



**Universidade Federal de Itajubá**  
IMC - Instituto de Matemática e Computação  
Curso de Ciência da Computação  
**Smart OvitrapS**

*A Cloud IoT-Ovitrap System*

Daniel Pinheiro dos Reis

Orientador: Prof. Dr. Adler Diniz de Souza  
Co-Orientadora: Profa. Dra. Elisa Rodrigues

Itajubá, Novembro / 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 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 Ovitrap, 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 dos dados. Para isso aplica modelos de IoT no desenvolvimento de uma ovitrap "inteligente" capaz de capturar os dados e disponibilizá-los na nuvem para que possam ser acessados de maneira remota em qualquer lugar com acesso à Internet.

**Palavras-chave:** Armadilha, Ovitrap, IoT, Dengue, Zika, 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, IoT models are applied 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, IoT, Dengue, Zika, Chikungunya

## **LISTA DE ILUSTRAÇÕES**

## **LISTA DE TABELAS**

## **LISTA DE ABREVIATURAS E SIGLAS**

IoT	Internet of Things
HTTP	HyperText Transfer Protocol
AWS	Amazon Web Services
S3	Amazon Simple Storage Service
EC2	Amazon Elastic Compute Cloud
MUI	Material-UI Artificiais

## **SUMÁRIO**

## 1 INTRODUÇÃO

A intensificação dos arbovírus (dengue, zika, chikungunya, etc) 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 (??). Uma ferramenta de controle que vem sendo utilizada desde a década de 70 são as ovitraps (??)

### 1.1 O que é uma ovitrap?

Ovitraps são armadilhas desenvolvidas para capturar larvas e ou mosquitos. A primeira ovitrap que se tem informação (Figura 1) é creditada a Loki 1977 (??). A ovitrap de Loki 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 fêmea do mosquito que deposita seus ovos na armadilha não seja morta, os filhotes que eclodem dos ovos ficam presos pela malha trançada de morrem afogados.

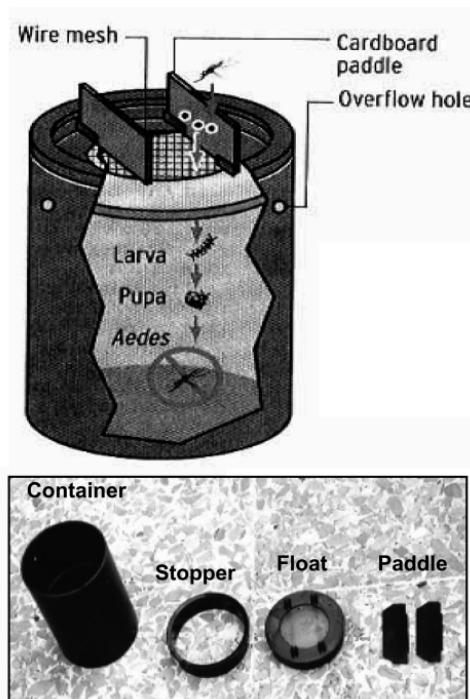


Figura 1 – Primeiro modelo de ovitrap, desenvolvido na década de 70

Disponível em: <<https://www.appropedia.org/Ovitrap>>. Acesso em: 22 de novembro de 2021

### 1.2 Evolução das ovitraps

Na evolução das ovitraps foram criadas 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 (??). Depois foi desenvolvido um modelo que ao invés de uma fita com inseticida utilizava uma fita adesiva que capturava a fêmea do mosquito (??). 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 atrativa as fêmeas do mosquito (??). Foi desenvolvido então, modelos maiores, mais atrativos as fêmeas do mosquito e que demandavam manutenções em períodos de tempo maiores.

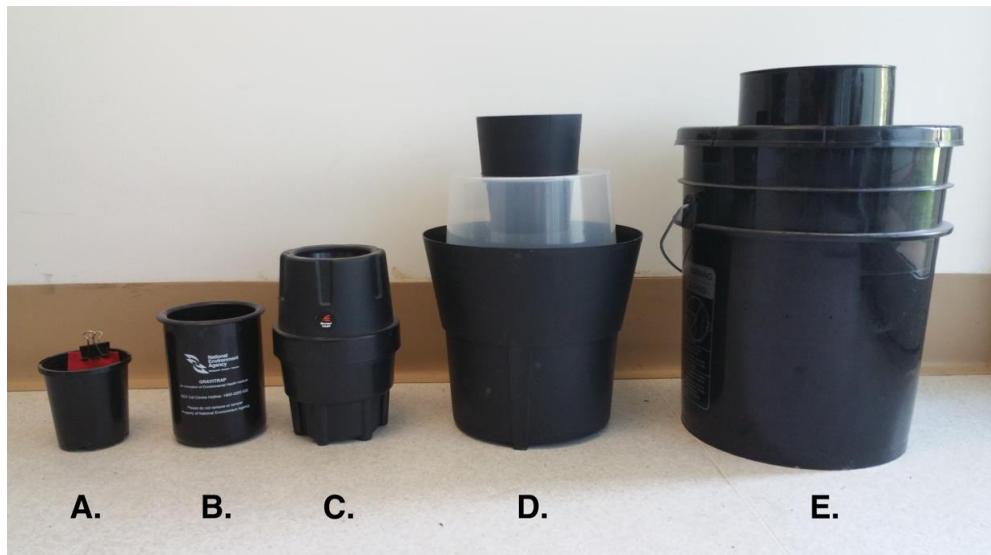


Figura 2 – (A) Standard Lethal Ovitrap (LO), (B) National Environmental Agency Singapore Sticky Ovitrap (SO), (C) MosquiTRAP Sticky Ovitrap (SO), (D) Biogents Gravid Aedes Trap (GAT), (E) Centers for Disease Control (CDC) Autocidal Gravid Ovitrap(AGO)

