

**UNIVERSIDADE FEDERAL DE ITAJUBÁ – UNIFEI**



**DESENVOLVIMENTO DE UM SISTEMA EM NUVEM PARA  
MONITORAMENTO DE ARMADILHAS OVITRAPS  
(SMART OVITRAPS)**

Daniel Pinheiro dos Reis

UNIFEI  
Itajubá  
2021

**DESENVOLVIMENTO DE UM SISTEMA EM NUVEM PARA  
MONITORAMENTO DE ARMADILHAS OVITRAPS  
(SMART OVITRAPS)**

Daniel Pinheiro dos Reis

Monografia apresentada como trabalho final de graduação, requisito parcial para obtenção do título de Bacharel em Ciência da Computação, sob orientação do Prof. Dr. Adler Diniz de Souza.

UNIFEI  
Itajubá  
2021

*"O insucesso é apenas uma oportunidade para recomeçar com mais inteligência"*  
*(Henry Ford)*

## **AGRADECIMENTOS**

Agradeço aos meus pais, por me darem a oportunidade do estudo, à todos os meus professores e amigos por partilharem o conhecimento comigo e por poder estar concluindo esse ciclo.

## **RESUMO**

A intensificação dos arbovírus (dengue, zika, chikungunya, etc) destaca a necessidade de um controle eficaz do seu vetor de transmissão, o mosquito Aedes. Para isso, foram criadas as Ovitraps, armadilhas que capturam os ovos depositados pelas fêmeas impedindo a multiplicação da população de mosquitos. Apesar dos vários modelos de Ovitraps existentes poucos, ou quase nenhum deles, aplicam algum conceito de tecnologia em seu desenvolvimento, tais modelos necessitam de um profissional que realize o trabalho de campo de coleta das armadilhas e de um profissional para análise dos dados coletados. Este trabalho busca desenvolver uma ovitrap que facilite o trabalho de campo de coleta e de análise de dados, aplicando sensores no desenvolvimento de uma ovitrap "inteligente" capaz de capturar os dados e disponibiliza-los na nuvem para que possam ser acessados de maneira remota em qualquer lugar com acesso a internet.

**Palavras-chave:** Armadilha, Ovitrap, Dengue, Zika e Chikungunya.

## **ABSTRACT**

The intensification of arboviroses (dengue, zika, chikungunya, etc.) highlights the need for effective control of its transmission vector, the female Aedes mosquito. For this purpose, Ovitrap were created, traps that capture the eggs deposited by the females, preventing the multiplication of the mosquito population. Despite the several existing Ovitrap models, few or almost none of them apply any concept of technology in their development, such models need a professional to carry out the fieldwork to collect the traps and a professional to analyze the collected data. This work seeks to develop a platform that facilitates the field work of collecting and analyzing data. For this, sensors were used in the development of a "smart" device capable of capturing data and making it available in the cloud so that it can be accessed remotely anywhere with Internet access.

**Palavras-chave:** Traps, Ovitrap, Dengue, Zika and Chikungunya.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Primeiro modelo de ovitrap, desenvolvido na década de 70.. . . . .	14
Figura 2 – (A) Standard Lethal Ovitrap (LO), (B) National Environmental Agency Singapore Sticky Ovitrap (SO), (C) MosquiTRAP. Sticky Ovitrap (SO), (D) Biogents Gravid Aedes Trap (GAT), (E) Centers for Disease Control (CDC) Autocidal Gravid Ovitrap(AGO) . . . . .	15
Figura 3 – Modelo de ovitrap desenvolvido por Ismaliza Isa et al. . . . .	16
Figura 4 – Primeiros testes, módulo NodeMCU, um sensor laser (o mesmo utilizado no protótipo final),, um sensor ultrassônico, um módulo ESP32-CAM e uma câmera GoPro . . . . .	17
Figura 5 – Estrutura de pastas de fotos para larvas e mosquitos no Amazon S3. . . . .	18
Figura 6 – Foto registrada pelo ESP32-CAM. . . . .	18
Figura 7 – Padrões em ovitraps. . . . .	19
Figura 8 – Protótipo de ovitrap desenvolvido neste trabalho (sem sensores). . . . .	20
Figura 9 – Vaso maior. . . . .	21
Figura 10 – Vaso menor. . . . .	21
Figura 11 – Prato. . . . .	21
Figura 12 – Boleira. . . . .	22
Figura 13 – Tampa da boleira após o recorte. . . . .	22
Figura 14 – Tela mosquiteira. . . . .	22
Figura 15 – Posição final da tela mosquiteira. . . . .	22
Figura 16 – Breu. . . . .	23
Figura 17 – Abertura com prato superior. . . . .	23
Figura 18 – Sensores conectados e fixos ao prato da armadilha. . . . .	24
Figura 19 – Esquema de ligação dos sensores. . . . .	25
Figura 20 – NodeMCU. . . . .	25
Figura 21 – VL53L0X-V2 (Time-of-Flight). . . . .	26
Figura 22 – Diagrama de infraestrutura. . . . .	26
Figura 23 – Diagrama do banco. . . . .	27
Figura 24 – Tema claro. . . . .	28
Figura 25 – Tema escuro. . . . .	28
Figura 26 – Tela de login. . . . .	29
Figura 27 – Tela de listagem de armadilhas. . . . .	29
Figura 28 – Tela de cadastro de armadilhas. . . . .	30
Figura 29 – Visão geral do sistema. . . . .	31
Figura 30 – Protótipo final da armadilha em funcionamento. . . . .	32
Figura 31 – Diagrama do sistema . . . . .	40
Figura 32 – Diagrama de Casos de Uso . . . . .	42

## **LISTA DE TABELAS**

Tabela 1 – Dimensões vaso maior. . . . .	21
Tabela 2 – Dimensões vaso menor. . . . .	21
Tabela 3 – Dimensões prato. . . . .	21
Tabela 4 – Dimensões boleira. . . . .	22
Tabela 5 – Tabela de custo. . . . .	33
Tabela 6 – Tabela de resultados. . . . .	34

## **LISTA DE ABREVIATURAS E SIGLAS**

API	Applications Protocol Interface
HTTP	HyperText Transfer Protocol
AWS	Amazon Web Services
S3	Amazon Simple Storage Service
EC2	Amazon Elastic Compute Cloud
MUI	Material-UI Artificiais

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>12</b>
1.1	Motivação . . . . .	12
1.2	Justificativa . . . . .	12
1.3	Objetivos . . . . .	13
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA . . . . .</b>	<b>14</b>
2.1	O que é uma ovitrap? . . . . .	14
2.2	Evolução das ovitraps . . . . .	14
<b>3</b>	<b>DESENVOLVIMENTO . . . . .</b>	<b>17</b>
3.1	A ideia inicial . . . . .	17
3.2	Ovitrap . . . . .	18
3.2.1	Vaso maior . . . . .	20
3.2.2	Vaso menor . . . . .	21
3.2.3	Prato . . . . .	21
3.2.4	Boleira . . . . .	22
3.2.5	Tela mosquiteira . . . . .	22
3.2.6	Breu . . . . .	23
3.2.7	Palitos de churrasco e fitas hellerman . . . . .	23
3.3	Hardware . . . . .	23
3.3.1	NodeMCU . . . . .	25
3.3.2	VL53L0X (Time-of-Flight) . . . . .	26
3.4	Software . . . . .	26
3.4.1	Banco de dados . . . . .	26
3.4.2	API . . . . .	27
3.4.3	Aplicação Cliente . . . . .	27
<b>4</b>	<b>VISÃO GERAL DO SISTEMA . . . . .</b>	<b>31</b>
4.1	Alimentação . . . . .	31
4.2	Comunicação . . . . .	32
<b>5</b>	<b>CUSTOS . . . . .</b>	<b>33</b>
<b>6</b>	<b>VALIDAÇÃO E RESULTADOS . . . . .</b>	<b>34</b>
<b>7</b>	<b>CONCLUSÃO E TRABALHOS FUTUROS . . . . .</b>	<b>35</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>36</b>
	<b>APENDICE A - DOCUMENTO DE REQUISITOS . . . . .</b>	<b>38</b>

## 1 INTRODUÇÃO

Dengue é a doença viral mais comum no mundo transmitida por mosquitos (BHATT et al., 2013). Definida como uma doença febril, aguda e não contagiosa, a dengue é um dos arbovírus mais importantes que afetam o homem. Isso se deve à sua alta incidência e potencial de disseminação. É transmitida por *Aedes aegypti* (Linnaeus, 1762) e *Aedes albopictus* (Skuse, 1894) infectados, sendo o *Aedes aegypti* o vetor urbano responsável pela transmissão no Brasil (ANDRIOLI et al., 2020).

O Brasil registrou 1.544.987 casos de dengue em 2019, um aumento de 488% em relação a 2018, segundo dados do Ministério da Saúde. Desse total, 782 pessoas morreram em todo o país (G1, 2020).

Em 2020, até 14 de novembro, foram registrados 971.136 casos de dengue no país. Os estados com as maiores taxas de incidência são o Paraná, Mato Grosso do Sul, Distrito Federal e Mato Grosso. Nesse período foram confirmados 528 óbitos por dengue, 76,8% deles no Paraná, São Paulo, Distrito Federal e Mato Grosso do Sul. Também até o dia 14 de novembro, foram notificados 78.808 casos de chikungunya, sendo 67,4% na Bahia e Espírito Santo. Foram confirmados 25 óbitos por chikungunya nesse período. Em 2020, até 24 de outubro, foram notificados 7.006 casos de zika. Os estados da Bahia e Rio Grande do Norte concentravam 45,8% dos casos de zika (HEALTH, 2020).

### 1.1 Motivação

A intensificação dos arbovírus destaca a necessidade de um controle eficaz do seu vetor de transmissão, o mosquito *Aedes*. A principal medida de prevenção dos arbovírus é o controle da população do mosquito. A implementação de medidas preventivas requer ferramentas de vigilância eficientes que permitam prever a população real de mosquitos (ISA et al., 2019). Uma ferramenta de controle que vem sendo utilizada desde a década de 70 são as ovitraps (LOK et al., 1977).

### 1.2 Justificativa

Apesar da variedade de modelos de ovitraps, a grande maioria deles necessitam de um agente que faça o trabalho de campo de coletar os dados e prestar manutenção nas armadilhas. Este trabalho geralmente é feito por um ou mais agentes que vão até o local da armadilha para coletar os ovos e mosquitos para análises futuras. Tomando como exemplo um cenário com mais de 100 armadilhas, o trabalho de coleta de todas as armadilhas consumiria um tempo de deslocamento do agente proporcional ao número de armadilhas. Este tempo poderia ser reduzido utilizando a solução proposta por esse trabalho, em que a quantidade capturada de mosquitos e a localização de cada armadilha são exibidos no "dashboard" interativo da aplicação web, facilitando a análise e posterior tomada de decisão.

