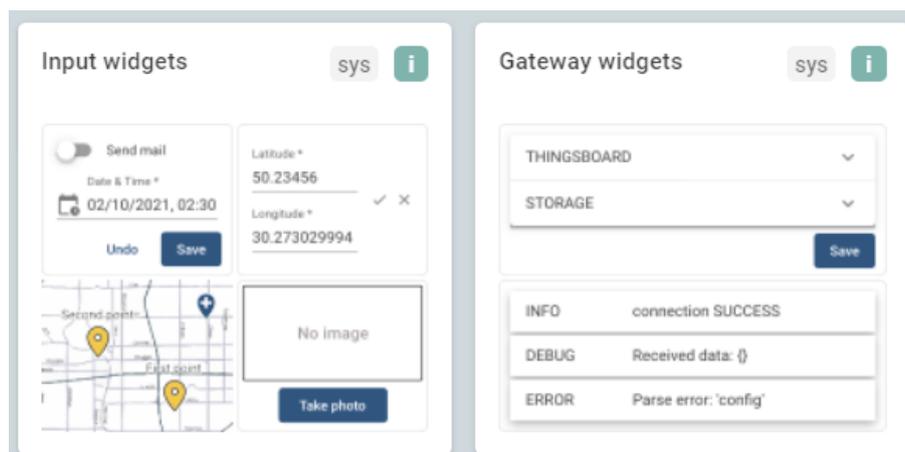


Figura 2. Exemplo de *widgets* do Thingsboard Cloud. Fonte: ThingsBoard Cloud.

O conceito de *widgets* é a base de qualquer desenvolvimento dentro da plataforma Thingsboard (TB) Cloud, pois esses *widgets* tem o papel de recebimento, envio e exibição de informações. Os *dashboards* do TB são formados por um conjunto de *widgets* que estão a disposição para serem utilizados, caso nenhum dos *widgets* sirva os propósitos almejados há uma opção HTML que permite qualquer edição desejada. Por fim, é importante salientar que a disposição dos *widgets* dentro do *dashboard* cumpre um papel de suma importância na hora que esse *dashboard* for visualizado na aplicação móvel.

3. Trabalhos Relacionados

Na literatura científica é possível observar diversos trabalhos que descrevem a utilização de sensores para o monitoramento de animais selvagens. Com a finalidade da proteção a vida desses animais é possível citar um sistema de monitoramento de localização e temperatura baseado em Internet das Coisas descrito por [Sangeetha et al. 2017], que auxilia na localização de animais feridos ou doentes a partir da variação da temperatura corporal, dentro de uma grande extensão geográfica, podendo ser de florestas, reservas animais, zoológicos ou parques nacionais, facilitando a busca pela localização exata do animal. O artigo em questão traz uma rica revisão bibliográfica que contém oito artigos correlatos que compõem conhecimentos imprescindíveis para o ramo de monitoramento animal através de diversos sistemas de hardware, com designs específicos para o contexto dos animais observados.

Referente a conservação das tartarugas marinhas, o Brasil possui o projeto Tamar que atua na preservação e pesquisa das tartarugas marinhas. Através do seu site, o projeto ressalta: “Todas as espécies de tartarugas marinhas continuam sendo ameaçadas por atividades humanas como a poluição do mar, a captura incidental pela pesca, e a degradação e ocupação de áreas de desova, dentre outras.”, [projeto Tamar 2022].

Tendo em vista a situação de ameaça à vida das tartarugas marinhas, foram desenvolvidas diversas plataformas que auxiliassem o trabalho de conservação de alguma maneira. Através da organização Sea Turtle Status ⁴, Craig Turley criou um pequeno guia

⁴<https://www.seaturtlestatus.org/articles/2020/2/27/top-smartphone-apps-for-sea-turtle-work>

(Figura 3) que reúne alguns aplicativos voltados a tartarugas marinhas e explicou brevemente as suas funcionalidades. Esse conjunto de aplicativos pode servir como base para o entendimento das necessidades encontradas na atividade de preservação, como também serve de apoio para o desenvolvimento de novos aplicativos.

AN OVERVIEW OF SEA TURTLE APPS		
NESTING AND IN-WATER CENSUS 	RASTR (Records Assistant for Sea Turtle Researchers) 	This app collects data on nesting, bycatch, and turtle products, with fields for morphometry, biological samples, and more. (iOS)
	Siren Turtles 	This app is used to centralize and standardize nesting data for comparative studies using SWOT minimum standards. (Android)
	Nest Tracker 	Designed by the Cayman Islands Department of Environment (DOE) to monitor all turtle-related data collection (nesting, excavations, disorientations, and more), features include mobile network backup, as well as daily autogenerated spreadsheets sent directly to DOE staff. Available by request to nesttracker.ky@gmail.com. (iOS)
	Clutch Keeper	Able to monitor all aspects of beach monitoring and nest tracking, this app uses an interactive map alert when excavation is required. (Android and iOS)
	Iris	Designed for at-sea aerial surveys, this app allows users to simply and quickly record sightings data on multiple taxa, including sea turtles. For a copy, email info@mrf-asia.org. (Android)
CITIZEN SCIENCE 	Turtles Uniting Researchers and Tourists (TURT) 	This app is designed for use in reporting turtle sightings, and products using an interactive web-based map. (Android and iOS)
	eTurtle	Users can report sea turtle sightings in the Mediterranean. (Android)
	SEAlly	This app allows users to report sea turtle and shark sightings, bycatch, and entanglement in the Mediterranean region, and is linked to an online interactive map. (Android and iOS)
	Cero Carey	This app allows tourists in Cartagena, Colombia, to identify and report hawksbill products, in Spanish, directly to the environmental police. (Android)
	iNaturalist	Not specific to sea turtles, this widely used app allows users to identify, report, and catalog biodiversity using a personal profile linked to a global network of other users. (Android and iOS)
	HerpMapper	Used for reptile and amphibian sightings, this app can be customized for sea turtle-specific projects. (Android and iOS)
	Ghost Gear Reporter 	Designed for fishers, divers, and citizen conservationists, this app allows users to report sightings, bycatch, or ghost gear entanglement. (Android and iOS)
CUSTOM PLATFORMS 	Fulcrum	This highly customizable platform requires a monthly subscription and has 20 GB of cloud storage. (Android and iOS)
	Open Data Kit (ODK) Collect 	This free, open-source app is powerful and customizable. (Android & iOS)

Figura 3. Lista de aplicativos voltados a conservação de tartarugas marinhas e suas respectivas funcionalidades. Fonte: Sea Turtle Status

Por fim, é interessante ressaltar que existem sistemas similares ao sistema proposto nesse presente artigo. Um sistema que é notável é TurtleSense descrito por [Clabough et al. 2022]. O TurtleSense utiliza um conjunto de hardware (um sensor de temperatura, um acelerômetro e um microprocessador) e possui um software que segundo [Clabough et al. 2022] realiza o processo inicial de registro das informações do ninho (data e localização do dispositivo). Também cumpre o papel de testar o sensor. Com esse sistema é possível ter uma visão ampla de como foi desenvolvido e seus experimentos, trazendo conhecimento agregado ao corrente artigo.

4. Abordagem proposta

Para a criação do aplicativo foram utilizadas duas ferramentas chaves, sendo elas a plataforma do ThingsBoard Cloud e o ThingsBoard PE mobile application. O Thingsboard

é uma plataforma IoT de código-aberto e em nuvem que permite um rápido desenvolvimento, gerenciamento e escalonamento de projetos IoT [autores do ThingsBoard]⁵. Já o ThingsBoard mobile applicaton é um projeto de código-aberto, escrito na linguagem Flutter, que pode ser modificado e customizado para suprir as necessidades de qualquer projeto IoT, servindo como extensão *mobile* da plataforma Thingsboard.

A utilização do aplicativo móvel do ThingsBoard foi uma escolha estratégica para otimizar o controle e monitoramento de ninhos de tartaruga marinha. O uso do aplicativo ThingsBoard permite a visualização de vários parâmetros essenciais, garantindo que os responsáveis pelo monitoramento estejam sempre atualizados com as condições dos ninhos em seus *smartphones*. Essa tecnologia não só simplifica o processo de coleta de dados como também enriquece a qualidade das informações obtidas, permitindo uma resposta rápida e eficiente a qualquer mudança nas condições dos ninhos.

O monitoramento dos ninhos é feito a partir de duas variáveis chaves, sendo elas temperatura e a movimentação. A temperatura é uma variável importante, pois segundo o [dos Santos et al. 2011] existe uma janela ideal de temperatura para incubação dos ovos, sendo a temperatura mínima 25°C e a máxima de 35°C, caso a temperatura monitorada esteja fora desta janela há possibilidade de falha na incubação dos ovos. Já a movimentação é uma variável chave pois ela é que permite uma melhor previsão na abertura dos ninhos, ou seja, o curto intervalo entre movimentações significa que muitos ovos já eclodiram e as tartarugas marinhas estão tentando alcançar a superfície. A identificação do período de muita movimentação traz facilidade aos responsáveis na tomada de decisão da abertura, na criação de contenções para predadores naturais e para a realização de atividades de conscientização da população. Ao invés de se fazer checagens diárias, as inspeções são realizadas em períodos de tempo mais espaçados e de maneira mais precisa.