Disponível em: <[https://www.researchgate.net/publication/312158747\\_The\\_State\\_of\\_the\\_Art\\_of\\_Lethal\\_Oviposition\\_Trap-Based\\_Mass\\_Interventions\\_for\\_Arboviral\\_Control/figures](https://www.researchgate.net/publication/312158747_The_State_of_the_Art_of_Lethal_Oviposition_Trap-Based_Mass_Interventions_for_Arboviral_Control/figures)>. Acesso em: 22 de novembro de 2021

Embora as armadilhas sejam diferentes no design, tanto o AGO (E) quanto o GAT (D) alcançaram o efeito desejado superando as ovitraps padrão 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 do sexo feminino que duas variações de ovitraps, o MosquiTRAP e o ovitrap pegajoso duplo.

### 1.3 Motivação

Apesar da variação de modelos, todos eles precisam de um agente que faça o trabalho de campo de coletar os dados e prestar manutenção nas armadilhas. Em uma pesquisa bibliográfica encontrou-se apenas um modelo que emprega o conceito de IoT em ovitrap, o modelo desenvolvido por (??) (Figura 3)

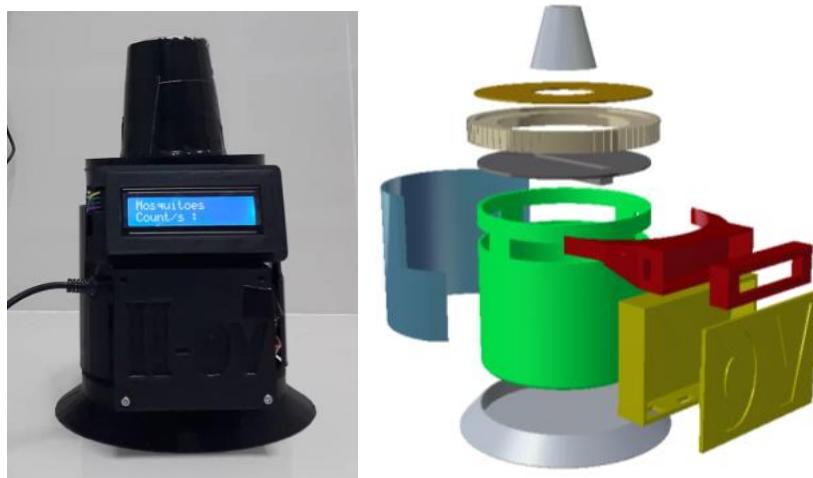


Figura 3 – Modelo de ovitrap desenvolvido por Ismaliza Isa et al.

Disponível em: <[https://www.researchgate.net/publication/337160411\\_An\\_IoT-Based\\_Ovitrap\\_System\\_Applied\\_for\\_Aedes\\_Mosquito\\_Surveillance/figures](https://www.researchgate.net/publication/337160411_An_IoT-Based_Ovitrap_System_Applied_for_Aedes_Mosquito_Surveillance/figures)>. Acesso em: 22 de novembro de 2021

O modelo desenvolvido por ISMALIZA ISA et al é uma ovitrap sticker, além de capturar o mosquito, a armadilha possui um sensor (VL53L0X) em seu interior que conta a quantidade 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 à 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.

#### 1.4 Justificativa

O método tradicional de controle, exige trabalho de campo para coletar e analisar os dados das ovitraps. 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, bem como a localização de cada armadilha é mostrada em um mapa, facilitando a análise e posterior tomada de decisão.

## 2 DESENVOLVIMENTO

Nesta seção busca-se descrever as etapas de desenvolvimento da solução proposta. O desenvolvimento se dividiu em três frentes:

- Ovitrap:
  - Desenvolver uma ovitrap que atraía e capture mosquitos fêmea.
- Hardware:
  - Contar quantos mosquitos passaram pela armadilha.
  - Enviar dados coletados com respectiva geolocalização da armadilha de maneira segura para uma API.
- Software:
  - Desenvolver uma API para receber dados de múltiplas armadilhas trata-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.

### 2.1 Ovitrap

Um dos grandes desafios ao se desenvolver uma ovitrap é torna-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 (??) e (??) constataram que os mosquitos fêmeas são atraídos por ovitraps totalmente pretas. Outra forma de atrair as fêmeas do mosquito além da própria água, é utilizar substâncias com cheiro atrativo para o mosquito, como por exemplo a alfafa prensada, 10mg por litro de água (??).