### 1.3 Objetivos

O objetivo principal deste trabalho é desenvolver uma aplicação web do tipo "dashboard" que permita o monitoramento a distância de armadilhas ovitraps (Software). Além da aplicação web também foram desenvolvidos: um protótipo de armadilha (Ovitrap) e o sistema embarcado da mesma (Hardware).

- Software:
  - Desenvolver uma API para receber dados de múltiplas armadilhas, tratá-los e armazená-los de maneira segura em um banco de dados.
  - Disponibilizar os dados armazenados para que sejam consumidos por outras aplicações.
  - Desenvolver uma aplicação web para consumir os dados fornecidos pela API.
- Ovitrap:
  - Desenvolver uma ovitrap que atraia e capture mosquitos.
- Hardware:
  - Contar quantos mosquitos passaram pela armadilha através de sensores.
  - Enviar dados coletados da armadilha de maneira segura para a API.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 O que é uma ovitrap?

Ovitraps são armadilhas desenvolvidas para capturar larvas e ou mosquitos. A primeira ovitrap que se tem informação é creditada a Lok e outros, 1977 (LOK et al., 1977). A ovitrap de Lok consiste de um recipiente cilíndrico preto, cheio de água, com uma abertura de malha trançada no topo, com duas pás de madeira sob ela. Embora a fêmea do mosquito que deposita seus ovos na armadilha não seja morta, os filhotes que eclodem dos ovos ficam presos pela malha trançada e morrem afogados.

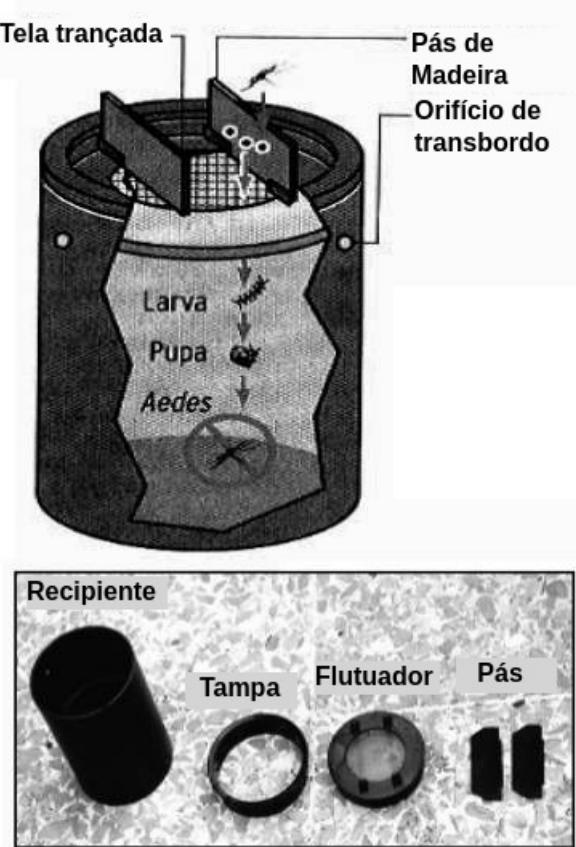


Figura 1 – Primeiro modelo de ovitrap, desenvolvido na década de 70..

FONTE: Adaptado de (EE et al., 2006)

### 2.2 Evolução das ovitraps

Na evolução das ovitraps foram desenvolvidas ovitraps letais, no sentido de capturar e matar o mosquito fêmea. A primeira delas utilizava uma fita tratada com inseticida nas paredes do seu interior, que matava as fêmeas atraídas pela água, porém foi observado que o mosquito ganhava resistência ao inseticida ao longo do tempo (JOHNSON et al., 2017). Depois foi desenvolvido um modelo que ao invés de uma fita com inseticida utilizava uma fita adesiva que capturava a fêmea do mosquito (JOHNSON et al., 2017). Apesar de eficientes e baratas, as ovitraps até então eram

pequenas, o que além de exigir manutenção em curtos períodos de tempo, não eram tão atrativas as fêmeas do mosquito (JOHNSON et al., 2017). Foi desenvolvido então, modelos maiores, mais atrativos às fêmeas do mosquito e que demandavam manutenções em períodos de tempo maiores.

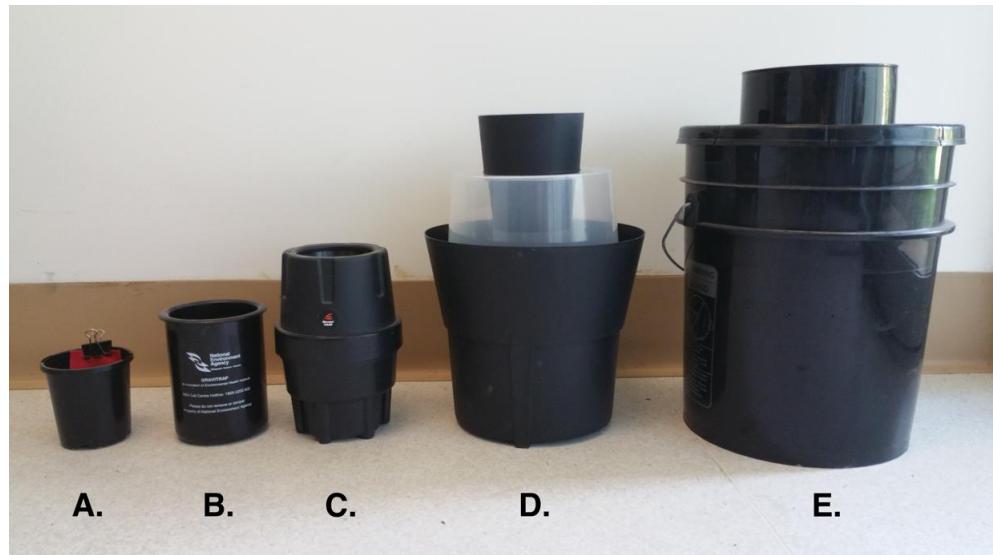


Figura 2 – (A) Standard Lethal Ovitrap (LO), (B) National Environmental Agency Singapore Sticky Ovitrap (SO), (C) MosquiTRAP. Sticky Ovitrap (SO), (D) Biogents Gravid Aedes Trap (GAT), (E) Centers for Disease Control (CDC) Autocidal Gravid Ovitrap(AGO)

FONTE: (JOHNSON et al., 2017)

Embora as armadilhas sejam diferentes no design, tanto o AGO (E) quanto o GAT (D) alcançaram o efeito desejado superando os demais modelos apresentados em atratividade para o Aedes. Estudos em Porto Rico demonstraram que a AGO capturou mais fêmeas grávidas e forneceu maior sensibilidade do que as ovitraps convencionais, enquanto em testes no norte da Austrália, os GATs coletaram de 2 a 4 vezes mais Aedes fêmeas que duas variações de ovitraps, o MosquiTRAP e o ovitrap pegajoso duplo (JOHNSON et al., 2017).

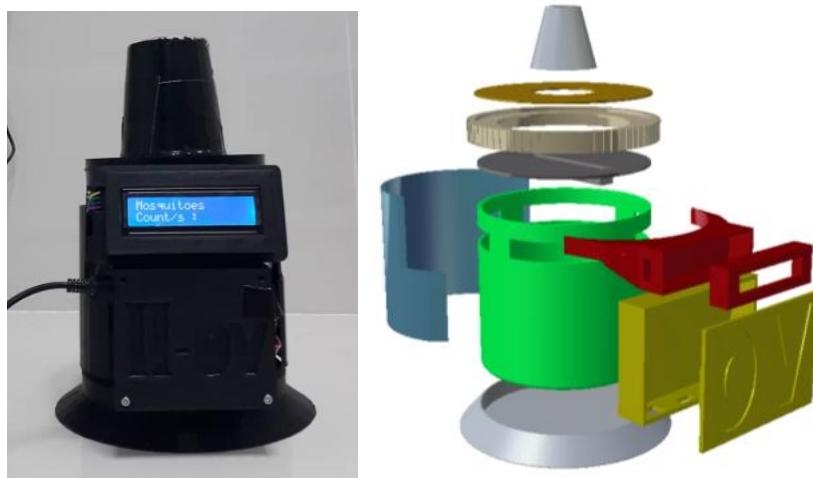


Figura 3 – Modelo de ovitrap desenvolvido por Ismaliza Isa et al. .

FONTE: (ISA et al., 2019)

O modelo desenvolvido por Ismaliza e outros é uma ovitrap do tipo sticker, além de capturar o mosquito, a armadilha possui um sensor (VL53L0X) em seu interior que conta o número de mosquitos que passaram pela armadilha, sendo eles capturados ou não por ela. Este número é mostrado no display da armadilha e enviado para uma aplicação web que pode ser acessada através de um navegador em qualquer dispositivo conectado a internet.

Diante do exposto, este trabalho busca desenvolver uma solução para o controle da população do mosquito Aedes, que seja mais rápida e eficaz que o método tradicional de controle, utilizando para isso recursos de tecnologia como sensores e aplicações web.

### 3 DESENVOLVIMENTO

#### 3.1 A ideia inicial

A ideia inicial deste trabalho era desenvolver uma armadilha capaz de capturar e também fotografar os mosquitos, para depois aplicar processamento digital de imagens para catalogar as espécies de mosquitos fotografadas. Também havia uma câmera submersa na água da armadilha para que fosse possível fotografar as larvas do mosquito.