5. Materiais e métodos

Esta seção contém uma visão geral da infraestrutura do sistema incluindo explicações sobre quais *softwares* foram utilizados, como a plataforma Thingsboard Cloud e Thingsboard PE mobile app se comunicam, qual o ambiente de desenvolvimento, em qual dispositivo físico o aplicativo móvel foi executado, como a geolocalização é obtida, como a aplicação alarma situações de perigo, como o Thingsboard Professional Edition (PE) Mobile App se comunica com o hardware e qual a composição do hardware.

5.1. Thingsboard Cloud e Thingsboard Professional Edition Mobile Application

Esta seção visa explicar a diferença entre os produtos Thingsboard que são utilizados e o motivo pelo qual o produto é utilizado dentro do projeto.

5.1.1. Thingsboard Cloud

O Thingsboard (TB) possui três distribuições de servidores, uma de código-aberto que é a *Community Edition* e duas que são pagas, sendo elas: *TB Professional Edition* e *TB Cloud*. O *Thingsboard Cloud*⁶ é uma das edições pagas de plataforma como serviço do

⁵<https://thingsboard.io/>

⁶<https://thingsboard.io/products/paas/>

Thingsboard, que serve para criar uma instância do *TB Professional Edition* sem ter que criar o próprio *host*. Todas essas distribuições são plataformas *Web* acessadas através de um navegador. A Figura 4 é uma imagem retirada diretamente do site do TB Cloud e ilustra a comparação entre as edições do TB.

Feature Comparison Matrix			
	Community Edition	Professional Edition	Cloud
Asset management & Data collection	✓	✓	✓
End-user real-time dashboards	✓	✓	✓
Customizable rule chains, widgets	✓	✓	✓
MQTT, HTTP, CoAP, OPC-UA transport	✓	✓	✓
Integrations with BigData systems	✓	✓	✓
NB-IoT, SigFox, LoRaWAN support	Basic	Advanced	Advanced
Rule Engine: Components [®]	Basic	Advanced	Advanced
Entity groups [®]	✗	✓	✓
Advanced RBAC for IoT [®]	✗	✓	✓
Scheduler [®]	✗	✓	✓
Reporting [®]	✗	✓	✓
White-labeling [®]	✗	✓	✓
CSV/XLS data export [®]	✗	✓	✓
Platform Integrations [®]	✗	✓	✓
Domain management [®]	✗	✗	✓

Figura 4. Comparação das funcionalidades entre distribuições do Thingsboard.
Fonte: ThingsBoard Cloud.

Inicialmente o projeto foi desenvolvido para utilizar a edição *community* com servidor próprio, porém esta edição não possui ferramentas de integração com a plataforma utilizada no projeto para comunicação através do protocolo LoRaWAN. Este protocolo é considerado o mais adequado para o cenário do projeto onde há: longas distâncias; demanda por baixo consumo energético; e ausência de sinal WiFi nas áreas de praia onde acontecem as desovas.

Como o fator da conexão de rede é crucial, a edição Cloud na versão 3.6.2. foi utilizada por realizar esta integração direta com a plataforma *The Things Network (TTN)* que cumpre o papel de receber dados via LoRaWAN, excluindo a necessidade do hardware possuir conexão direta com a Internet. A partir da integração as informações geradas pelos sensores são enviadas para TTN e de lá, são enviadas para o TB Cloud.

Dispositivos e Localização					
Dispositivo	Latitude	Longitude	Temperatura	Movimentação	Humidade
Ninho-Teste	0	0	30.1 °C	0	77 %
Ninho-01	-8.0441625	-34.8994907	29.9 °C	5	78 %

Figura 5. Widget que mostra informações coletadas pelos sensores do hardware.
Fonte: Autora.

Com o auxílio de um *widget*, é possível visualizar os dados enviados para a plataforma (Figura 5).

5.1.2. Thingsboard Professional Edition Mobile Application

O aplicativo móvel do Thingsboard funciona como uma extensão Thingsboard servidor, trazendo algumas de suas funcionalidades de forma adaptada para aparelhos móveis. Neste projeto a funcionalidade principal é a portabilidade dos *dashboards* e dos dispositivos, permitindo com que o usuário tenha acesso a visualização dos *dashboards* e de informações do dispositivo direto do celular, ao invés de ter que fazer login na plataforma web. Porém, essa portabilidade não permite edição, qualquer edição nos *dashboards* tem que ser feita via *Thingsboard Cloud*.

O Thingsboard Mobile está disponível em duas versões de código aberto, cada uma projetada para servir como extensão de uma distribuição específica do Thingsboard servidor. O aplicativo na edição *community* só serve de extensão para a edição *community* do TB servidor e o aplicativo na edição *professional* serve de extensão para o TB Professional Edition e o TB Cloud.

É importante destacar que existe uma distinção clara entre os produtos oferecidos. *ThingsBoard*, independentemente de sua versão, refere-se especificamente à plataforma web. Por outro lado, *Thingsboard Mobile* designa o aplicativo móvel, que funciona como uma extensão portátil da plataforma web.

A decisão de utilizar o *Thingsboard Professional Edition Mobile Application* foi feita com base na utilização do Thingsboard Cloud para o desenvolvimento dos *dashboards*, tendo em vista que a edição *community* do aplicativo móvel não se contacta com o TB Cloud.

5.2. Ambiente de desenvolvimento

A principal ferramenta de desenvolvimento utilizada durante o projeto foi o Android Studio versão Giraffe 2022 juntamente com o Flutter Software Development Kit.

5.2.1. Android Studio

O Android Studio é um *Integrated Development Environment* (IDE), ou seja, um ambiente de desenvolvimento integrado. IDE's tem a função de reunir um conjunto de ferramentas de desenvolvimento em um único ambiente, com o propósito de trazer agilidade e facilidade. Para o presente artigo, as principais funções do Android Studio utilizadas foram:

1. Editor de texto com suporte a projetos Flutter
2. Plugin de linguagem Dart
3. Compilação e execução do código
4. Emulador de android, que permite tanto emular um dispositivo móvel com um android específico, ou instalar e rodar o aplicativo em um dispositivo físico. Para este projeto, de início foi utilizado um dispositivo emulado e eventualmente substituído por um Samsung Galaxy.

5.2.2. Flutter e Flutter Software Development Kit

Flutter é um *framework* de código-aberto desenvolvido pela empresa Google que serve para o desenvolvimento de aplicações multi-plataforma a partir de um único código, ou seja, com o Flutter é possível desenvolver uma aplicação *mobile* que funciona tanto em um aparelho móvel que possui o sistema operacional Android quanto o iOS, além disso, permite a construção de aplicações web e *desktop*.

Flutter Software Development Kit (SDK), é o conjunto de compilador, depurador, componentes e bibliotecas que permitem a execução do código. Para que o Android Studio consiga compilar o projeto ele necessita que o SDK esteja instalado em máquina local, caso contrário, ele não terá funcionalidade. Se faz obrigatório o Flutter SDK instalado e propriamente configurado para que o Thingsboard Professional Edition (PE) Mobile App execute.

A página de configuração do Thingsboard PE <https://thingsboard.io/docs/pe/mobile/getting-started/> informa que para a execução do Thingsboard Mobile App se faz necessário o Flutter SDK a partir da versão 2.12.0, em diante. A versão utilizada para o desenvolvimento foi a 3.16.8.

5.3. Linguagens de programação

Este tópico aborda quais linguagens de programação foram utilizadas e qual o papel elas empregaram durante o desenvolvimento do projeto.

5.3.1. Dart

Dart é uma linguagem de programação desenvolvida pela Google com o propósito de servir para a construção de aplicações em diversas plataformas, ou seja, possui suporte para desenvolvimento de aplicações web, aplicações *mobile* (iOS, Android, etc...). É importante ressaltar que a leitura da documentação é imprescindível para qualquer desenvolvimento⁷. A linguagem Dart foi utilizada para realizar as alterações que ocorreram no código fonte do aplicativo móvel Thingsboard.

⁷Documentação de Dart - <https://dart.dev/guides>

5.3.2. JavaScript

Criada pela empresa Netscape Communications Corporation, o JavaScript é uma linguagem amplamente utilizada para desenvolvimento Web, pois permite que as páginas web sejam interativas [Flanagan 2013]. Segundo uma pesquisa feita por [Kumar and Dahiya 2019], o JavaScript está na sétima posição do ranking de linguagens mais populares, além disso a pesquisa indica que no GitHub, o JavaScript ocupa a primeira posição quando considerado os projetos desenvolvidos na linguagem.

Tendo em vista as informações supracitadas, JavaScript é a linguagem utilizada pelo TB para o desenvolvimento das funcionalidades e *mobile actions* dos *widgets*. O TB fornece um conjunto de funções em JavaScript pré-definidas e permite a edição dessas funções para que satisfaça a necessidade do desenvolvedor. A escolha do JavaScript pela plataforma para essas funções vem justamente do poder que o JavaScript possui de dar funcionalidade a elementos HTML, pois o visual dos *widgets* é feito através do HTML e CSS, e a funcionalidade é adicionada a eles através do JavaScript.