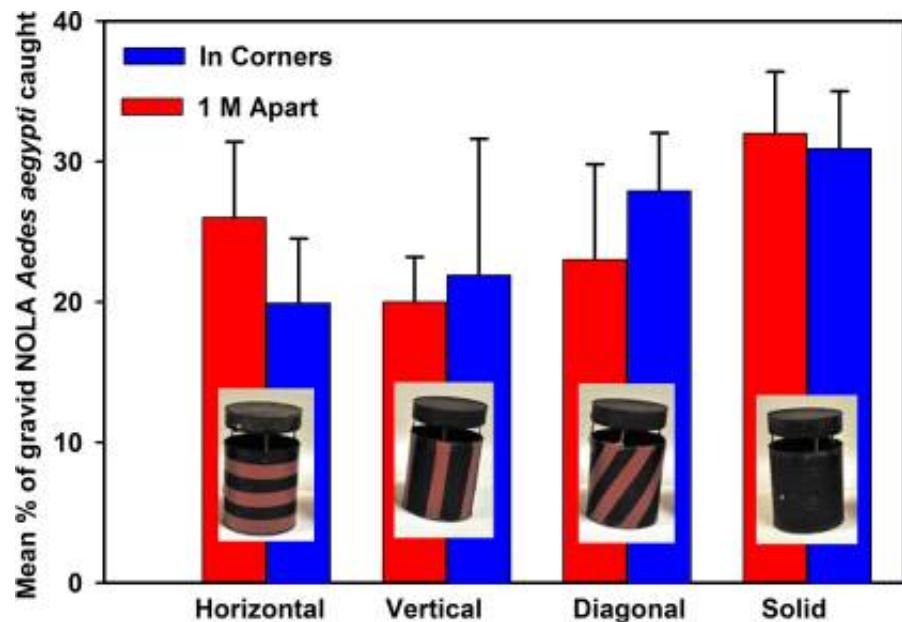


Figura 4 – Padrões em ovitraps

Disponível em: <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4988764/figure/pone.0160386.g009/>>. Acesso em: 22 de novembro de 2021

Outro ponto relevante ao se projetar uma ovitrap é o seu tamanho, segundo (??) 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 tem taxas de evaporação de água muito maiores que as ovitraps maiores.



Figura 5 – Protótipo de ovitrap desenvolvida neste trabalho (sem sensores)

FONTE: Autor

Levando em conta os pontos levantados anteriormente, foi desenvolvido um protótipo de ovitrap (Figura 5), buscou-se desenvolver uma ovitrap atrativa as fêmeas do mosquito, ou seja, grande, com a maior parte preta e que também demandasse manutenções com pouca frequência.

O prototipo foi inspirado no modelo Gravid Aedes Trap (GAT) (??), 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 possivel também passar um insecticida sob a parte transparente para antecipar a morte do mosquito.

Para o desenvolvimento do protótipo foram utilizados os seguintes materiais

### **2.1.1 Vaso maior**



Figura 6 – Vaso grande

FONTE: Autor

Dimensões:

- Diâmetro abertura: 30 cm
- Diâmetro superior 36 cm
- Diâmetro inferior 20 cm
- Altura: 14 cm

### **2.1.2 Vaso menor**



Figura 7 – Vaso menor

FONTE: Autor

Dimensões:

- Diâmetro abertura: 14,5 cm
- Diâmetro superior 17,5 cm
- Diâmetro inferior 10,5 cm
- Altura: 14 cm

### 2.1.3 Prato



Figura 8 – Prato

FONTE: Autor

Dimensões:

- Diâmetro abertura: 16,5 cm
- Diâmetro superior 17,5 cm
- Diâmetro inferior 15,5 cm
- Altura: 2,5 cm

#### 2.1.4 Boleira



Figura 9 – Boleira

FONTE: Autor



Figura 10 – Tampa da boleira após o recorte

FONTE: Autor

Dimensões:

- Diâmetro da abertura recortada: 14,5 cm
- Diâmetro superior 20 cm
- Diâmetro inferior 30 cm
- Altura: 11 cm

### 2.1.5 Tela mosquiteira



Figura 11 – Tela mosquiteira

FONTE: Autor



Figura 12 – Tela mosquiteira após o recorte

FONTE: Autor

Foi feito um recorte em forma de circulo com 30 cm de diâmetro.

### 2.1.6 Breu



Figura 13 – Breu

FONTE: Autor

Breu é uma resina natural utilizada para afinar violinos, envernizar moveis, 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.

### 2.1.7 Palitos de churrasco e fitas hellerman



Figura 14 – Abertura com prato superior

FONTE: Autor

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.

## 2.2 Hardware

Para que fosse possível contar os mosquitos que adentrassem na armadilha foram colocados peças de hardware na armadilha. 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 15 – Sensores conectados e fixos ao prato da armadilha

FONTE: Autor

Na figura acima vemos o módulo NodeMCU, alimentado por um cabo micro USB, e conectado a uma protoboard que o conecta a 5 sensores laser, fixados nas extremidades do prato. Foi utilizada a protoboard para fins de desenvolvimento e teste do conjunto, após a validação do mesmo a protoboard bem como todo esse emaranhado de fios será substituído por um circuito digital bem mais enxuto.