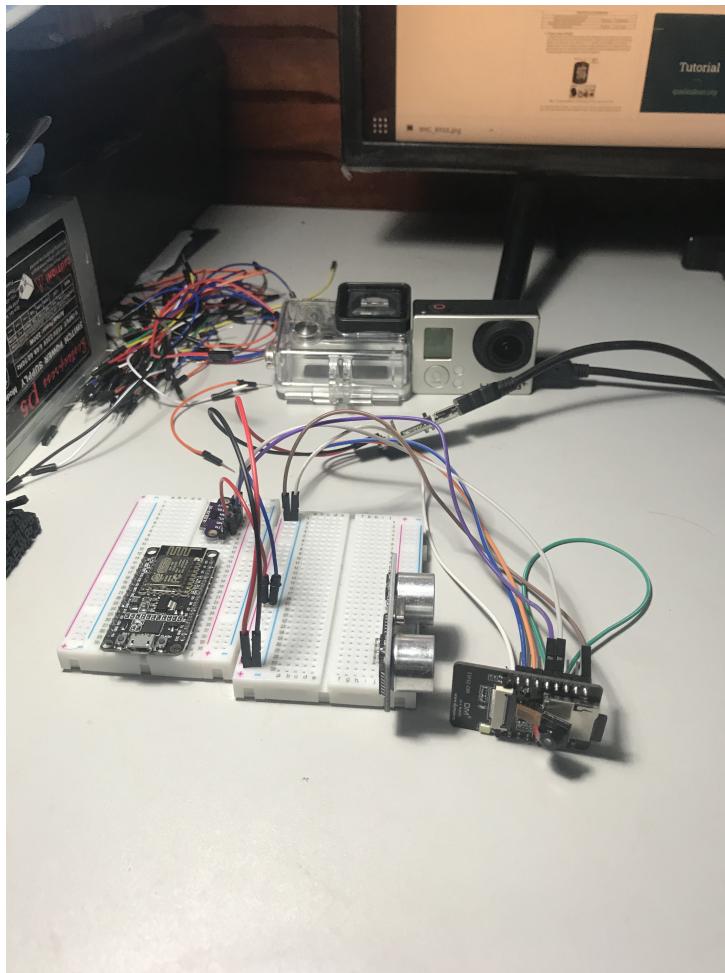


Figura 4 – Primeiros testes, módulo NodeMCU, um sensor laser (o mesmo utilizado no protótipo final),, um sensor ultrassônico, um módulo ESP32-CAM e uma câmera GoPro

O primeiro teste consistia em ter duas câmeras, uma para fotografar os mosquitos (ESP32-CAM), e uma para as larvas (GoPro). A câmera GoPro possui um web server integrado onde é possível via requisições HTTP, tirar fotos, gravar vídeos e controlar várias outras funções da câmera. O ESP32-CAM também possui wi-fi integrado. Então, neste primeiro modelo, a ESP32-CAM era ativada pelo sensor laser e a GoPro era ativada a cada 1 hora, destaca-se que a GoPro ficaria submersa e seria responsável por fotografar o desenvolvimento das larvas do mosquito.

Uma das dificuldades encontradas neste primeiro modelo era que além de se conectar com o web server da GoPro, o módulo ESP32-CAM também se conectava com uma rede wi-fi externa para

envio das fotos via requisição HTTP para API. O tempo de envio das fotos da GoPro para API era de aproximadamente 30 segundos, já o tempo do ESP32-CAM era em torno de 3 segundos. Outro ponto era que essas fotos ficavam armazenadas em um cartão Micro-SD no módulo ESP32-CAM antes de serem enviadas para API.

A necessidade de se conectar em duas redes wi-fi diferentes, ter que usar um cartão Micro-SD, além do custo do projeto ficar acima de R\$2000, fez com que fosse desenvolvido um segundo teste sem a GoPro.

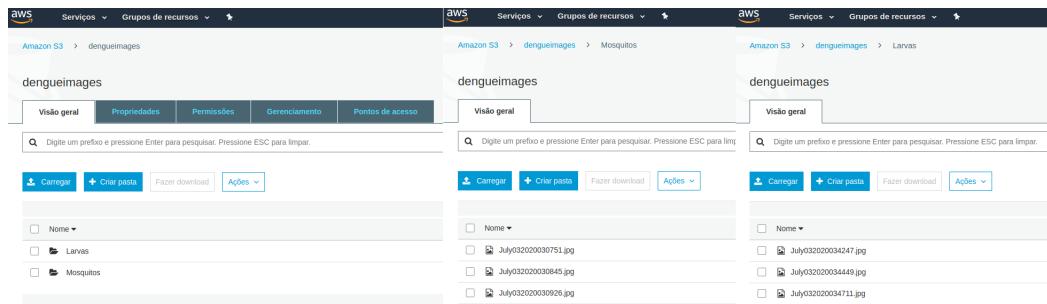


Figura 5 – Estrutura de pastas de fotos para larvas e mosquitos no Amazon S3.

Neste segundo teste havia apenas a câmera ESP32-CAM fixada ao teto da armadilha, ainda utilizando o cartão Micro-SD. O fato de ter apenas uma câmera, eliminou a necessidade de se conectar em duas redes diferentes, porém a resolução da câmera ESP32-CAM é de no máximo 1600 x 1200 pixels o que torna praticamente impossível detectar um mosquito na foto registrada, fator esse que levou a um terceiro teste que deu origem ao protótipo apresentado a seguir.



Figura 6 – Foto registrada pelo ESP32-CAM.

### 3.2 Ovitrap

Um dos grandes desafios ao se desenvolver uma ovitrap é torná-la atraente para as fêmeas do mosquito, isto é, fazer com que a fêmea do mosquito deposite seus ovos na ovitrap e não em outro lugar próximo a ela. Testes feitos por (HOEL et al., 2011) e (PAZ-SOLDAN et al., 2016) constataram que os mosquitos fêmeas são atraídos por ovitraps totalmente pretas.

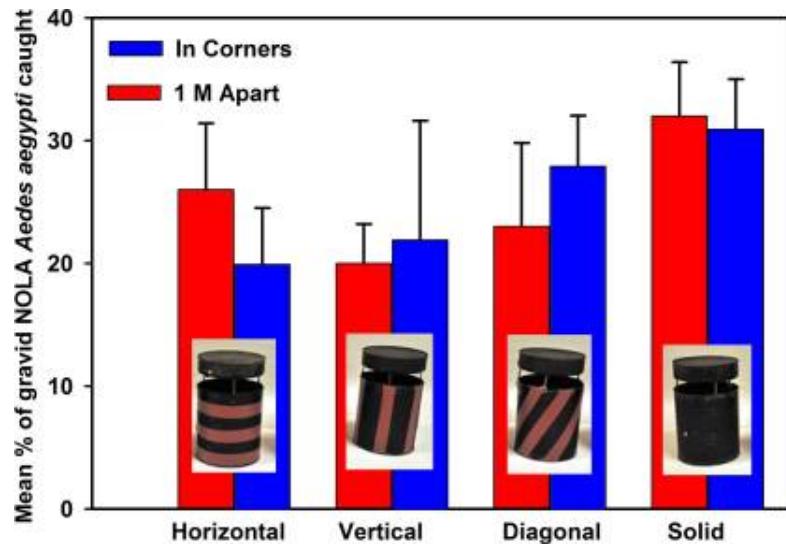


Figura 7 – Padrões em ovitraps.

FONTE: (PAZ-SOLDAN et al., 2016)

Outro ponto relevante ao se projetar uma ovitrap é o seu tamanho, segundo (JOHNSON et al., 2017) ovitraps de tamanho grande são mais atrativas as fêmeas do mosquito que as ovitraps pequenas além de demandarem manutenções menos frequentes, uma vez que as ovitraps pequenas têm taxas de evaporação de água muito maiores que as ovitraps maiores.



Figura 8 – Protótipo de ovitrap desenvolvido neste trabalho (sem sensores).

Levando em conta os pontos levantados anteriormente, foi desenvolvido uma ovitrap atrativa ou seja, grande, com a maior parte preta e que também demandasse manutenções com pouca frequência.

O protótipo foi influenciado pelo modelo Gravid Aedes Trap (GAT) (EIRAS et al., 2014), a parte transparente tem a intenção de confundir o mosquito, que se não for capturado pela cola nas paredes pretas da abertura, ao tentar sair da armadilha, orientado pela luz, ficará preso no interior da armadilha. É possível também passar um inseticida sob a parte transparente para antecipar a morte do mosquito.

Para o desenvolvimento do protótipo foram utilizados os seguintes materiais:

### 3.2.1 Vaso maior

Vaso de plástico utilizado em jardinagem.



Tabela 1 – Dimensões vaso maior.

Descrição	Valor
Diâmetro abertura	30 cm
Diâmetro externo	36 cm
Diâmetro inferior	20 cm
Altura	30 cm

Figura 9 – Vaso maior.

### 3.2.2 Vaso menor

Vaso de plástico utilizado em jardinagem.



Tabela 2 – Dimensões vaso menor.

Descrição	Valor
Diâmetro abertura	14,5 cm
Diâmetro externo	17,5 cm
Diâmetro inferior	10,5 cm
Altura	14 cm

Figura 10 – Vaso menor.

### 3.2.3 Prato

Prato de plástico utilizado embaixo de vasos de jardinagem.



Tabela 3 – Dimensões prato.

Descrição	Valor
Diâmetro abertura	16,5 cm
Diâmetro superior	17,5 cm
Diâmetro inferior	15,5 cm
Altura	2,5 cm

Figura 11 – Prato.

### 3.2.4 Boleira

Boleira de plástico utilizada para guardar bolos.



Figura 12 – Boleira.

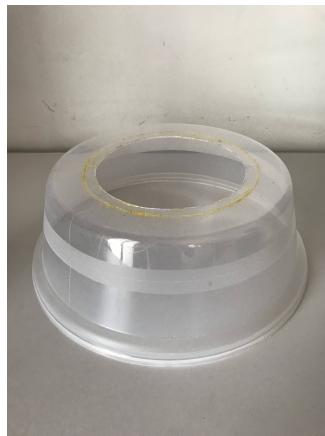


Figura 13 – Tampa da boleira  
após o recorte.

Tabela 4 – Dimensões boleira.

Descrição	Valor
Diâmetro recorte	14,5 cm
Diâmetro superior	20 cm
Diâmetro inferior	30 cm
Altura	11 cm

### 3.2.5 Tela mosquiteira

A tela mosquiteira foi recortada em formato de circulo com 30 cm de diâmetro e fixada com velcro as paredes internas do vaso maior.



Figura 14 – Tela mosquiteira.



Figura 15 – Posição final da tela mosquiteira.

### 3.2.6 Breu



Figura 16 – Breu.

Breu é uma resina natural utilizada para afinar violinos e envernizar móveis, pode ser encontrado em casas de materiais agrícolas. No protótipo da armadilha ele foi utilizado nas paredes internas da abertura da armadilha como cola para captura do mosquito. Para o preparo da cola, basta misturar 100g de breu com 50ml de óleo de soja e aquecer a mistura em um fogão em fogo baixo por aproximadamente 25 minutos (Epagri Vídeos, 2017).

### 3.2.7 Palitos de churrasco e fitas hellerman



Figura 17 – Abertura com prato superior.

Foram utilizados palitos de churrasco cortados em pedaços de 10 cm cada, para sustentar o prato superior da armadilha, e também 3 fitas hellerman para sustentar o conjunto de módulos e sensores ao prato superior.