Para facilitação do entendimento a Figura 6 ilustra um código JavaScript pré-definido pelo TB que recebeu alterações que estão grifadas em amarelo. Isso demonstra a versatilidade da plataforma, pois o código pode receber alterações.

```
function processLocation(latitude, longitude, Sevent, widgetContext, entityId, entityName, additionalParams, entityLabel) {  
  1 // Function body to process current location of the phone.  
  2 // - latitude - phone location latitude  
  3 // - longitude - phone location longitude  
  4  
  5 showLocationDialog('Location', latitude, longitude);  
  6  
  7  
  8 function saveEntityLocationAttributes(latitudeAttributeName, longitudeAttributeName, latitude, longitude, deviceService) {  
  9     const entityId = {  
 10         id: "  
 11         entityType: "DEVICE"  
 12     };  
 13     if (entityId && entityId.id) {  
 14         let attributes = {  
 15             { key: latitudeAttributeName, value: latitude },  
 16             { key: longitudeAttributeName, value: longitude }  
 17         };  
 18         widgetContext.attributesService.saveEntityAttributes(entityId, "SERVER_SCOPE", attributes).subscribe(  
 19             function() {  
 20                 widgetContext.showSuccessToast('Location attributes saved!');  
 21             },  
 22             function(error) {  
 23                 widgetContext.dialogs.alert('Location attributes save failed', JSON.stringify(error));  
 24             }  
 25         );  
 26     }  
 27 }  
 28 saveEntityLocationAttributes('latitude', 'longitude', latitude, longitude);  
 29  
 30 function showLocationDialog(title, latitude, longitude) {  
 31     setTimeout(function() {  
 32         widgetContext.dialogs.alert(title, 'Latitude: '+latitude+'<br>Longitude: ' + longitude).subscribe();  
 33     }, 100);  
 34 }
```

Figura 6. Código desenvolvido para atribuir localização aos dispositivos. Fonte: Autora.

5.4. Dispositivo móvel

Com o propósito de permitir a visualização do funcionamento do aplicativo foi utilizado um Samsung Galaxy J6, com Android na versão 8.0 e que possui acesso a internet. Esse aparelho móvel teve função crucial no papel de depuração, permitindo que o aplicativo fosse executado em suas diversas fases do desenvolvimento e testado constantemente.

5.5. Configuração e instalação do Thingsboard PE Mobile App

O desenvolvimento da aplicação móvel utilizou como base o código aberto do Thingsboard PE Mobile Application na versão 1.0.9⁸. O repositório possui várias versões até o momento, para visualizar a versão mais recente basta selecionar visualmente o branch *release/1.0.9* como demonstrado abaixo na Figura 7.

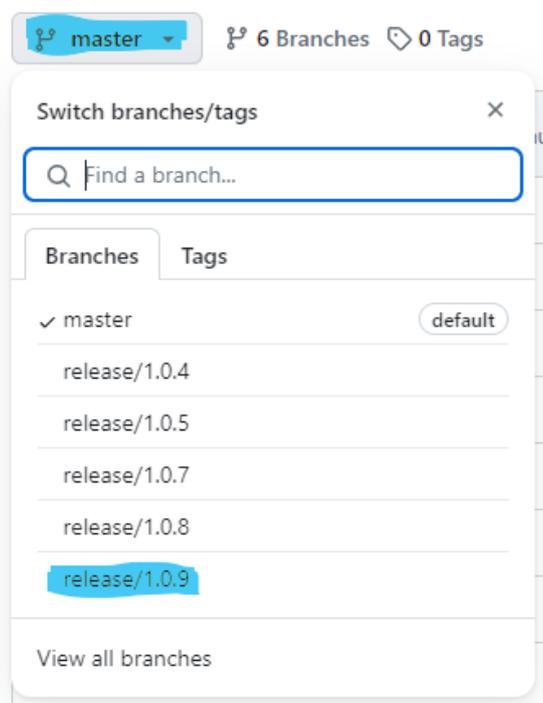


Figura 7. Versões da aplicação móvel disponíveis no repositório GitHub. Fonte: GitHub Thingsboard PE App.

O site do Thingsboard disponibiliza um guia⁹ que auxilia em todo o processo de instalação até a execução do aplicativo em uma máquina local. Para a visualização do aplicativo é necessário um emulador ou um celular.

Dentro do processo de instalação a principal atividade é criar o clone do repositório TB. Para o clone é preciso uma pasta vazia que o Android Studio possua acesso, além disso, é necessário executar o comando *git clone* dentro da pasta designada ao projeto. O *git clone* utiliza a url do repositório para criar um clone dele. A url do repositório é https://github.com/thingsboard/flutter_thingsboard_pe_app. Para clonagem de um branch ou ramo em tradução direta, o comando *git clone -b "branch"* deve ser usado, porém é obrigatória a substituição do "branch" pelo real nome do ramo que será utilizado. No caso deste projeto é o ramo *release/1.0.8*.

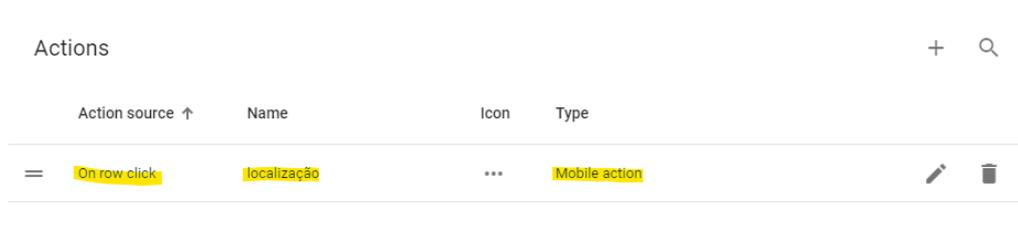
5.6. Mobile Actions

As *mobile actions* são uma funcionalidade do Thingsboard que dispara ações específicas de um dispositivo móvel. Essa funcionalidade estende o poder de controle e automação

⁸Repositório do ThingsBoard PE Mobile App - https://github.com/thingsboard/flutter_thingsboard_pe_app

⁹Guia de instalação - <https://thingsboard.io/docs/pe/mobile/getting-started/>

diretamente para as mãos dos usuários, permitindo ações como abrir a câmera do celular, capturar *screenshots*, obter a localização atual do dispositivo, entre outras. Em tradução direta se chamam ações móveis, e essas ações são configuradas no lado do servidor, ou seja, na plataforma Thingsboard Cloud no caso deste projeto, porém só podem ser ativadas através do celular que contém a aplicação móvel.



Action source ↑	Name	Icon	Type
On row click	localização	...	Mobile action

Figura 8. Configuração da ação móvel. Fonte: ThingsBoard Cloud.

Este projeto empregou apenas uma *mobile action* para retornar a localização do telefone. A *mobile action* foi atribuída ao *dashboard* de dispositivos e localização (Figura 8), e programada para apenas atribuir localização a um dispositivo caso o usuário toque uma vez na linha que contém informações do dispositivo desejado.

5.7. Integração do Thingsboard Cloud com The Things Network

Para o recebimento de dados gerados pelos sensores do dispositivo físico se faz necessária uma integração entre plataformas. Neste projeto não houve necessidade de implementação da integração, pois o Thingsboard Cloud possui um recurso (Figura 9) que permite a integração entre diversas plataformas de IoT diferentes.

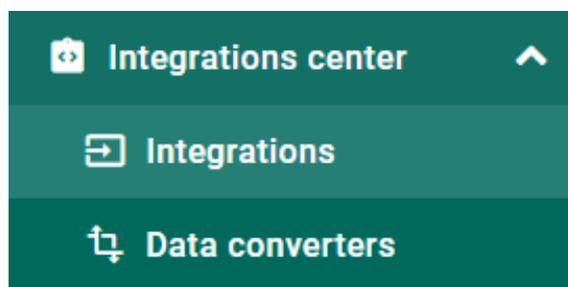


Figura 9. Função de integração do Thingsboard Cloud. Fonte: ThingsBoard Cloud.

O centro de integrações permite criação, configuração e edição de integrações, atuando como ferramenta principal do gerenciamento de integrações do TB Cloud.

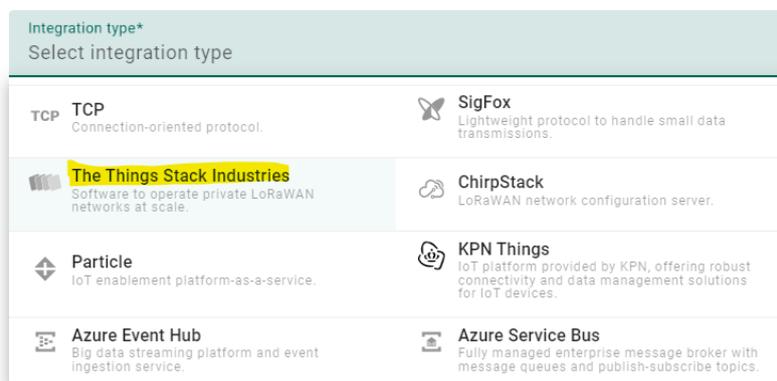


Figura 10. Função de seleção do tipo de integração

Para a utilização da integração é preciso criar uma configuração específica. No caso deste projeto o tipo de integração utilizada foi o *The Things Stack Community* (Figura 10), juntamente com a configuração da porta 1883 que é a porta padrão do protocolo MQTT, configuração do tipo de hospedagem, região e as credenciais do servidor TTN. Após a configuração dessas informações, é possível testar a conexão, caso seja estabelecida é só adicionar a integração configurada e os dados serão enviados automaticamente.

5.8. Dispositivos no Thingsboard

No Thingsboard, os dispositivos são representações virtuais de seus equivalentes físicos. Sua criação e configuração são realizadas diretamente na plataforma do Thingsboard Cloud ou por meio de integração com outras plataformas. Esses dispositivos apresentam três tipos de atributos: atributos do cliente, atributos do servidor e atributos compartilhados. Esses atributos funcionam como receptáculos para informações predefinidas, que podem ser modificadas conforme necessário. Por exemplo, um dispositivo pode inicialmente ter seus atributos de geolocalização configurados como 0 e, posteriormente, serem ajustados manual ou automaticamente para refletir a geolocalização desejada, talvez por meio de uma função JavaScript. A Figura 11 ilustra os atributos fictícios de um desses dispositivos, destacando em amarelo os atributos de geolocalização e atividade.