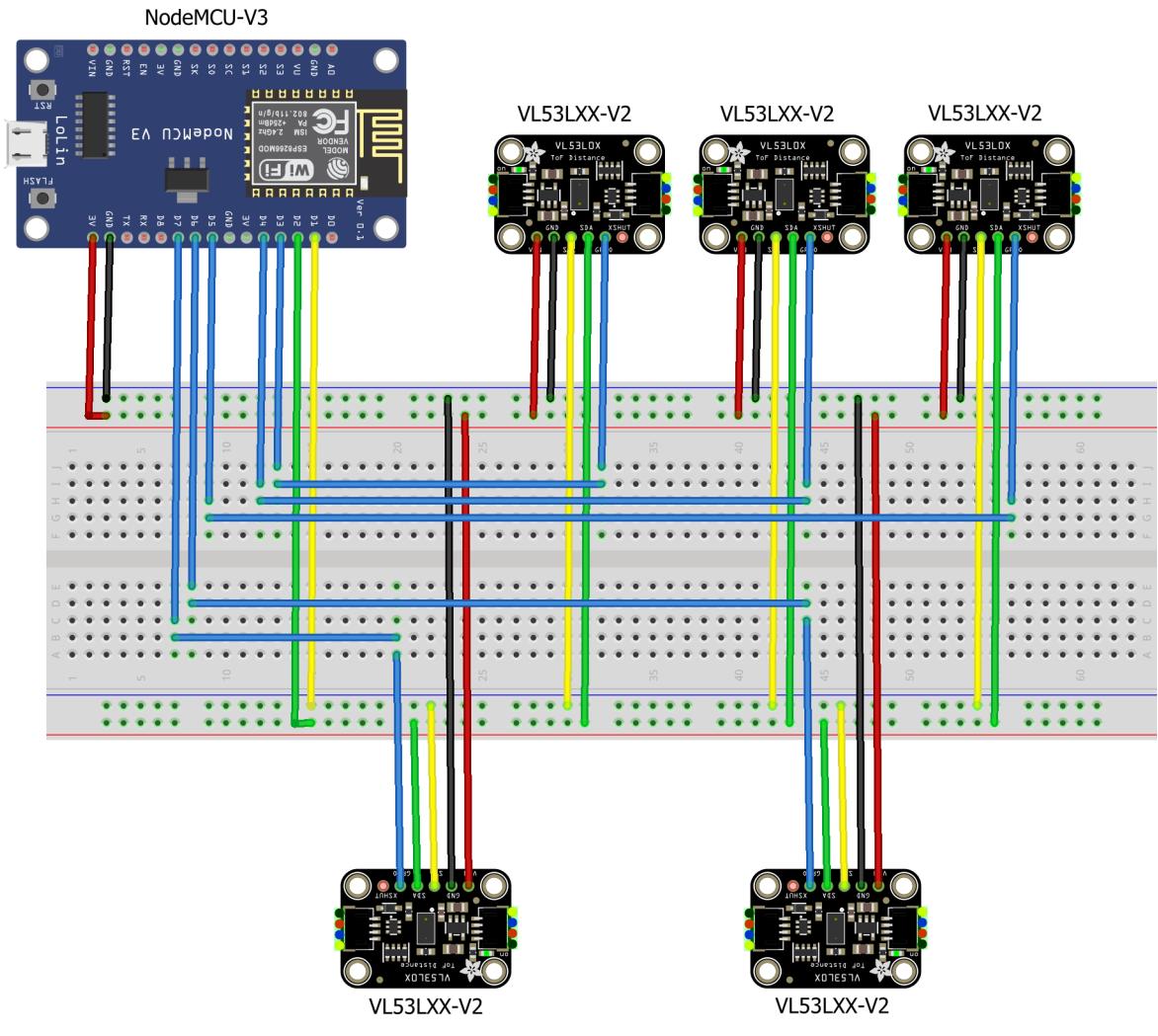


Figura 16 – Esquema de ligação dos sensores

## FONTE· Autor

No esquema acima 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 do VSCode.

Os módulos e sensores utilizados na armadilha são:

### 2.2.1 NodeMCU



Figura 17 – NodeMCU

FONTE: Autor

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.

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



Figura 18 – VL53L0X-V2. (Time-of-Flight)

FONTE: Autor

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

## 2.3 Software

Para receber e armazenar os dados provenientes das armadilhas foi desenvolvido uma API em NodeJS, utilizando o framework Express e banco de dados Postgres.