## 3.3 Hardware

Para que fosse possível contar os mosquitos que adentrassem na armadilha foram colocados peças de hardware na mesma. Foram testados alguns módulos e sensores como o NodeMCU e o

sensor ultrassônico, porém devido a baixa precisão do sensor ultrassônico ele foi substituído por um sensor laser de maior precisão.



Figura 18 – Sensores conectados e fixos ao prato da armadilha.

Na Figura 18 vemos o módulo NodeMCU conectado a uma protoboard branca que o conecta a 5 sensores laser fixados nas extremidades do prato. Foram utilizados um campo micro USB e uma fonte de 5V para alimentação do módulo.

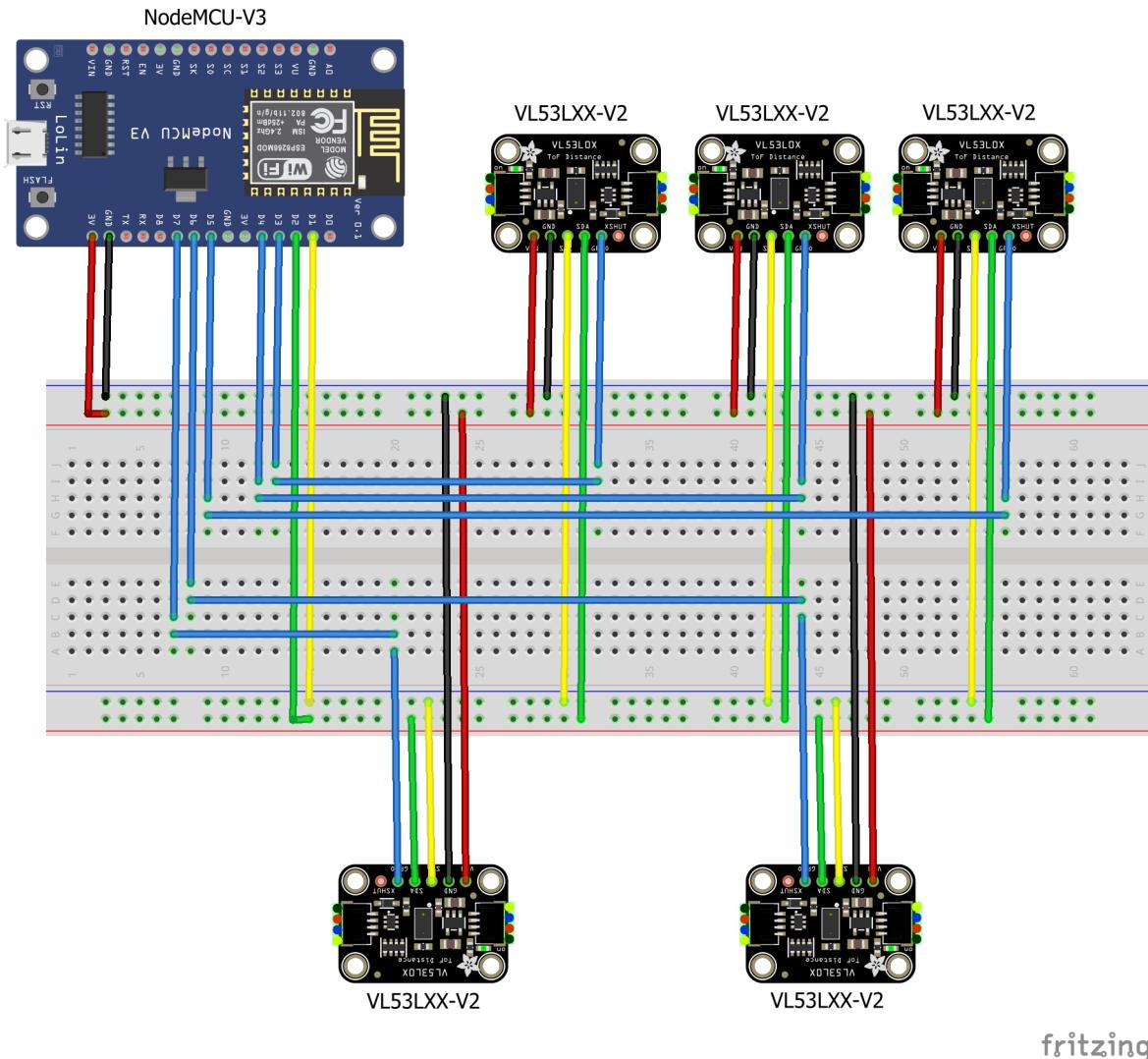


Figura 19 – Esquema de ligação dos sensores.

No esquema é possível ver detalhadamente as ligações entre o módulo e os sensores, note que os ponto verdes na protoboard representam linhas energizadas. A programação do módulo NodeMCU foi feita em C, utilizando a extensão PlatformIO (Platformio, 2021) do VSCode (VSCode, 2021).

Os módulos e sensores utilizados no protótipo finalv da armadilha são:

### 3.3.1 NodeMCU

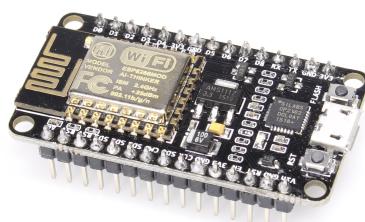


Figura 20 – NodeMCU.

O NodeMCU é uma placa de desenvolvimento com módulo wi-fi integrado, compatível com as linguagens de programação: Lua, Python, JavaScript e IDE do Arduino.

### 3.3.2 VL53L0X (Time-of-Flight)



Figura 21 – VL53L0X-V2 (Time-of-Flight).

O VL53L0X-V2 é um sensor a laser de tempo de vôo, ele é responsável por detectar quando o mosquito entra na armadilha e mandar as informações para o NodeMCU.

## 3.4 Software

Para receber e disponibilizar os dados provenientes das armadilhas foi desenvolvido uma API em NodeJS (NODEJS, 2021) utilizando o framework Express (EXPRESS, 2021). Para guardar os dados foi utilizado o banco de dados PostgreSQL (POSTGRESQL, 2021). Para consumir os dados disponibilizados pela API foi desenvolvido uma aplicação em ReactJS (REACT, 2021).

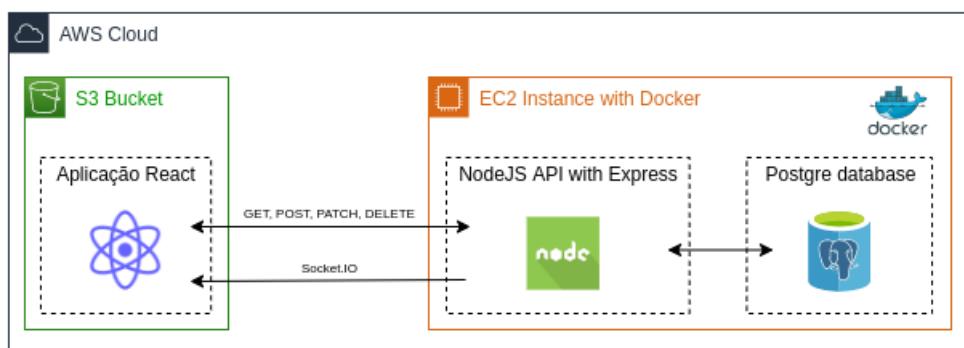


Figura 22 – Diagrama de infraestrutura.

A Figura 22 mostra a infraestrutura do projeto na nuvem utilizando os serviços do AWS da Amazon. O front-end está hospedado utilizando o serviço de hospedagem de site estático Amazon S3 (AWS, 2021). A API bem como o banco de dados estão rodando em containers Docker (DOCKER, 2021) dentro de uma instância do Amazon EC2 (Just me and Opensource, 2020). A comunicação entre a aplicação cliente e a API é feita utilizando o protocolo HTTP e web socket.

### 3.4.1 Banco de dados

Para a tarefa de armazenar os dados provenientes das armadilhas, das empresas/prefeituras e dos usuários da plataforma, foi escolhido o banco relacional PostgreSQL (POSTGRESQL, 2021). A

escolha do banco se deu pela familiaridade do autor com o mesmo e uma vez que os dados do sensor são enviados somente quando há uma captura, não houve necessidade de utilização de um banco de dados não relacional. A Figura 23 mostra o diagrama de tabelas do banco.

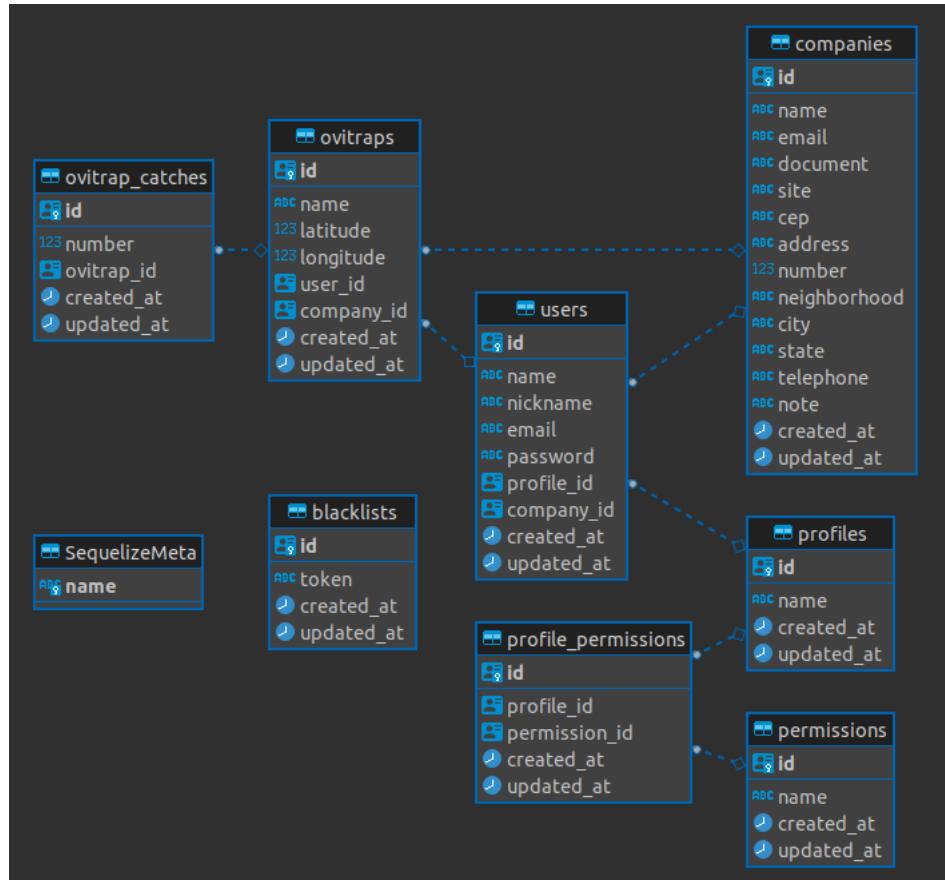


Figura 23 – Diagrama do banco.