Server attributes			Entity attributes scope
			Server attributes
<input type="checkbox"/>	Last update time	Key ↑	Value
<input type="checkbox"/>	2024-02-21 22:56:20	active	false
<input type="checkbox"/>	2024-02-21 22:56:20	inactivityAlarmTime	1708566980229
<input type="checkbox"/>	2024-02-21 22:46:02	lastActivityTime	1708566362800
<input type="checkbox"/>	2024-02-21 22:56:39	latitude	0
<input type="checkbox"/>	2024-02-21 22:56:45	longitude	0

Figura 11. Ilustração dos atributos do servidor. Fonte: Autora.

Neste projeto os dispositivos possuem a finalidade de receber telemetrias, atributos para reivindicação e atributos de localização. A criação desses dispositivos ocorre automaticamente por meio da integração com a TTN. O servidor da TTN gera uma réplica do dispositivo e a transfere para o servidor do Thingsboard Cloud, com o qual está integrado. Após a criação da réplica, o dispositivo passa a receber automaticamente as telemetrias do dispositivo físico por meio do protocolo MQTT.

5.9. Reivindicação de dispositivo

Reivindicação de dispositivo (*claiming*) é a ação de entregar a propriedade do dispositivo para o usuário final. Quando o hardware é manufaturado e faz a conexão com o servidor TB, apenas os administradores do servidor tem acesso ao dispositivo, mesmo que ele seja entregue ao usuário final, ele não terá acesso as informações do dispositivo. Para a obtenção dessas informações o usuário reivindica o dispositivo, que permite com que ele seja dono e controle aquele dispositivo. Esta seção abordará como o *claiming* foi desenvolvido e como usuário deve fazê-lo através da aplicação.

É importante ressaltar que há uma etapa prévia à reivindicação do dispositivo, que é a adição da conta do usuário ao servidor. O usuário precisa se cadastrar na plataforma Cloud através do link <https://thingsboard.cloud/signup> com o e-mail desejado, após o cadastro o administrador do servidor. A adição da conta deve conter o e-mail do usuário final para que ele consiga fazer login na aplicação móvel e ter acesso ao servidor correto. Esta etapa é feita através do próprio Thingsboard Cloud.

Neste projeto, a reivindicação de dispositivo é realizada por meio de um procedimento executado no lado do servidor, utilizando o aplicativo móvel. Abaixo é possível visualizar o fluxograma de reivindicação (Figura 12) desenvolvido pelos autores do Thingsboard¹⁰.

¹⁰<https://thingsboard.io/docs/user-guide/claiming-devices/?claimingscenario=serverside>

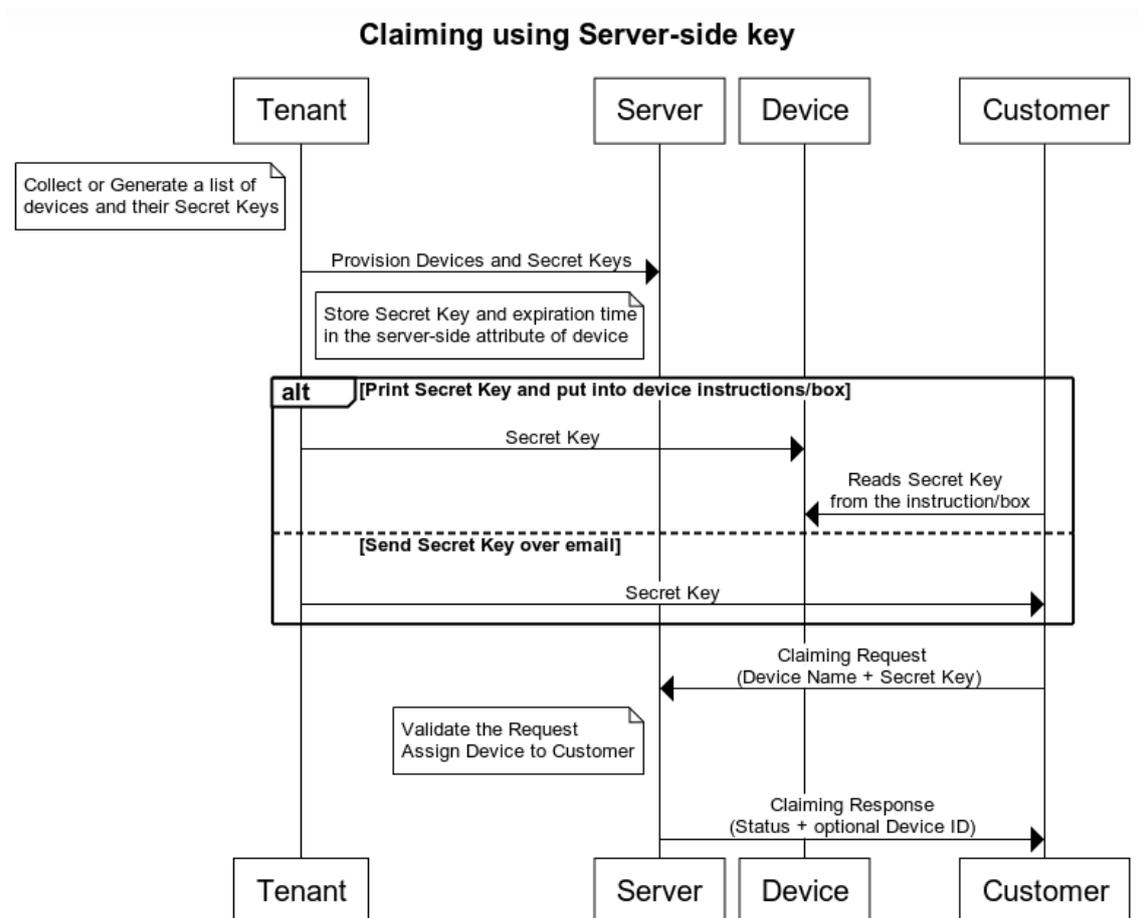


Figura 12. Fluxograma de reivindicação de dispositivo pelo lado do servidor.
Fonte: ThingsBoard.

O fluxograma acima (Figura 12) demonstra todas as etapas do processo de reivindicação utilizando a chave secreta pelo lado do servidor. Os passos são os seguintes:

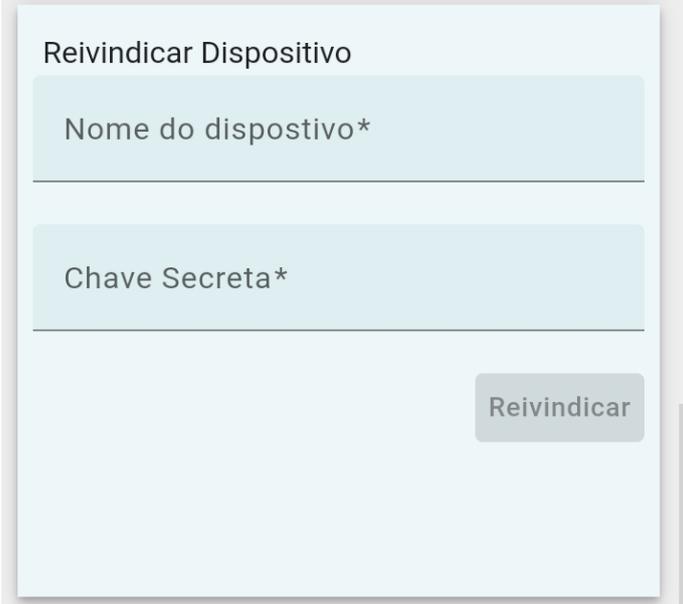
1. O administrador do servidor cria os dispositivos e suas chaves secretas
2. É adicionado no dispositivo o atributo de servidor “claimingData” e inserido manualmente nesse atributo a chave secreta de acordo com o padrão da Figura 13.
3. Posteriormente, o usuário recebe o dispositivo físico, e eventualmente a chave secreta e o nome do dispositivo, através de um e-mail.
4. O usuário insere manualmente as informações recebidas via e-mail no widget de claim.
5. O servidor valida a requisição, caso esteja tudo correto o dispositivo será atribuído ao cliente e o claim é finalizado.

É importante ressaltar que, no processo de reivindicação que utiliza chave gerada pelo lado do servidor, o administrador do servidor deve fornecer as informações necessárias ao dispositivo (Figura 13) e ao cliente. Sem essas informações, o processo de reivindicação não poderá ser concluído com sucesso.

```
{"secretKey": "YOUR_SECRET_KEY", "expirationTime": 1640995200000}
```

Figura 13. Formato da chave secreta e expiração. Fonte: ThingsBoard.

Após a configuração inicial do dispositivo, o próximo passo é a adição do *widget* de reivindicação de dispositivo (Figura 14) e modificação para que ele fique com o *design* desejado.



Reivindicar Dispositivo

Nome do dispositivo*

Chave Secreta*

Reivindicar

Figura 14. Widget de reivindicação. Fonte: ThingsBoard Cloud.