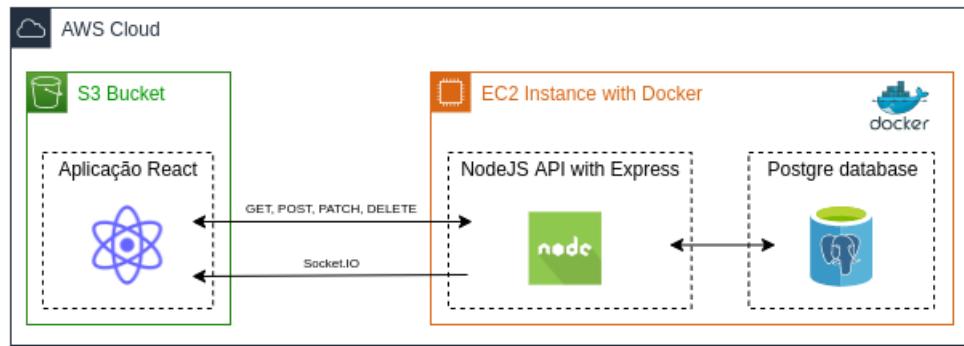


Figura 19 – Diagrama de infraestrutura

FONTE: Autor

O diagrama acima mostra a infraestrutura do projeto na nuvem, que foi hospedado utilizando os serviços da AWS da Amazon. O front-end está hospedado utilizando o serviço de hospedagem de site estático S3. A API bem como o banco de dados estão rodando em containers Docker dentro de uma instância do EC2. A comunicação entre a aplicação cliente e a API é feita utilizando o protocolo HTTP e Web Socket.

### 2.3.1 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 Postgres. A figura abaixo mostra o diagrama de tabelas do banco:

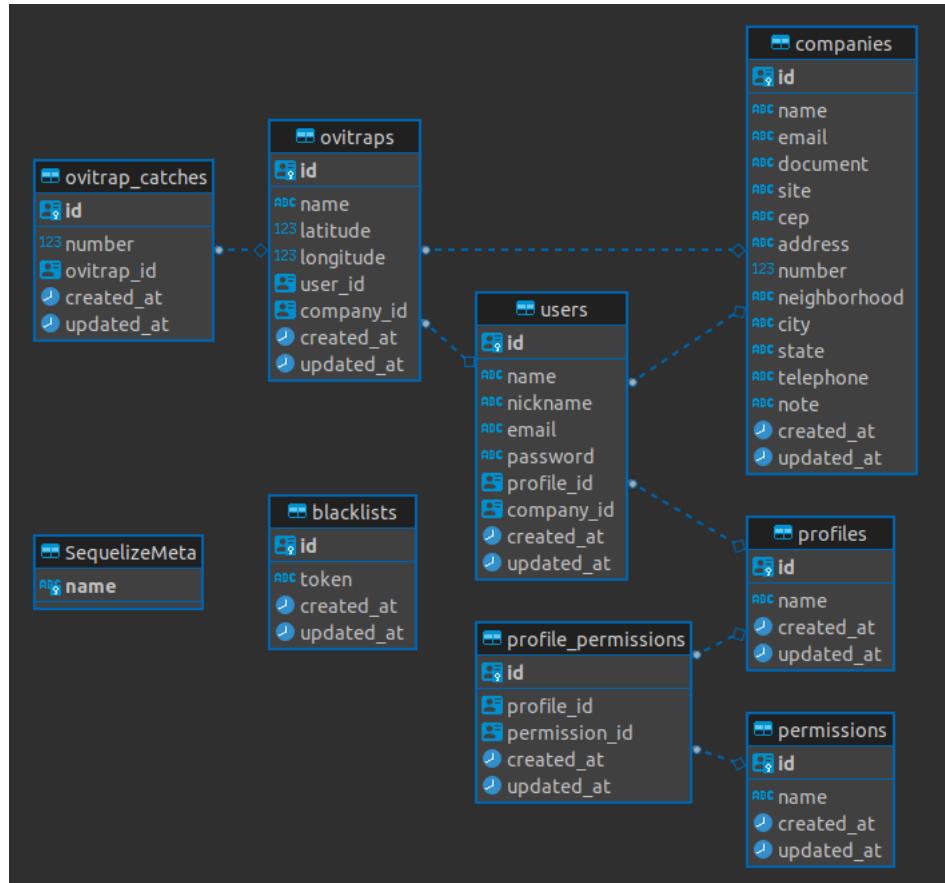


Figura 20 – Diagrama do banco

FONTE: Autor

É inserido um novo registro na tabela ovitrap\_catches a cada captura feita por qualquer armadilha, isso possibilita gerar um grafico do número de capturas em função do tempo.

As tabelas permission, profiles e profile\_permissions são responsáveis por fazer o controle de permissões do usuário na plataforma, uma vez que cada usuário tem um perfil e cada perfil possui permissões específicas para aquele perfil.

A tabela blacklists guarda os tokens inválidos que são inseridos quando o usuário faz o logout ou quando seu token expira.

Para criação das tabelas e inserção dos perfis e permissões foi escolhido o ORM Sequelize que gera a tabela SequelizeMeta para guardar as migrations que já foram executadas no banco.

### 2.3.2 API

A API para consumo dos dados armazenados no banco foi feita em NodeJS utilizando o framework Express. Todas as rotas (exceto login e esqueci minha senha) são protegidas por token, com token de 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, isso possibilita que à aplicação cliente receber dados em tempo real, como por exemplo se houver uma captura de mosquito e o usuário estiver logado na plataforma, o seu dashboard irá atualizar no mesmo instante.