### 3.4.2 API

A API para consumo e disponibilidade dos dados armazenados no banco, foi feita em NodeJS (NODEJS, 2021) utilizando o framework Express (EXPRESS, 2021). Todas as rotas (exceto login e "esqueci minha senha") são protegidas por token, com 8h de duração. Todas as rotas contam com validação dos campos recebidos.

Além de uma porta destinada a requisições HTTP, há outra porta destinada a comunicação via web socket utilizando Socket.io (SocketIo, 2021), possibilitando que a aplicação cliente receba dados em tempo real, como por exemplo, se houver uma captura de mosquito.

### 3.4.3 Aplicação Cliente

A aplicação cliente foi desenvolvida utilizando ReactJS (REACT, 2021), também foram utilizadas bibliotecas como redux-saga (REDUX-SAGA, 2021) para chamadas da API, MUI (MUI, 2021) para estilização de botões e painéis, Apexcharts (APEXCHARTS, 2021) para exibição do gráfico,

dentre outras. A aplicação é responsiva e conta com dois temas, claro e escuro, dois idiomas, português e inglês e pode ser acessada em qualquer navegador web.

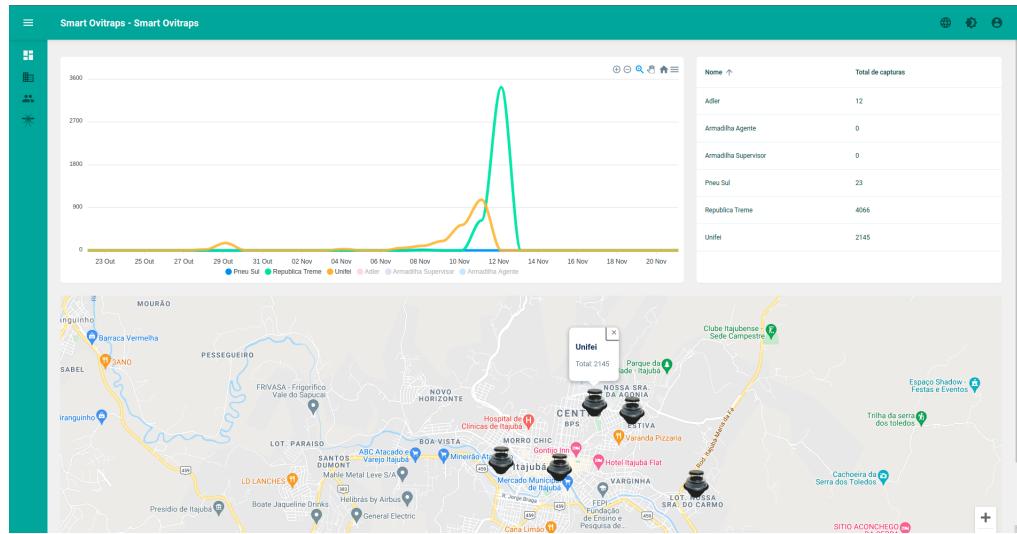


Figura 24 – Tema claro.

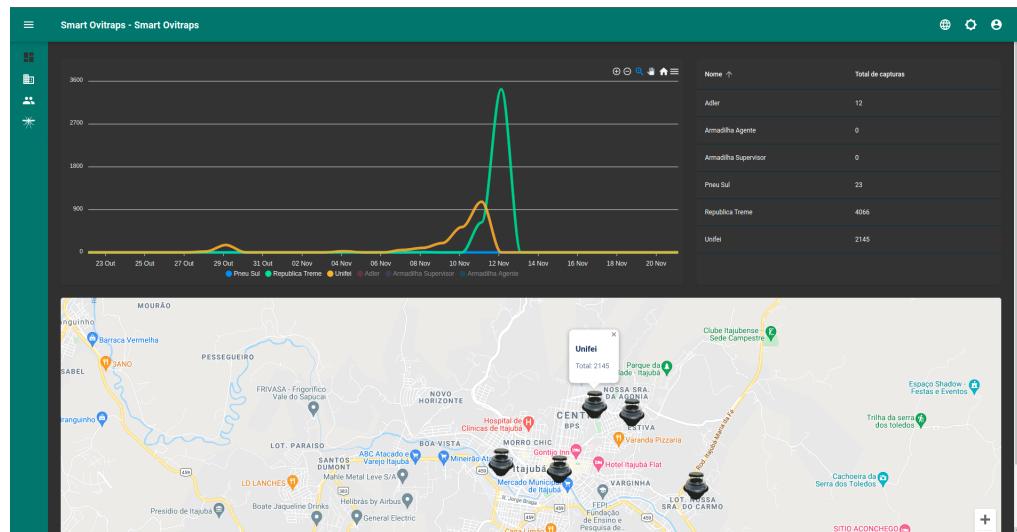


Figura 25 – Tema escuro.

No dashboard e no cadastro de armadilhas foi utilizado o serviço do Google Maps (API, 2021) para exibição das armadilhas conforme sua latitude e longitude. No mapa, ao clicar em cada armadilha, é mostrado o nome e a quantidade total de mosquitos capturados pela armadilha. Quando há uma captura em tempo real é mostrado o ícone de um mosquito acima da armadilha que realizou a captura.

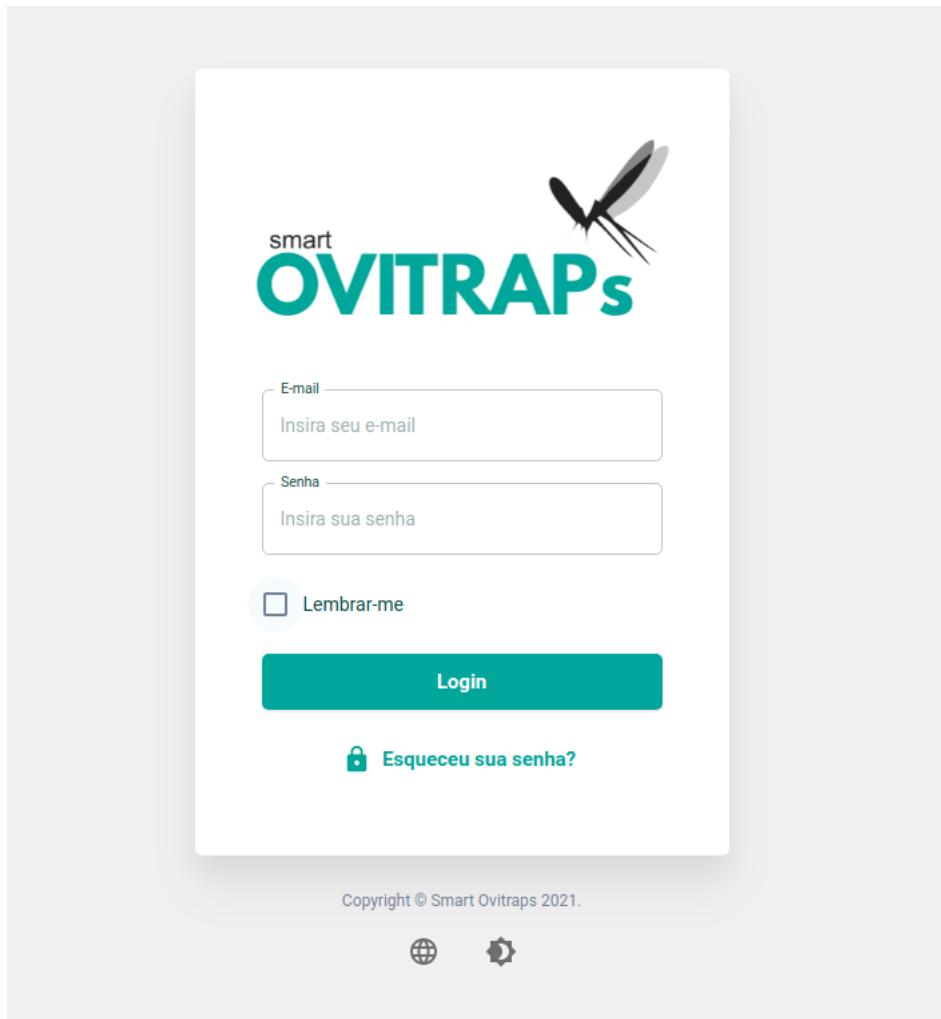


Figura 26 – Tela de login.

Para o envio de emails para o usuário no cadastro e na definição de uma nova senha foi utilizado o serviço de envio de emails SendGrid (SENDGRID, 2021).

Smart Ovitraps - Smart Ovitraps			
	Nome	Usuário	Empresa
<input type="checkbox"/>	Adler	Agente Adler	Armadilhas Adler
<input type="checkbox"/>	Armadilha Agente	Agente Itajubá	Prefeitura de Itajubá
<input type="checkbox"/>	Armadilha Supervisor	Supervisor Itajubá	Prefeitura de Itajubá
<input type="checkbox"/>	Pneu Sul	Agente Itajubá	Prefeitura de Itajubá
<input type="checkbox"/>	República Termo	Agente Daniel	Armadilhas Daniel

Figura 27 – Tela de listagem de armadilhas.

A plataforma permite cadastrar, listar, alterar e deletar: empresas, usuários e armadilhas. Existem 3 perfis de usuários: Administradores do Sistema, Gestores e Agentes. Administradores do Sistema têm acesso a todas as funcionalidades do sistema. Gestores têm acesso a todas as funcionalidades do sistema (exceto cadastrar, ler, alterar e deletar empresas/prefeituras). Agentes somente podem ver as armadilhas da sua própria empresa, cadastrar novas armadilhas, alterar e deletar suas próprias armadilhas.