Através do *widget* de reivindicação (Figura 14) o usuário deve inserir o nome do dispositivo, a chave secreta e clicar em reivindicar. As informações solicitadas pelo *widget* de *claim* são enviadas para o usuário via e-mail. Após concluir este processo, caso as informações estejam corretas o *claim* será concluído com sucesso e o usuário poderá visualizar o dispositivo e seus dados.

5.10. Geolocalização do dispositivo

Neste projeto a localização é um parâmetro indispensável, pois a localização faz parte da identificação dos ninhos

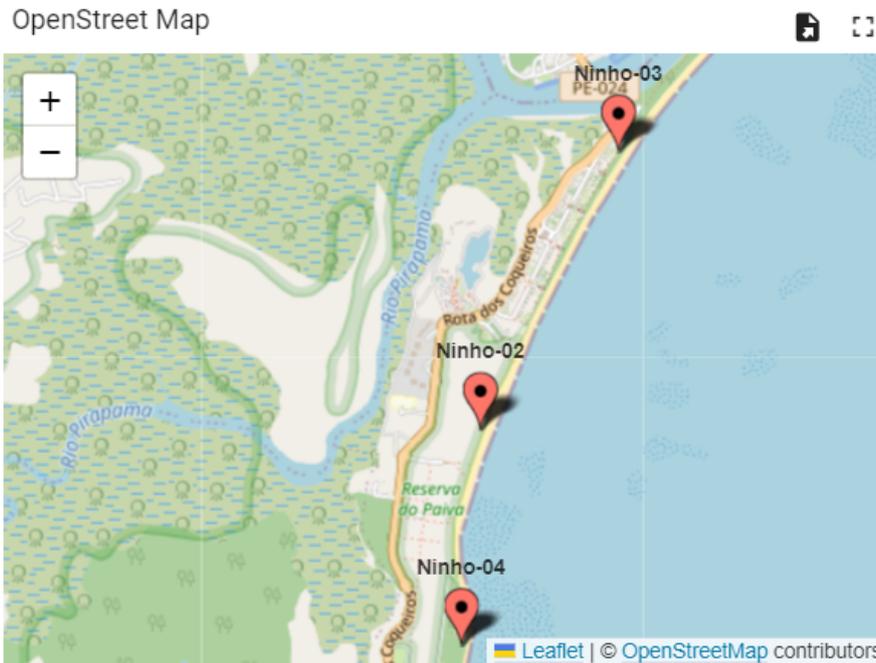


Figura 15. Imagem ilustrativa do mapa com os ninhos. Fonte: Autora.

Saber a localização do ninho é muito importante pois os alertas serão atrelados ao ninho em específico, então caso haja necessidade de alguma intervenção o responsável saberá exatamente em qual local da praia ele deve se locomover.

Para obter a geolocalização foi utilizado um esquema misto entre o usuário e o aplicativo móvel, pois o *hardware* sensor não possui GPS, então ele não consegue obter uma localização. Porém, através da *mobile action* do Thingsboard é possível obter a localização diretamente do dispositivo móvel. Após o parâmetro de localização ser obtido, automaticamente esse parâmetro será enviado para a instância de dispositivo na qual ele foi atribuída.

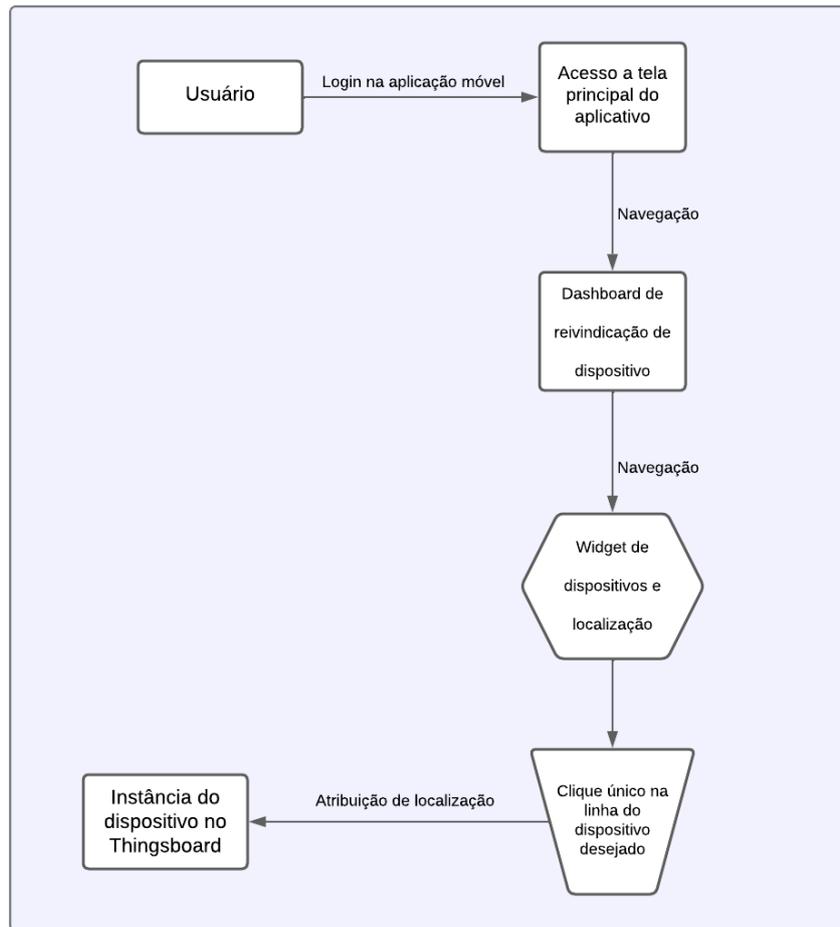


Figura 16. Fluxograma de atribuição de localização do dispositivo. Fonte: Autora.

A Figura 16 apresenta um fluxograma com os passos a serem seguidos pelo usuário para que o dispositivo receba as coordenadas de latitude e longitude.

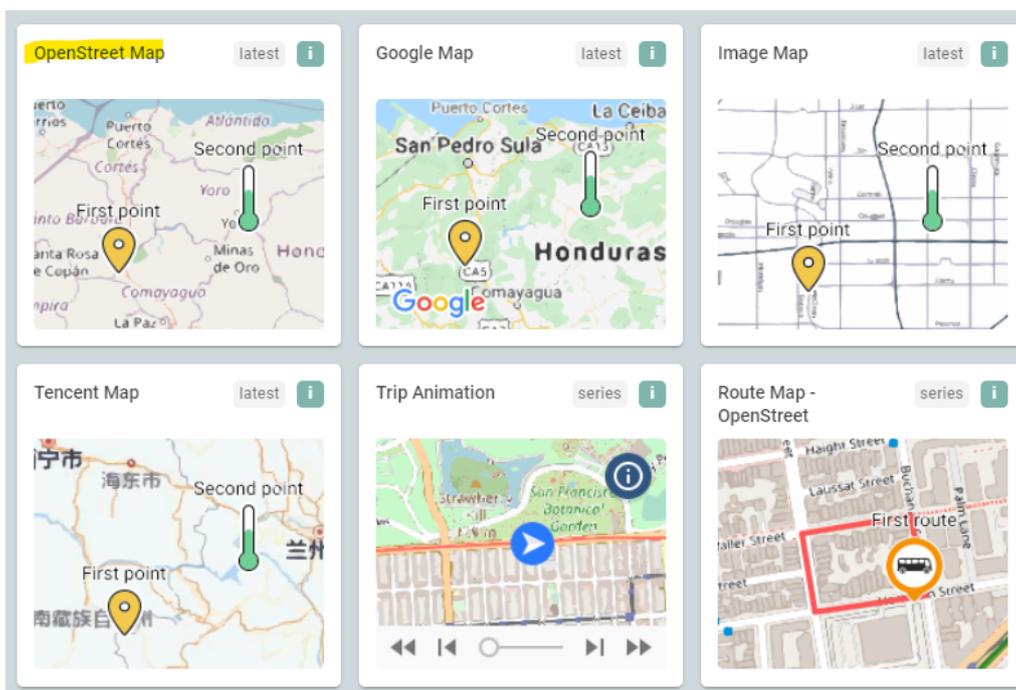


Figura 17. Figura de ilustração do mapa utilizado. Fonte: ThingsBoard Cloud.

O *widget* de mapa utilizado para mostrar a localização do dispositivo é o OpenStreet (Figura 17), apesar de haverem outras opções disponíveis na plataforma, o OpenStreet demonstrou ser a mais consistente.

5.11. Alarmes

Alarme é uma funcionalidade do Thingsboard que permite a configuração de um alerta caso alguma condição seja atingida. Para este projeto foram desenvolvidos dois alarmes, sendo eles: alarme de temperatura e alarme de movimentação.



Figura 18. Regra dos alarmes de temperatura. Fonte: Autora.

A Figura 18 descreve a lógica de alarme desenvolvida para alertar temperaturas indesejadas nos ninhos. Essa lógica funciona com duas checagens de temperatura que são descritas pelo intervalo: $A = x \in N | x \geq 35.5 \text{ and } x \leq 24.5$. Caso a temperatura exceda qualquer um dos limites definidos, um alerta crítico será enviado.

5.12. Dispositivo sensor

Apesar dos dados poderem ser simulados através da própria plataforma do Thingsboard, o projeto recebeu apoio de um hardware externo que cumpriu o papel de enviar telemetrias reais para o aplicativo, possibilitando chegar mais próximo do que seria o resultado final do projeto.