### 2.3.3 Aplicação Cliente

Para exibição dos dados em qualquer navegador web, a aplicação cliente foi desenvolvida utilizando ReactJS, redux-saga e MUI. A aplicação é responsiva, conta com dois temas, claro e escuro, dois idiomas, português e inglês.

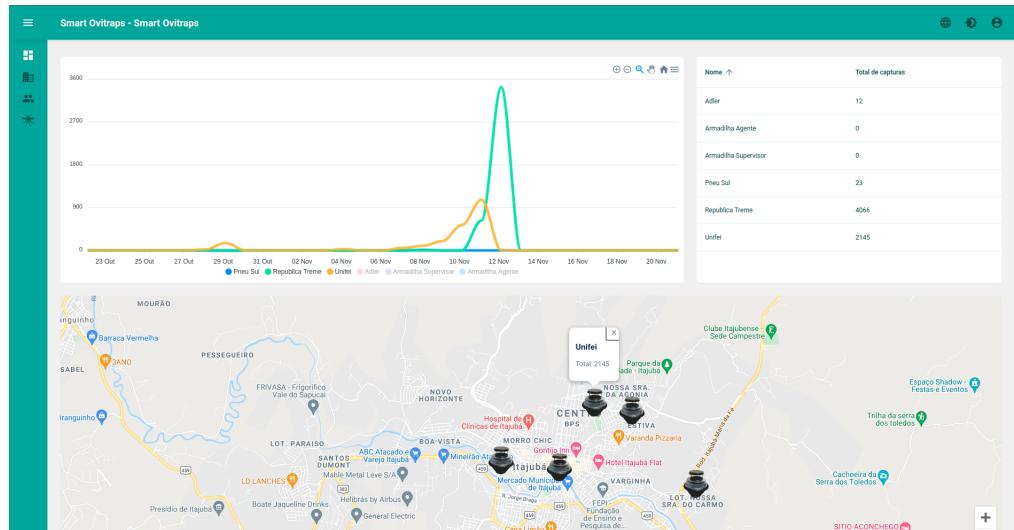


Figura 21 – Tema claro

FONTE: Autor

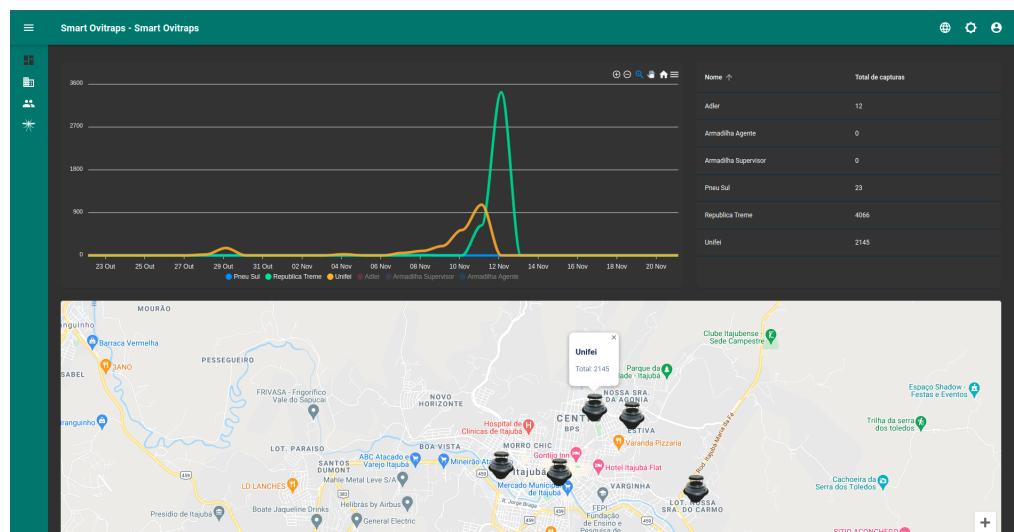


Figura 22 – Tema escuro

FONTE: Autor

No dashboard e no cadastro de armadilhas foram utilizados o serviço do Google Maps, para exibição das armadilhas conforme sua latitude e longitude. No mapa, ao clicar em cada armadilha, é mostrado 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.