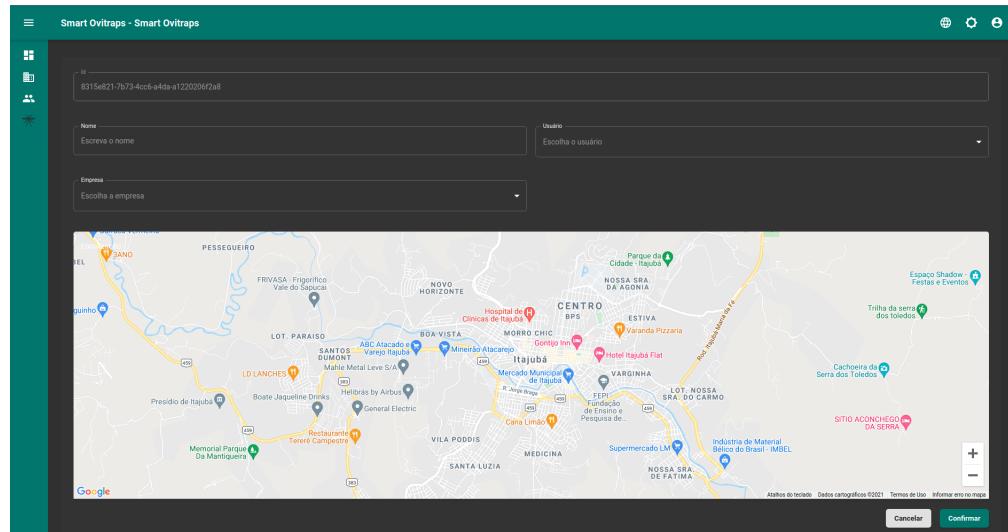


Figura 28 – Tela de cadastro de armadilhas.

## 4 VISÃO GERAL DO SISTEMA

Juntando protótipo, sistema embarcado e sistema hospedado em nuvem, temos o seguinte diagrama:

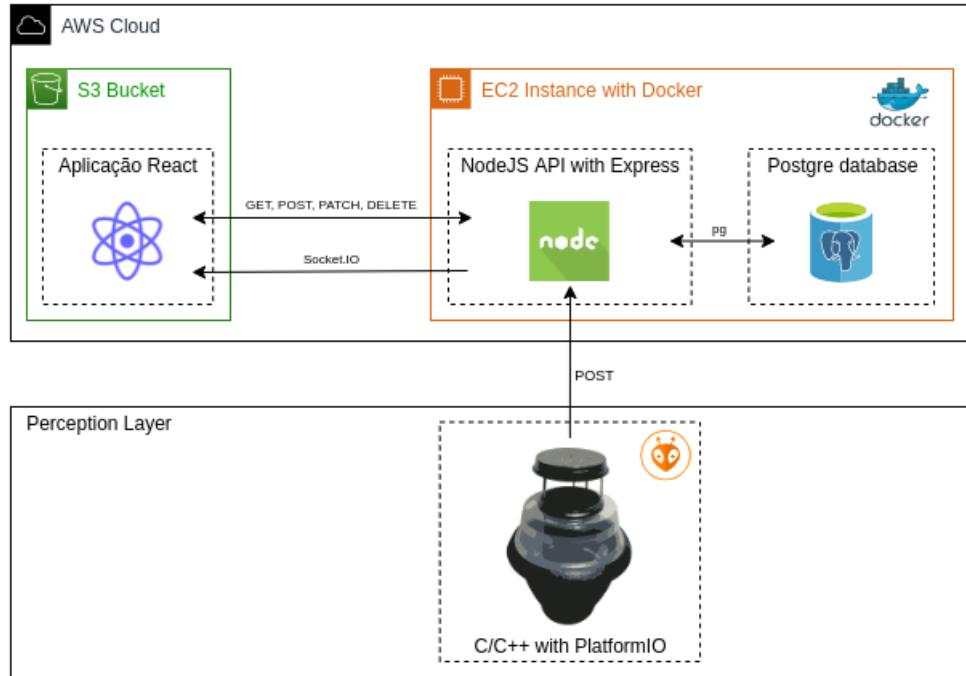


Figura 29 – Visão geral do sistema.

Nele temos uma visão completa do funcionamento do sistema, o registro de uma nova captura começa no sensor da armadilha, que ao medir uma distância menor que o intervalo configurado faz uma chamada HTTP POST para API. A API por sua vez insere a nova captura na tabela de capturas, e envia um payload via web socket para todos os usuários conectados daquela empresa. A aplicação web recebe o payload e exibe uma nova captura no mapa, no gráfico e na tabela.

O código de cada parte do sistema está disponível em <https://gitlab.com/users/danielreis58/projects> nos repositório: A Cloud IoT-Ovitrap System - ESP32, A Cloud IoT-Ovitrap System - API e A Cloud IoT-Ovitrap System - Dashboard.

### 4.1 Alimentação

A alimentação energética do módulo NodeMCU é feita através de um cabo micro USB ligado a uma fonte de 5V. Em locais onde não há rede elétrica não é possível utilizar este protótipo de armadilha. Uma solução para este locais seria a utilização de uma célula fotovoltaica ligada a uma bateria.

## 4.2 Comunicação

A comunicação do módulo com a nuvem utiliza a tecnologia de rede sem fio wi-fi, portanto é necessário que a armadilha esteja próxima a um roteador ou access point sem fio. Uma solução para locais onde não há disponibilidade de rede wi-fi seria a utilização de outra tecnologia de comunicação como GSM, LoRa, Sigfox ou até mesmo 5G.



Figura 30 – Protótipo final da armadilha em funcionamento.

## 5 CUSTOS

Para construção do protótipo foram utilizados materiais que podem ser adquiridos facilmente na internet, em lojas de materiais domésticos ou agrícolas. A Tabela 5 descreve todos os materiais e valores, cotados na data de publicação deste trabalho, na moeda brasileira real.

Tabela 5 – Tabela de custo.

Descrição	Quantidade por lote	Unidade de medida	Valor lote	Valor por unidade	Quantidade utilizada	Valor total
Vaso maior	1	un	8,99	8,99	1	8,99
Vaso menor	1	un	1,99	1,99	1	1,99
Boleira	1	un	9,99	9,99	1	9,99
Prato	1	un	1,99	1,99	1	1,99
Tela com velcro	1,95	$m^2$	12,99	6,66	0,066	0,44
Breu	100	g	7,7	0,08	100	7,70
Palito Churrasco	100	un	15	0,15	2	0,30
Fita herillerman 120 mm	100	un	4,99	0,05	3	0,15
Bastao cola quente	10	un	5,99	0,60	1	0,60
Jumpers	120	un	20,72	0,17	54	9,32
Protoboard	1	un	15	15,00	1	15,00
NodeMCU	1	un	25,5	25,50	1	25,50
V153X-V2	1	un	13,79	13,79	5	68,95
Cabo micro USB	1	un	4,9	4,90	1	4,90
Fonte de tomada USB 5v	1	un	10,23	10,23	1	10,23
					Total	<b>166,05</b>

Observamos a partir da tabela que o custo com componentes eletrônicos foi de R\$133,90 que representa 80,64% do valor total do protótipo e que o custo com componentes estruturais foi de R\$32,15 que representa 19,36% do valor total do protótipo.

## 6 VALIDAÇÃO E RESULTADOS

Foram realizados 3 testes de campo com o protótipo. Todos na cidade de Itajubá. O primeiro teste foi realizado de 31/07/2020 a 07/08/2020 no bairro N.Sra. da Agonia. O segundo teste foi realizado de 28/08/2020 a 04/09/2020 no bairro Medicina. O terceiro teste foi realizado de 01/12/2021 a 08/12/2021 no bairro Centro.

Tabela 6 – Tabela de resultados.

Data	Local	Capturas registradas	Capturas observadas
31/07/2020 a 07/08/2020	N.Sra. da Agonia	4685	0
28/08/2020 a 04/09/2020	Medicina	3986	0
01/12/2021 a 08/12/2021	Centro	6	0

Observando o resultado dos testes notamos que há uma discrepância entre o número de capturas dos dois primeiros testes e o número de capturas do terceiro teste. Isto é decorrente da forma como foram configurados os sensores. Nos dois primeiros testes os sensores estavam configurados para registrar uma nova captura caso a distância fosse menor que 100 mm, já no terceiro teste os sensores foram configurados para registrar uma nova captura caso a distância fosse maior do que 0 mm e menor que 100 mm, isto porque foi observado que ocasionalmente o sensor VL53L0X registra a distância 0 mm mesmo que não exista nenhum objeto em frente ao sensor, por isso foi adicionado a condição maior que 0 mm.

## 7 CONCLUSÃO E TRABALHOS FUTUROS

Baseado nos testes realizados com o protótipo em campo aberto, a maior dificuldade encontrada foi atrair mosquitos para armadilha. Houve alguns registros do sensor, porém nenhum mosquito foi capturado pela cola interna da armadilha. A cola desenvolvida com breu após alguns dias derreteu devido ao calor do sol e caiu na tela mosquiteira da armadilha. Após alguns meses a cola ressecou e perdeu sua aderência. Apesar do protótipo não ter sucesso em capturar mosquitos, todo o restante da aplicação se saiu bem, em testes com objetos maiores ou com chamadas HTTP que simulam uma armadilha real. A plataforma está consistente e pode ser replicada para qualquer outra rede de sensores com pouquíssimas modificações, como por exemplo uma rede de monitoramento de níveis de rios, quando o rio atingir determinado nível o sensor é ativado e envia uma requisição para API. A estrutura do banco está bem definida e todo o sistema de perfis e permissões pode ser customizado de acordo com as necessidades do cliente.

Como sugestão de trabalhos futuros temos os seguintes itens:

- Nível de protótipo
  - Validar o protótipo da armadilha, em ambiente controlado e em ambiente aberto. Analisar e comparar os resultados e a eficácia da mesma.
  - Analisar o uso de substâncias com cheiro atrativo para o mosquito, como por exemplo, a alfafa prensada.
  - Analisar a viabilidade de se trocar o sensor laser por um sensor sonoro, com aplicação de filtro para a frequência sonora de vôo do mosquito.
  - Analisar viabilidade de utilização de células fotovoltaicas juntamente com outra tecnologia de comunicação de baixo consumo energético como LoRa.
  - Procurar alternativas melhores para cola no interior da armadilha como: fitas pregá-rato ou fitas prega-mosca e/ou incluir uma ventoinha abaixo da abertura, para que o mosquito seja sugado para dentro da armadilha.
- Nível de aplicação
  - Cadastrar o id da armadilha, nome e senha da rede wi-fi de maneira dinâmica.
  - Melhorar a aplicação para que seja possível cadastrar uma foto de perfil para o usuário.