Nesta seção serão apresentados em um tópico único os componentes eletrônicos que compõem o dispositivo de sensores. Vale ressaltar que a escolha dos componentes foi feita considerando aspectos de eficiência energética.

5.12.1. Componentes eletrônicos

O esquema eletrônico do dispositivo é feito a partir dos seguintes componentes listados:

- IoT DevKit v1.0 da Robocore

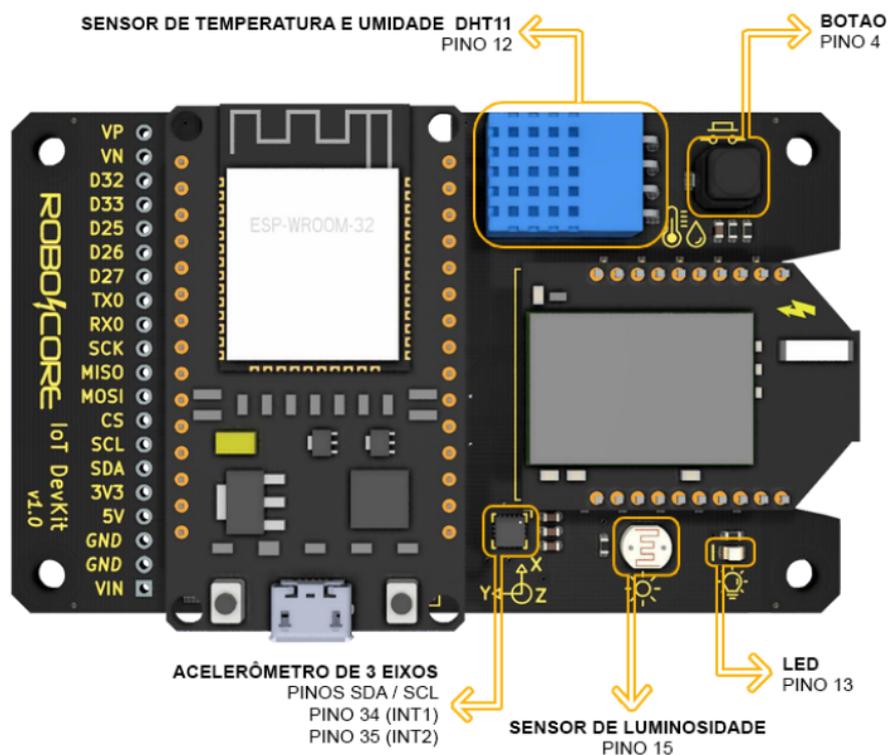


Figura 19. IoT DevKit da Robocore. Fonte: Site Robocore.

A figura 19 foi retirada diretamente do site do Robocore¹¹ e demonstra o esquema do conjunto de desenvolvimento Robocore. Esse conjunto de peças da Robocore contém:

1. Microprocessador ESP-WROOM-32 que possui Wi-Fi e Bluetooth integrados.
2. Sensor de umidade e temperatura integrados em um único módulo (DHT11)
3. Módulo acelerômetro MMA8452 de três eixos
4. Módulo LoRaWAN Bee V2

¹¹<https://www.robocore.net/lorawan/iot-devkit-lorawan>

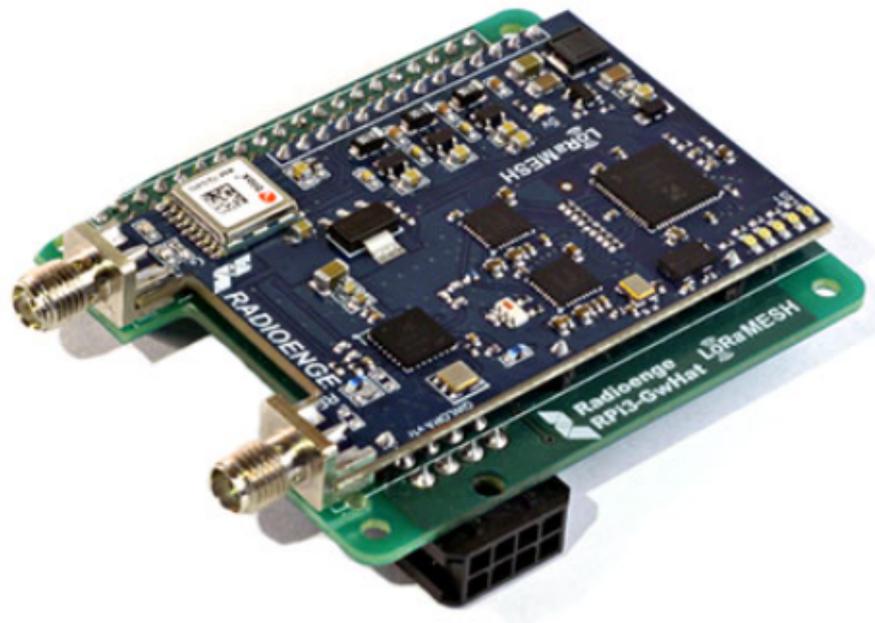


Figura 21. Exemplo de gateway LoRaWAN. Fonte: Radioenge.

A placa de *Gateway LoRaWAN* foi utilizada como ponte de comunicação entre o dispositivo de sensores e o servidor The Things Network (TTN), desempenhando o papel de emitir os dados por meio do protocolo MQTT para o servidor em nuvem TTN. Em termos gerais o dispositivo envia os dados via LoRa para o gateway, e o gateway envia os dados via MQTT para o servidor TTN.

É importante ressaltar que o dispositivo sensor não foi desenvolvido no contexto deste trabalho específico, mas sim como parte de outro trabalho de conclusão de curso realizado em paralelo. No entanto, conseguimos utilizar o dispositivo para realizar uma simulação do uso do aplicativo em um cenário mais próximo do real.

6. Resultados

O trabalho proposto teve como resultado um aplicativo móvel construído com base no Thingsboard PE Mobile Application, que reúne em um único ambiente a capacidade de instalação de dispositivos sensores e do monitoramento dos dados pertinentes a conservação das espécies de tartarugas marinhas incidentes no Brasil. Apesar da aplicação ainda não ter sido testada em campo, ela se demonstrou totalmente funcional nos testes realizados com o dispositivo sensor. A imagem a seguir (Figura 22) é um *screenshot* da página inicial da aplicação e apresenta os *dashboards* disponíveis.



Bluetooth, Wi-Fi, Signal, 50%, Battery, 19:43

ThingsBoard
Professional

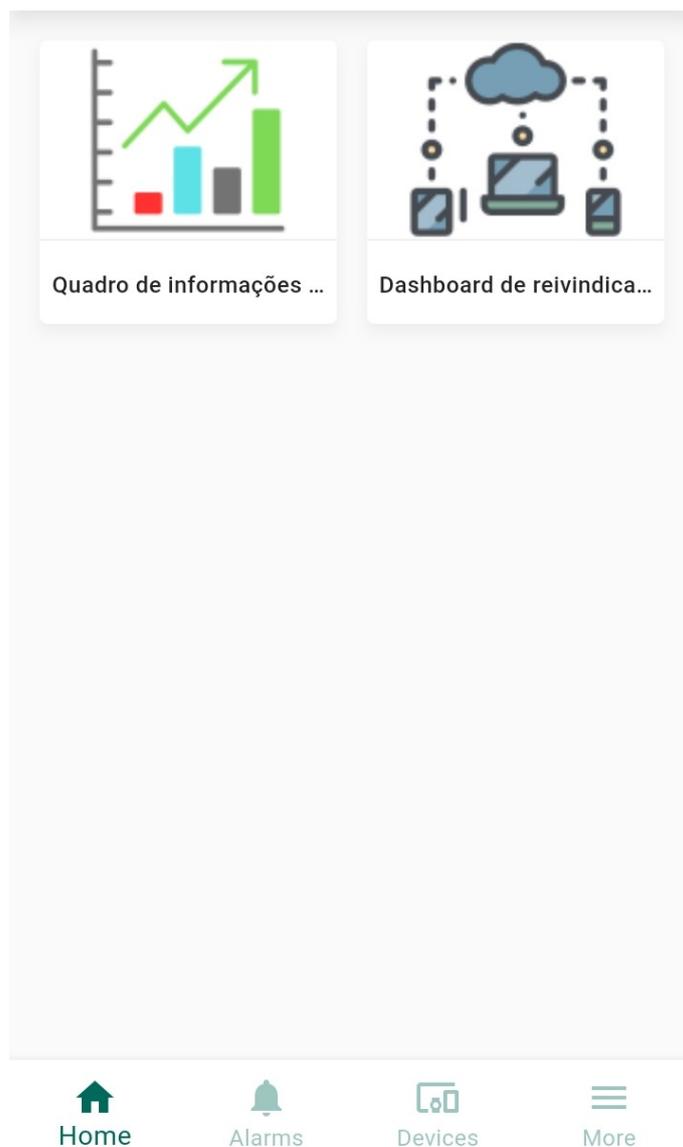


Figura 22. Tela de dashboards disponíveis. Fonte: Autora.

Cada *dashboard* desempenha um papel essencial dentro da aplicação, atendendo a propósitos distintos que facilitam a gestão dos dispositivos e o monitoramento das telemetrias. O *dashboard* dedicado às informações dos ninhos é projetado para centralizar dados cruciais, tais como telemetria, localização e alarmes, permitindo uma visão abrangente do estado e atividade dos dispositivos. Paralelamente, o *dashboard* de reivindicação é fundamental para que os usuários possam efetuar a reivindicação de dispositivos, concedendo-lhes a capacidade de integrar esses dispositivos à aplicação, controlá-los e, mais importante, atribuir geolocalização a cada um deles.