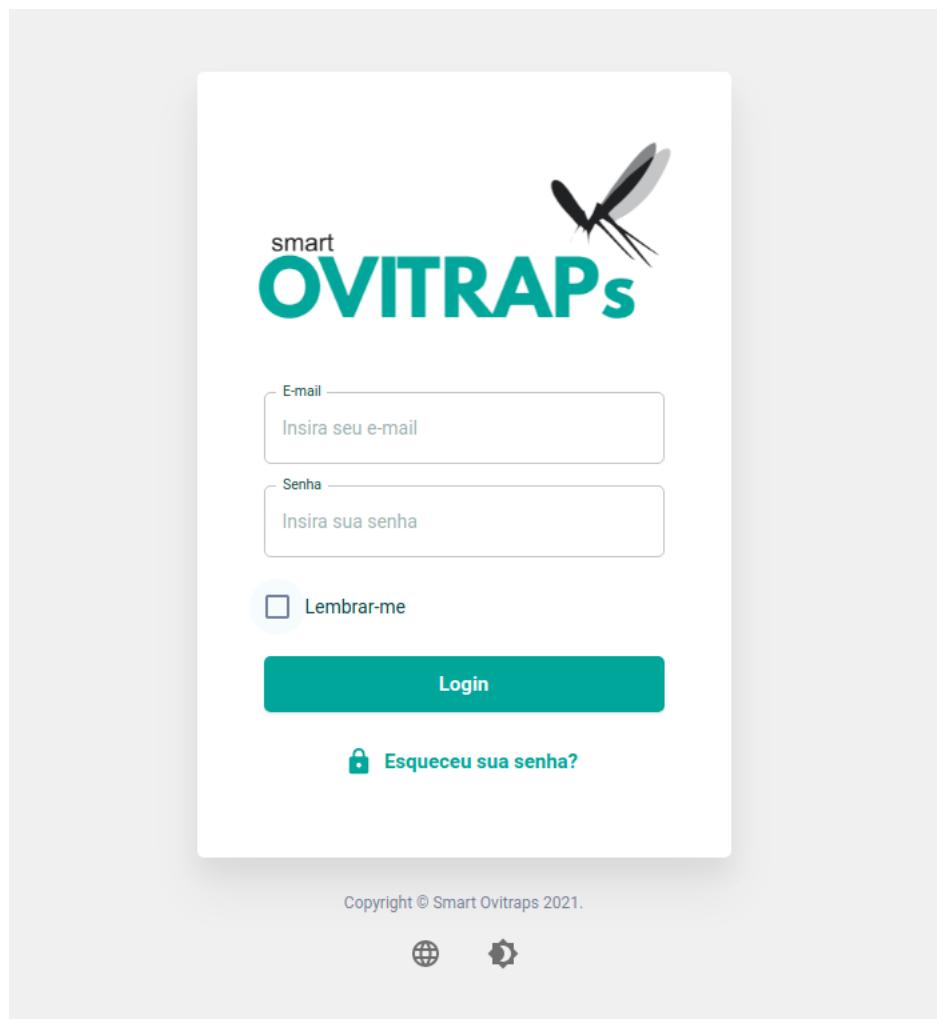


Figura 23 – Tela de login

FONTE: Autor

Para o envio de emails para o usuário no cadastro e no reset de senha foi utilizado o serviço de envio de emails SendGrid.

Smart Ovitraps - Smart Ovitraps			
Ovitraps	Usuário	Empresa	Ação
Adler	Agente Adler	Armadilhas Adler	
Armadilha Agente	Agente Itajubá	Prefeitura de Itajubá	
Armadilha Supervisor	Supervisor Itajubá	Prefeitura de Itajubá	
Pneu Sul	Agente Itajubá	Prefeitura de Itajubá	
República Treme	Agente Daniel	Armadilhas Daniel	

Figura 24 – Tela de listagem de armadilhas

FONTE: Autor

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 propria empresa, cadastrar novas armadilhas, alterar e deletar apenas suas proprias armadilhas.

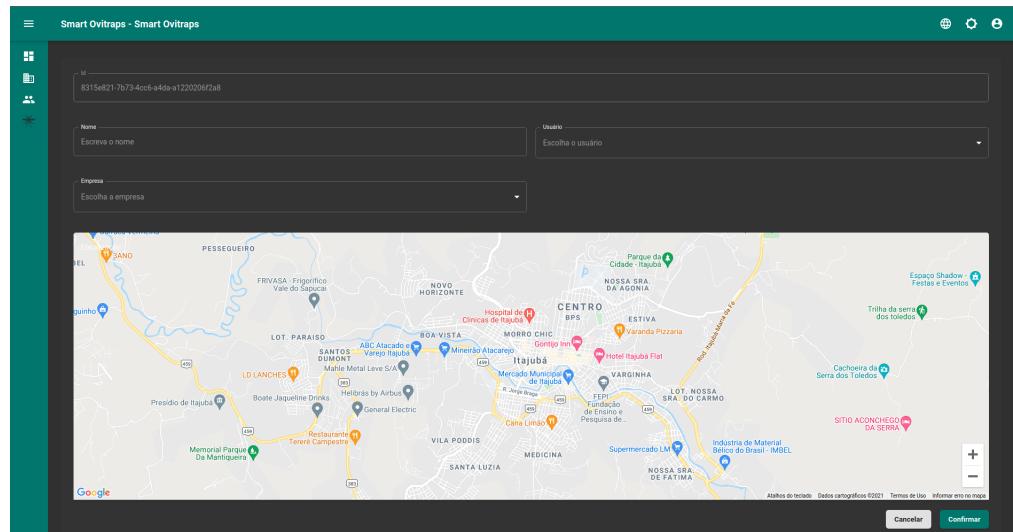


Figura 25 – Tela de cadastro de armadilhas

FONTE: Autor

## 2.4 Visão geral do sistema

Juntando protótipo, sistema embarcado no protótipo e sistema hospedado em nuvem temos o seguinte diagrama:

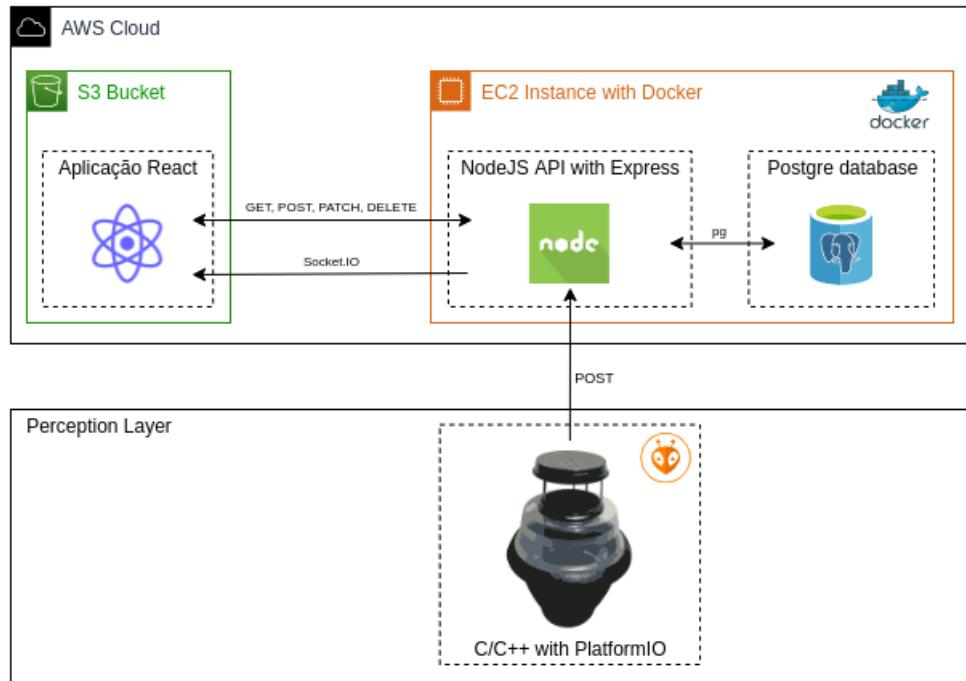


Figura 26 – Tela de cadastro de armadilhas

FONTE: Autor

Nele temos uma visão completa do funcionamento do sistema, tudo começa no sensor da armadilha, que ao medir uma distância menor que o range 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 socket para todos os usuários daquela empresa conectados. A aplicação recebe o payload e exibe uma nova captura no mapa, no gráfico e na tabela.

## 2.5 Custos

Para construção do protótipo foram utilizados materiais que podem ser adquiridos facilmente na internet, lojas de materiais domésticos ou agrícolas. A lista completa de todos os materiais e respectivos valores cotados em reais (R\$) na data da publicação destes trabalho está abaixo:

Descrição	Quantidade por lote	Unidade de medida	Valor lote	Valor por unidade	Quantidade utilizada	Valor total
<b>Vaso maior</b>	<b>1</b>	<b>un</b>	<b>8,99</b>	<b>8,99</b>	<b>1</b>	<b>8,99</b>
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,
Tela com velcro	1,95	m	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 herllerman 120mm	100	un	4,99	0,05	3	0,15
Jumpers	120	un	20,72	0,17	54	9,32
Protopboard	1	un	15	15,00	1	15,00
NodeMCU	1	un	25,5	25,50	1	25,50
VI53X-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
Bastao cola quente	10	un	5,99	0,60	1	0,60
					<b>Total</b>	<b>166,05</b>

### 3 CONCLUSÃO E TRABALHOS FUTUROS

Validar o protótipo da armadilha, em ambiente controlado e em ambiente aberto. Analisar os resultados e calcular a eficácia da mesma.

Cadastrar o id da armadilha, nome e senha da rede wifi de maneira dinamica.

Melhorar a aplicação para que seja possível cadastrar uma foto de perfil para o usuário

Analizar a viabilidade de se trocar o sensor laser por um sensor sonoro, com aplicação de filtro sonoro para a frequência sonora de voo do mosquito.

Procurar alternativas melhores para cola no interior da armadilha, como fitas prega-rato ou fitas prega-mosca. Existe a possibilidade de se incluir uma ventoinha abaixo da abertura para que o mosquito seja sugado para dentro da armadilha.

Apesar de as armadilhas somente mandarem dados para API quando um sensor é disparado e não a cada segundo. Seria interessante separar os dados em 2 bancos, um relacional para os dados de empresas e usuários, e um não relacional como MongoDB os dados vindo das armadilhas. O banco não relacional trabalha melhor que o relacional quando existe uma quantidade muito grande de dados. Também seria interessante pensar em uma arquitetura de banco multi-tenant (um banco para cada cliente) visando a escalabilidade e o isolamento dos dados.

**APENDICE A - DOCUMENTO DE REQUISITOS**

## **REFERÊNCIAS**