## REFERÊNCIAS

- ANDRIOLI, D. C.; BUSATO, M. A.; LUTINSKI, J. A. Spatial and temporal distribution of dengue in brazil, 1990 - 2017. *PLoS ONE*, Feb 2020.
- APEXCHARTS. *ApexCharts*. 2021. Disponível em: <<https://apexcharts.com/docs/installation/>>. Acesso em: 15/12/2021.
- API, R. G. M. *React Google Maps API*. 2021. Disponível em: <<https://www.npmjs.com/package/@react-google-maps/api>>. Acesso em: 15/12/2021.
- AWS. *Amazon S3*. 2021. Disponível em: <<https://aws.amazon.com/pt/s3/>>. Acesso em: 15/12/2021.
- BHATT, S. et al. The global distribution and burden of dengue. *Nature*, Apr 2013.
- DOCKER. *Deploying Docker containers on ECS*. 2021. Disponível em: <<https://docs.docker.com/cloud/ecs-integration/>>. Acesso em: 15/12/2021.
- EE, O.; KT, G.; DJ, G. Effects of beauveria bassiana on survival, blood-feeding success, and fecundity of aedes aegypti in laboratory and semi-field conditions. *Emerging infectious diseases*, v. 12, 2006.
- EIRAS, A. E.; BUHAGIAR, T. S.; RITCHIE, S. A. Efficacy of ovitrap colors and patterns for attracting aedes albopictus at suburban field sites in north-central florida. *Journal of Medical Entomology, Volume 51, Issue 1, 1 January, 2014*.
- Epagri Vídeos. *Armadilhas para insetos*. 2017. Disponível em: <[https://www.youtube.com/watch?v=my\\_9UzPPTUg&ab\\_channel=EpagriV%C3%ADdeos](https://www.youtube.com/watch?v=my_9UzPPTUg&ab_channel=EpagriV%C3%ADdeos)>. Acesso em: 15/12/2021.
- EXPRESS. *Express*. 2021. Disponível em: <<https://expressjs.com/pt-br/4x/api.html#express>>. Acesso em: 15/12/2021.
- G1. *Brasil teve aumento de 488 por cento nos casos de dengue em 2019*. 2020. Disponível em: <<https://g1.globo.com/ciencia-e-saude/noticia/2020/01/13/brasil-teve-aumento-de-488percent-nos-casos-de-dengue-em-2019.ghtml>>. Acesso em: 15/12/2021.
- HEALTH, B. M. of. *Ministério da Saúde lança campanha nacional de combate ao mosquito Aedes aegypti*. 2020. Disponível em: <<https://www.gov.br/pt-br/noticias/saude-e-vigilancia-sanitaria/2020/11/ministerio-da-saude-lanca-campanha-nacional-de-combate-ao-mosquito-aedes-aegypti>>. Acesso em: 15/12/2021.
- HOEL, D. F. et al. Efficacy of ovitrap colors and patterns for attracting aedes albopictus at suburban field sites in north-central florida. *Journal of the American Mosquito Control Association*, 2011.
- ISA, I. et al. An iot-based ovitrap system applied for aedes mosquito surveillance. *IJEAT*, Out 2019.
- JOHNSON, B. J.; RITCHIE, S. A.; FONSECA, D. M. The state of the art of lethal oviposition trap-based mass interventions for arboviral control. *Insects*, Jan 2017.

Just me and Opensource. [ Docker 4 ] Using docker-machine to create EC2 instance in AWS. 2020. Disponível em: <[https://www.youtube.com/watch?v=7MXG7bxrokk&ab\\_channel=JustmeandOpensource](https://www.youtube.com/watch?v=7MXG7bxrokk&ab_channel=JustmeandOpensource)>. Acesso em: 15/12/2021.

LOK et al. An autocidal ovitrap for the control and possible eradication of aedes aegypti. *Southeast Asian J. Trop. Med. Publ. Health*, p. 2669–2677, 1977.

MUI. MUI. 2021. Disponível em: <<https://mui.com/pt/getting-started/usage/>>. Acesso em: 15/12/2021.

NODEJS. NodeJS. 2021. Disponível em: <<https://nodejs.org/en/docs/>>. Acesso em: 15/12/2021.

PAZ-SOLDAN, V. A. et al. Design and testing of novel lethal ovitrap to reduce populations of aedes mosquitoes community-based participatory research between industry academia and communities in peru and thailand. *Plos One*, n. 0160386, Aug 2016.

Platformio. Platformio. 2021. Disponível em: <<https://docs.platformio.org/en/latest/>>. Acesso em: 15/12/2021.

POSTGRESQL. PostgreSQL. 2021. Disponível em: <<https://www.postgresql.org/docs/14/index.html>>. Acesso em: 15/12/2021.

REACT. React. 2021. Disponível em: <<https://pt-br.reactjs.org/docs/getting-started.html>>. Acesso em: 15/12/2021.

REDUX-SAGA. Redux-Saga. 2021. Disponível em: <<https://redux-saga.js.org/>>. Acesso em: 15/12/2021.

SENDGRID. SendGrid. 2021. Disponível em: <<https://docs.sendgrid.com/pt-br/>>. Acesso em: 15/12/2021.

SocketIo. SocketIo. 2021. Disponível em: <<https://socket.io/docs/v4/>>. Acesso em: 15/12/2021.

VSCODE. VSCode. 2021. Disponível em: <<https://code.visualstudio.com/>>. Acesso em: 15/12/2021.

APENDICE A - DOCUMENTO DE REQUISITOS

---

# **DOCUMENTO DE REQUISITOS ESPECIFICAÇÃO**

**PARA**

**SMART OVITRAPS**

**Version 2.0**

**Preparado por : Daniel Pinheiro dos Reis**

**27 de dezembro de 2021**

## INTRODUÇÃO

### **Proposta**

SMART OVITRAP é uma plataforma integrada com armadilhas ovitraps para monitoramento e controle de arbovírus como dengue, zika e chikungunya.

### **Público Alvo**

Prefeituras, órgãos governamentais, grandes empresas ou entusiastas interessados no combate e monitoramento de arbovírus.

### **Escopo de projeto**

SMART OVITRAP cria um espaço para que gestores e agentes de órgãos de saúde, de saneamento ou grandes empresas, possam fazer o controle e monitoramento das armadilhas ovitraps instaladas na sua região.

O sistema consiste de uma plataforma com 3 tipos de usuários Agentes, Gestores e Administradores do Sistema. Para uma empresa ou prefeitura utilizar a plataforma é preciso previamente solicitar o cadastro para os Administradores do Sistema. Depois de cadastrado, a empresa ou prefeitura, e um primeiro usuário gestor, o gestor poderá logar na plataforma com sua senha e se desejar cadastrar novos gestores ou novos agentes.

## DESCRIÇÃO GERAL

### Perspectiva do Produto

SMART OVITRAP substitui as armadilhas ovitraps convencionais, permitindo o monitoramento das mesmas de maneira remota, o único trabalho de campo necessário é a troca da água ou da fita adesiva, a contagem de mosquitos é acessada de maneira remota através de qualquer navegador de internet.

### Casos de Uso e Características do Sistema

SMART OVITRAP possui três tipos de usuários:

- Administradores do Sistema
- Gestores
- Agentes

SMART OVITRAP possui uma plataforma única com funcionalidades diferentes para cada tipo de usuário.

#### Gestor

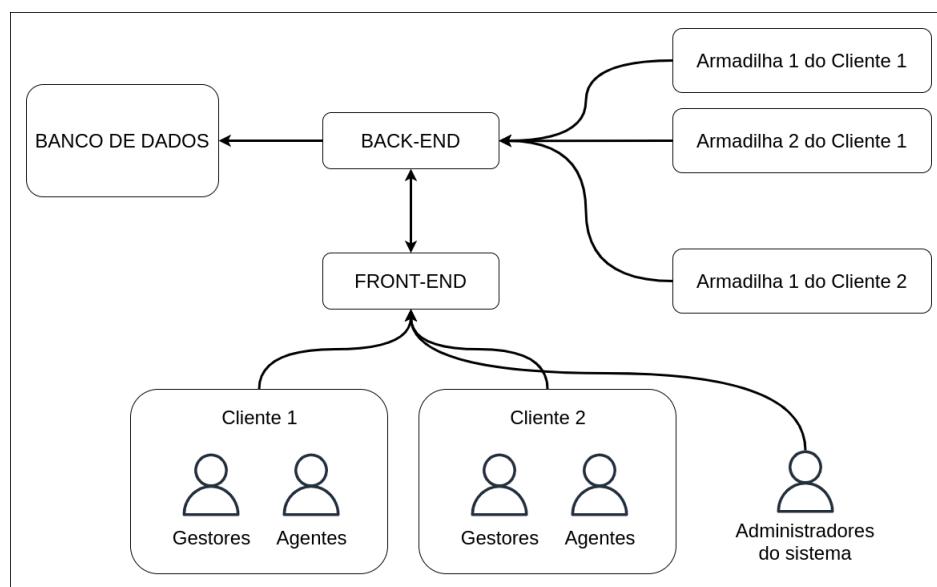


Figura 31 – Diagrama do sistema

Administradores do Sistema têm permissões para cadastrar: empresas, gestores, agentes e armadilhas. Gestores podem cadastrar: gestores, agentes e armadilhas. Agentes conseguem visualizar todas as armadilhas porém só podem editar e excluir suas próprias armadilhas.

Os Administradores de Sistema podem:

- Cadastrar, listar, alterar e deletar empresas ou prefeituras do sistema.
- Cadastrar, listar alterar e deletar usuários de determinada empresa ou prefeitura.
- Cadastrar, listar alterar e deletar armadilhas de determinada empresa ou prefeitura.