O processo começa no *dashboard* de reivindicação, onde o usuário, seguindo os

procedimentos detalhados na seção 5.9, inicia a integração dos dispositivos. Esta etapa é crucial para trazer as informações do *hardware* para a aplicação. Para a configuração da geolocalização específica para cada dispositivo se faz necessário seguir os procedimentos da seção 5.10, estabelecendo assim a base para uma gestão eficaz e um monitoramento preciso.

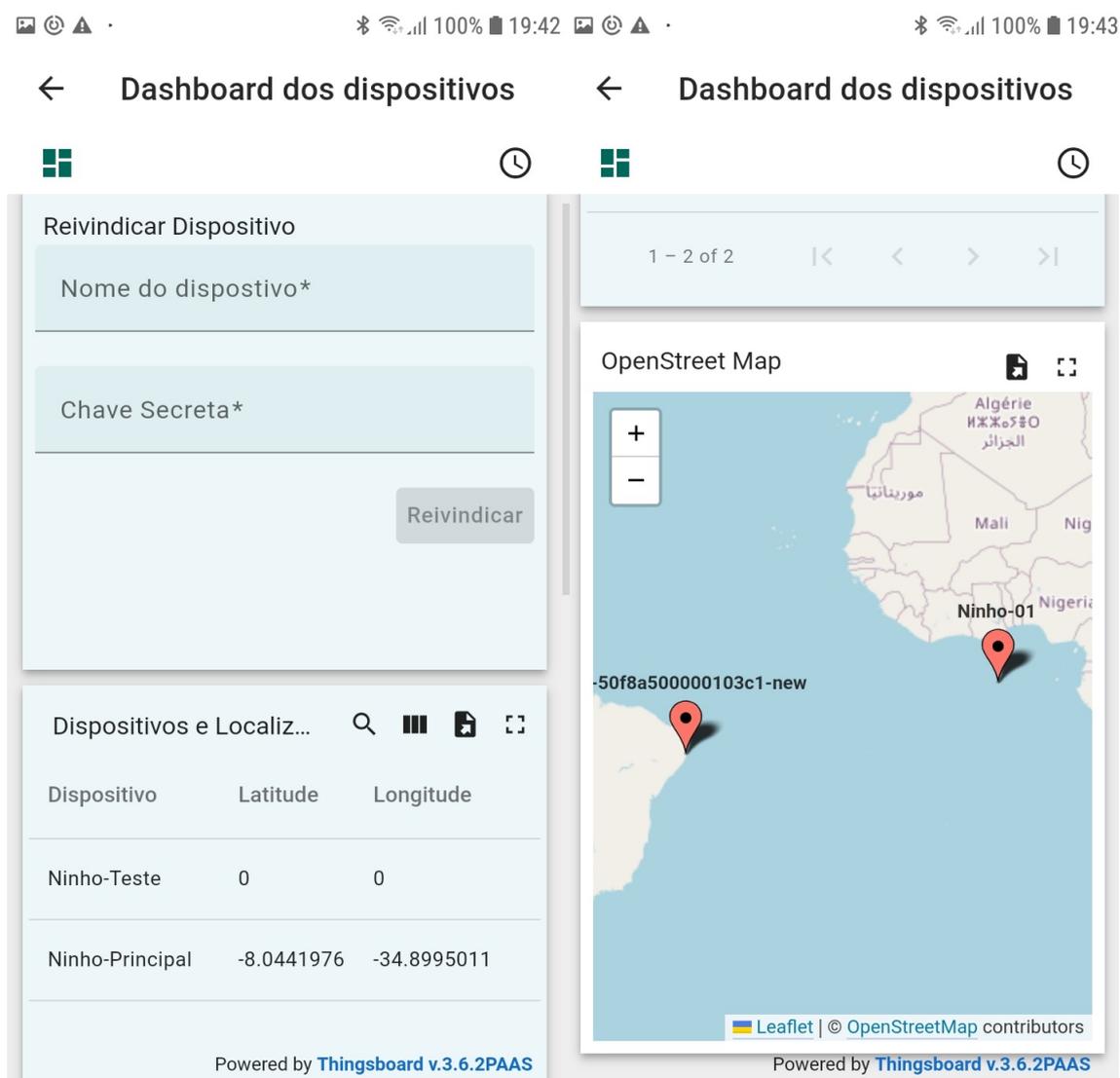


Figura 23. Visão da tela de reivindicação. Fonte: Autora.

Na tela de reivindicação (Figura 23), é apresentado o resultado final da funcionalidade de reivindicação e atribuição de geolocalização aos dispositivos. O dashboard de reivindicação é equipado com os *widgets* essenciais a tarefa de *claim*, incluindo um mapa interativo que permite ao usuário confirmar a localização do dispositivo. Este recurso cartográfico foi adicionado exclusivamente para que o usuário tenha uma confirmação da localização atribuída ao dispositivo.

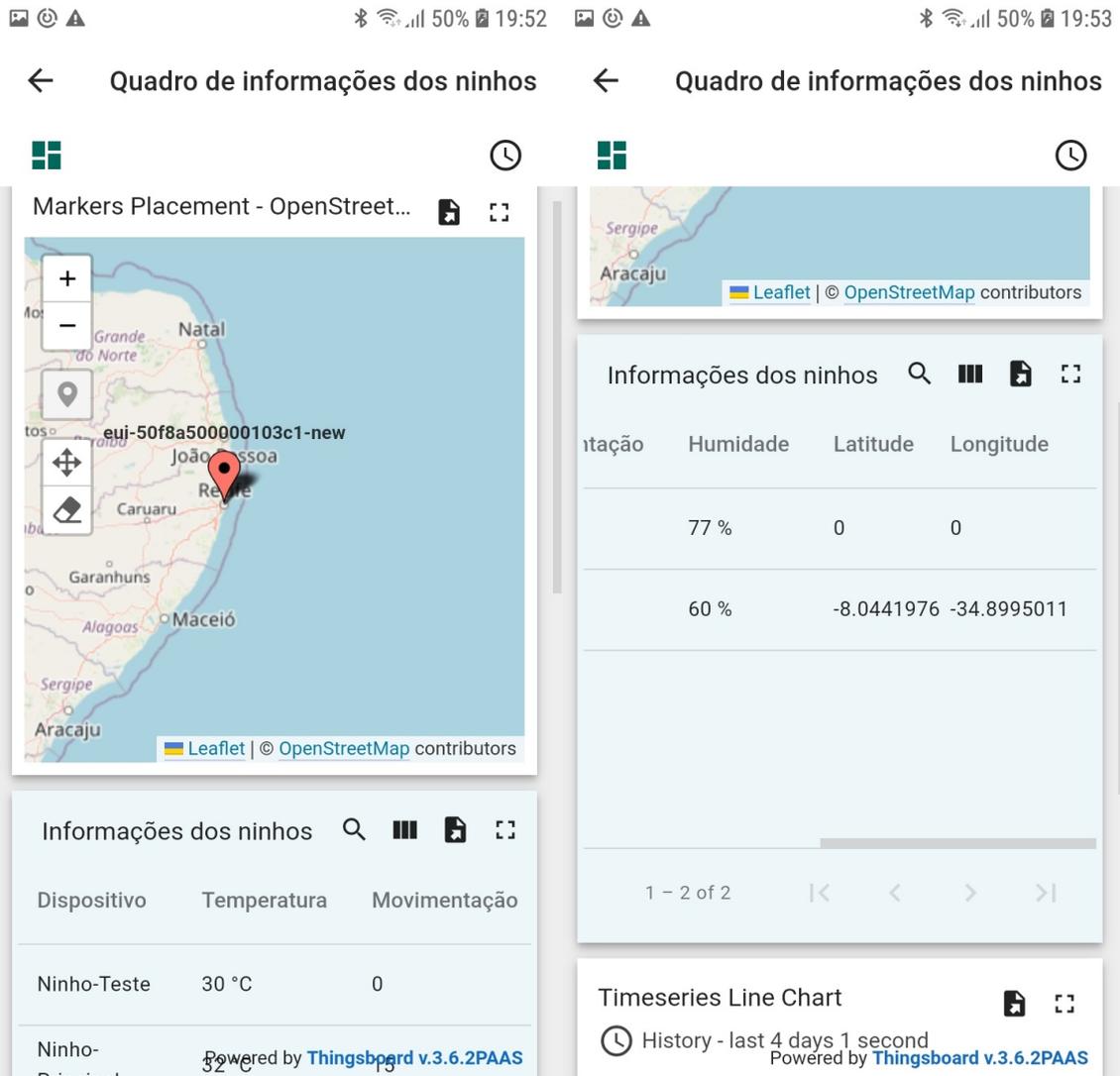


Figura 24. Visão do dashboard via aplicativo móvel. Fonte: Autora.

Abaixo é possível visualizar o produto final do quadro de informações (Figura 25). Através dessa imagem é possível perceber que o projeto obteve bons resultados no que diz respeito ao funcionamento como extensão do Thingsboard servidor.