Os Gestores podem:

- Cadastrar, listar, alterar e deletar Gestores da sua empresa
- Cadastrar, lista, alterar e deletar Agentes da sua empresa
- Cadastrar, listar, alterar e deletar Armadilhas da sua empresa

Os Agentes podem:

- Cadastrar e listar suas próprias armadilhas
- Visualizar o total de capturas de cada armadilha da sua empresa

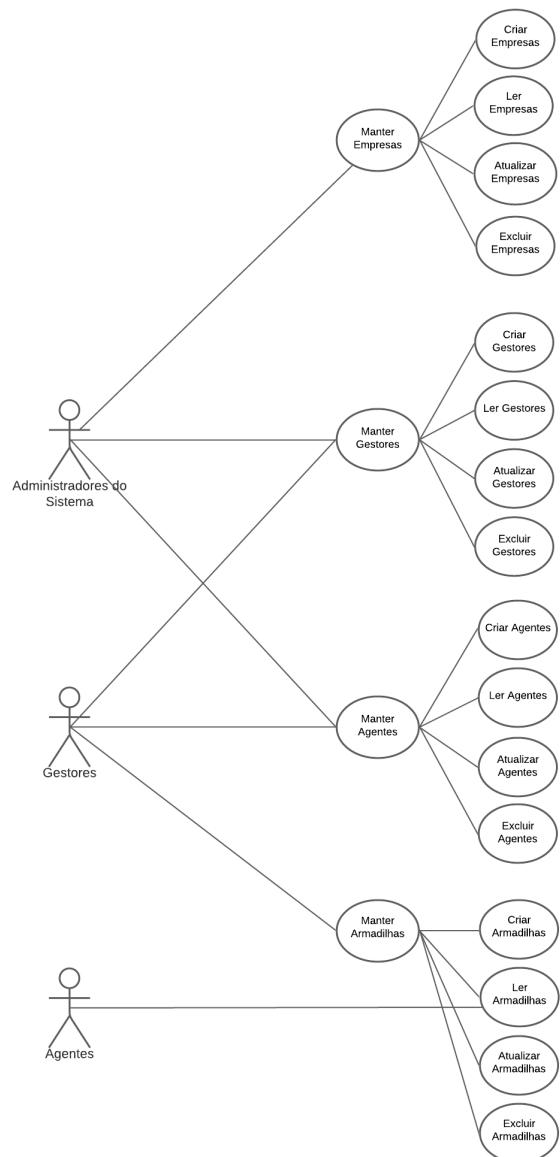


Figura 32 – Diagrama de Casos de Uso

### Funções do Produto

Permitir o monitoramento remoto de armadilhas ovitraps, quanto a quantidade de mosquitos, a localização da armadilha, a ultima manutenção realizada e outros dados que forem necessários para geração de relatórios e tomada de decisão de organizações interessadas no combate aos arbovírus

### Ambiente de Operação

A plataforma de monitoramento pode ser acessada de qualquer dispositivo conectado a internet e que possua um navegador de internet.

## **Design**

O design da plataforma seguirá as 10 heurísticas de Nielsen para garantir os padrões de usabilidade.

O desenvolvimento da plataforma seguirá rigorosamente este documento, os requisitos e a modelagem das telas aqui explicitados

## FUNCIONALIDADES DO SISTEMA

Esta seção descreve todos os requisitos e funcionalidades do sistema e as convenções e termos utilizados.

### **Descrição e Prioridade**

Esta seção descreve os usuários e requisitos da plataforma:

Os usuários são divididos em 3 grupos:

- Administradores do Sistema
- Gestores
- Agentes

Os requisitos de cada plataforma são divididos em dois grupos descritos abaixo:

- Requisito funcional (RF): Requisitos de extrema importância para o funcionamento da plataforma, estão diretamente ligados às funcionalidades do sistema.
- Requisito não-funcional (RNF): Requisitos ligados às métricas e atributos de qualidade do sistema.

Cada um dos requisitos acima possui uma das três prioridades abaixo:

- Essencial: Requisitos imprescindíveis para o funcionamento correto do sistema.
- Importante: Requisitos não essenciais para o funcionamento do sistema, mas que sem eles o sistema não se comporta de maneira satisfatória.
- Desejável: Requisitos que não interferem no funcionamento do sistema, complementos que auxiliam no funcionamento do sistema.

### **Requisitos Funcionais**

---

#### [RF01] Logar na plataforma

---

**Usuário:** Administradores do Sistema, Gestores e Agentes

O sistema deve permitir o login na plataforma.

Campo	Descrição	Tamanho	Métrica	Obrigatório
E-mail	E-mail do usuário	255	caracteres	sim
Senha	Senha do usuário	64	caracteres	sim

**Prioridade:**

- Essencial
- Importante
- Desejável

**[RF02] Cadastrar Administradores do Sistema****Usuário:** Administradores do Sistema

O sistema deve permitir o cadastro de usuários do tipo Administradores do Sistema

Campo	Descrição	Tamanho	Métrica	Obrigatório
Nome	Nome completo do usuário a ser cadastrado	255	caracteres	sim
E-mail	E-mail do usuário a ser cadastrado	255	caracteres	sim
Perfil	Perfil do usuário a ser cadastrado	64	caracteres	sim
Apelido	Apelido do usuário a ser cadastrado	64	caracteres	não

**Prioridade:**

- Essencial
- Importante
- Desejável

**[RF03] Cadastrar Empresas (Clientes)****Usuário:** Administradores do Sistema

O sistema deve permitir o cadastro de Empresas (Clientes)

Campo	Descrição	Tamanho	Métrica	Obrigatório
Nome	Nome completo da empresa a ser cadastrada	255	caracteres	sim
E-mail	E-mail da empresa a ser cadastrada	255	caracteres	sim
CPF/CNPJ	CPF/CNPJ da empresa a ser cadastrada	14/18	caracteres	sim
PF/PJ	Se o documento é pessoa física ou jurídica	2	caracteres	sim
Site	Site da empresa a ser cadastrada	255	caracteres	não
CEP	CEP da empresa a ser cadastrada	9	caracteres	não
Endereço	Endereço da empresa a ser cadastrada	255	caracteres	não
Número	Número da empresa a ser cadastrada	16	caracteres	não
Bairro	Bairro da empresa a ser cadastrada	64	caracteres	não
Cidade	Cidade da empresa a ser cadastrada	64	caracteres	não
Estado	Estado da empresa a ser cadastrada	64	caracteres	não
Telefone	Telefone da empresa a ser cadastrada	64	caracteres	não
Observações	Observações da empresa a ser cadastrada	255	caracteres	não

**Prioridade:**

- Essencial
- Importante
- Desejável

---

**[RF04] Cadastrar Usuários**

---

**Usuário:** Administradores do Sistema, Gestores e Agentes

O sistema deve permitir o cadastro de usuários do tipo administradores do sistema, gestores e agentes, sendo que os administradores do sistema somente podem ser cadastrados por administradores do sistema. Gestores podem cadastrar outros gestores e agentes.

Campo	Descrição	Tamanho	Métrica	Obrigatório
Nome	Nome completo do usuário a ser cadastrado	255	caracteres	sim
E-mail	E-mail do usuário a ser cadastrado	255	caracteres	sim
Perfil	Perfil do usuário a ser cadastrado: Gestor ou Agente	64	caracteres	sim
Apelido	Apelido do usuário a ser cadastrado	64	caracteres	não

**Prioridade:**

- Essencial

Importante

Desejável

### [RF05] Editar Empresa

**Usuário:** Administradores do Sistema

O sistema deve permitir ao cliente editar os dados da própria empresa

Campo	Descrição	Tamanho	Métrica	Obrigatório
Site	Site da empresa a ser cadastrada	255	caracteres	não
CEP	CEP da empresa a ser cadastrada	9	caracteres	não
Endereço	Endereço da empresa a ser cadastrada	255	caracteres	não
Número	Número da empresa a ser cadastrada	16	caracteres	não
Bairro	Bairro da empresa a ser cadastrada	64	caracteres	não
Cidade	Cidade da empresa a ser cadastrada	64	caracteres	não
Estado	Estado da empresa a ser cadastrada	64	caracteres	não
Telefone	Telefone da empresa a ser cadastrada	64	caracteres	não
Observações	Observações da empresa a ser cadastrada	255	caracteres	não

**Prioridade:**

Essencial

Importante

Desejável

### [RF06] Cadastrar Armadilhas

**Usuário:** Administradores do Sistema, Gestores e Agentes

O sistema deve permitir o cadastro de armadilhas

Campo	Descrição	Tamanho	Métrica	Obrigatório
Nome	Nome para armadilha Ex: Quintal, Jardim	255	caracteres	sim
UUID	UUID que deve ser inserido na armadilha	255	caracteres	sim

**Prioridade:**

Essencial

Importante

Desejável

**[RF07] Obter quantidade de mosquitos da armadilha**

**Usuário:** Administradores do Sistema, Gestores e Agentes

O sistema deve fornecer a quantidade de mosquitos por armadilha

Campo	Descrição	Tamanho	Métrica	Obrigatório
Quantidade	Quantidade de mosquitos	-	inteiro	-

**Prioridade:**

Essencial

Importante

Desejável

**[RF08] Obter a localização da armadilha**

**Usuário:** Administradores do Sistema, Gestores e Agentes

O sistema deve fornecer a localização da armadilha

Campo	Descrição	Tamanho	Métrica	Obrigatório
Latitude	Latitude da armadilha	-	flutuante	-
Longitude	Longitude da armadilha	-	flutuante	-

**Prioridade:**

Essencial

Importante

Desejável

## Não Funcionais

---

### [RNF01] Preenchimento automático do formulário

---

#### **Usuário:** Administradores do Sistema

No formulario de empresa [RF03], ao digitar o CEP os campos: Endereço, Bairro, Cidade e Estado devem ser preenchidos automaticamente

---

### [RNF02] Exibição do número de mosquitos em gráfico de linha

---

#### **Usuário:** Agentes e Gestores

Exibir o número de mosquitos dos últimos 7 dias, em um gráfico de linhas, sendo o eixo y: o número de mosquitos, e o eixo x: o período. É desejável que o gráfico filtre por armadilhas e por períodos de tempo de: Todos, 1 ano, 6 meses, 1 mês, 7 dias, 1 dia. sendo que quando escolhido 1 dia o período mostrado no eixo x deve ser em horas.

---

### [RNF03] Exibição da localização das armadilhas em mapa

---

#### **Usuário:** Agentes e Gestores

Exibir a localização das armadilhas no mapa usando API do Google Maps. É desejável que mostre um mapa de calor com base no número de mosquitos .

## Requisitos de Performance

A navegação entre as telas da plataforma bem como as conexões com o banco devem ocorrer de maneira fluida, sem gargalos ou travamentos que impliquem em uma baixa usabilidade do usuário final.

**Requisitos de Segurança**

Somente usuários cadastrados poderão acessar a plataforma, todas as senhas e dados sensíveis serão persistidos na base de dados utilizando métodos de criptografia confiáveis. As funcionalidades da plataforma devem respeitar o tipo de perfil do usuário não podendo por exemplo um usuário com perfil gestor cadastrar uma nova empresa.

**Atributos de Qualidade**

Para que a qualidade do software seja mantida e todos os requisitos sejam atendidos, serão realizados teste durante a fase de desenvolvimento do software e após entrega de sua primeira versão.

## **OUTROS REQUISITOS**

Por se tratar de um produto de software, a plataforma carece de manutenções para acompanhar o desenvolvimento de novas tecnologias. Os desenvolvedores do projeto se comprometem em fornecer manutenções periódicas para que todos os requisitos deste documento sejam atendidos de maneira vitalícia.