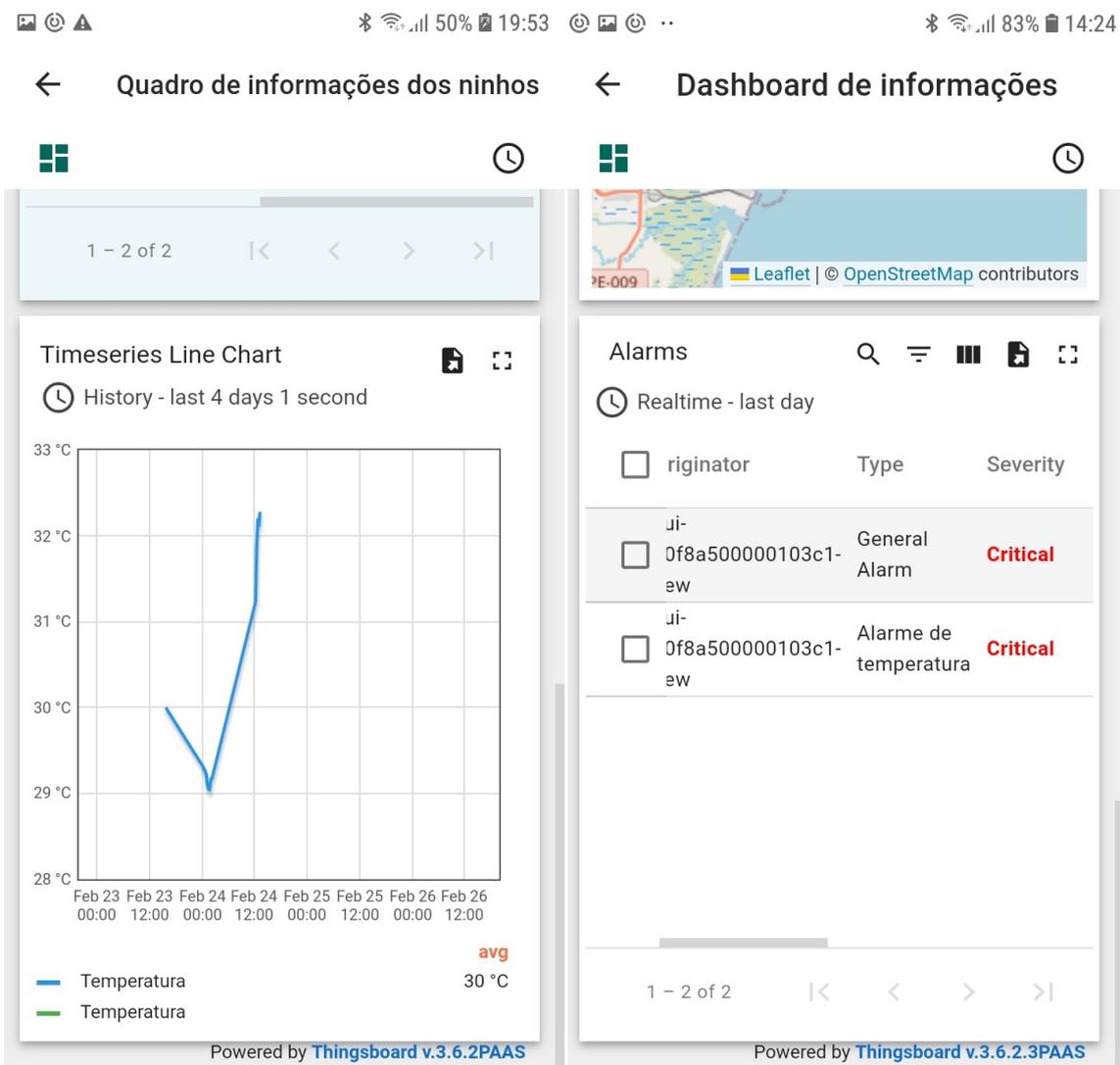


Figura 25. Tela com gráfico e lista de alarmes. Fonte: Autora.

Por fim, a Figura 25 exibe a seção final do dashboard, destacando o gráfico que mostra a variação da temperatura ao longo do tempo juntamente com a lista de alarmes. O aplicativo enriquece a experiência ao vincular a localização, via GPS do smartphone, ao dispositivo físico equipado com sensores. Isso permitiu o monitoramento das telemetrias coletadas próximas ao tempo real e a emissão de alertas automáticos caso as condições detectadas se desviarem dos parâmetros ideais para a incubação dos ovos de tartaruga marinha.

7. Conclusões

O objetivo principal do projeto relatado neste artigo foi de desenvolver uma aplicação móvel que cumprisse o papel de dar apoio à ferramentas de monitoramento de ninhos de tartaruga marinha. O projeto foi desenvolvido especificamente através da plataforma Thingsboard e seus produtos. Essa escolha deve-se ao fato de que a plataforma Thingsboard se mostrou altamente eficaz no desenvolvimento de soluções para Internet das Coisas,

destacando-se como uma ferramenta poderosa nesse domínio, e também, permitindo que o projeto fosse financeiramente acessível.

Este trabalho evidenciou diversas opções de desenvolvimento para uma ferramenta de apoio ao monitoramento de tartarugas marinhas. Embora a abordagem seguida tenha sido centrada na plataforma Thingsboard, é importante ressaltar que existem diversos serviços de código aberto disponíveis para alcançar resultados similares.

7.1. Trabalhos futuros

Embora tenha atingido os objetivos específicos, o projeto apresenta um conjunto de possíveis aprimoramentos que podem ser implementados. O cenário ideal inclui a substituição do servidor Thingsboard Cloud por um servidor dedicado, contudo, para viabilizar essa mudança, seria necessário implementar uma função de integração entre os servidores.

Além disso, o processo de reivindicação pode ser aprimorado para oferecer maior facilidade ao usuário. Em vez de exigir que o usuário insira manualmente os dados de claim recebidos por e-mail, esse processo poderia ser substituído para ocorrer de forma automatizada. Nesse caso, as informações de claim poderiam ser incorporadas em um código QR, e a função de leitura do aplicativo seria configurada para enviar automaticamente esses dados para o widget de claim assim que o código QR fosse escaneado pelo usuário.

Outro quesito de aprimoramento seria a elaboração de relatórios a partir dos dados coletados pelos sensores. A aplicação desenvolvida demonstra um grande potencial no agrupamento de informações, podendo, futuramente, incluir uma interface que apresente uma série de gráficos para ilustrar a variação dos dados ao longo do tempo, similarmente ao que já é feito atualmente com os dados de temperatura.

Em conclusão, é importante destacar que para futuras pesquisas, a realização de testes em campo trará um valor significativo ao projeto. Os testes em campo possibilitam a coleta de dados reais e o desenvolvimento de novas funcionalidades para o aplicativo, com base no funcionamento do mesmo dentro de um cenário real.

Referências

- Benakouche, T. (1999). *TECNOLOGIA É SOCIEDADE: CONTRA A NOÇÃO DE IMPACTO TECNOLÓGICO*. PhD thesis, Universidade da Califórnia.
- Clabough, E., Kaplan, E., Hermeyer, D., Zimmerman, T., Chamberlin, J., and Wantman, S. (2022). The secret life of baby turtles: A novel system to predict hatchling emergence, detect infertile nests, and remotely monitor sea turtle nest events. *plos one*, 17(10):0275088.
- dos Santos, A. S., de Pádua Almeida, A., Santos, A. J. B., Gallo, B., Giffoni, B., and Baptistotte, C. (2011). Plano de ação nacional para a conservação das tartarugas marinhas.
- Flanagan, D. (2013). *JavaScript: o guia definitivo*. bookman.
- free Software Foundation (2022). *The Free Software Foundation (FSF) is a nonprofit with a worldwide mission to promote computer user freedom*. Disponível em: <https://www.fsf.org/> Acessado em 08/02/2024.

- Hillar, G. C. (2017). *MQTT Essentials - A Lightweight IoT Protocol*. Packt.
- Kumar, K. and Dahiya, S. (2019). Programming languages: A survey. *International Journal on Recent and Innovation Trends in Computing and Communication*, 5(5):307–313.
- Martins, M. and Molina, F. (2008). *Livro Vermelho da Fauna Brasileira Ameaçada de Extinção*, chapter Panorama geral dos répteis ameaçados do Brasil. Ministerio do Meio Ambiente e Fundação Biodiversitas.
- McEwen, A. and Cassimally, H. (2014). *Designing the Internet of Things*. John Wiley and Sons Ltd.
- Pereira, F. V. (2023). *CONDOMÍNIO ATLÂNTICO: ESTUDOS E DEBATES*, chapter A POLUIÇÃO DO OCEANO: IMPACTOS E POSSIBILIDADES. Editora textos.
- projeto Tamar (2022). *Como estão as tartarugas marinhas na nova lista de espécies ameaçadas*. Disponível em: <https://www.tamar.org.br/> Acessado em 02/02/2024.
- Sangeetha, V., Pavithra, S., Shanmugasundaram, R., Tamilselvan, S., and Ahmed, A. H. T. (2017). Iot based animal tracking and monitoring system in zoo. *South Asian Journal of Engineering and Technology*, 3(2):162–168.
- Sarmiento, A. M. S. (2013). Determinação de pesticidas organoclorados em tecidos de tartarugas-verdes (*Chelonia mydas*) provenientes da costa sudeste do Brasil: estudo da ocorrência em animais com e sem fibropapilomatose.
- Semtech (2024). *What is LoRa*. Disponível em: <https://www.semtech.com/lorawhat-is-lora> Acessado em: 16/02/2024.
- Timko, R. E. and Kolz, A. L. (1982). *Marine Fisheries Review*, chapter Satellite sea turtle tracking, pages 19–24. National Marine Fisheries Service.