



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 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 dos dados. Aplicando 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 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, 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 and Chikungunya.

LISTA DE ILUSTRAÇÕES

Figura 1 – Primeiro modelo de ovitrap, desenvolvido na década de 70	12
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)	13
Figura 3 – Modelo de ovitrap desenvolvido por Ismaliza Isa et al.	14
Figura 4 – Primeiros testes	15
Figura 5 – Esquema de pastas de fotos para larvas e mosquitos no Amazon S3	16
Figura 6 – Foto registrada pelo ESP32-CAM	16
Figura 7 – Padrões em ovitraps	17
Figura 8 – Protótipo de ovitrap desenvolvida neste trabalho (sem sensores)	18
Figura 9 – Vaso maior	19
Figura 10 – Vaso menor	19
Figura 11 – Prato	20
Figura 12 – Boleira	21
Figura 13 – Tampa da boleira após o recorte	21
Figura 14 – Tela mosquiteira	22
Figura 15 – Tela mosquiteira após o recorte	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	26
Figura 21 – VL53L0X-V2. (Time-of-Flight)	26
Figura 22 – Diagrama de infraestrutura	27
Figura 23 – Diagrama do banco	28
Figura 24 – Tema claro	29
Figura 25 – Tema escuro	29
Figura 26 – Tela de login	30
Figura 27 – Tela de listagem de armadilhas	31
Figura 28 – Tela de cadastro de armadilhas	31
Figura 29 – Tela de cadastro de armadilhas	32
Figura 30 – Protótipo final da armadilha em funcionamento	33
Figura 31 – Diagrama do sistema	39
Figura 32 – Diagrama de Casos de Uso	41

LISTA DE TABELAS

Tabela 1 – Tabela de custo	34
--------------------------------------	----

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	12
1.1	O que é uma ovitrap?	12
1.2	Evolução das ovitraps	12
1.3	Motivação	13
1.4	Justificativa	14
2	DESENVOLVIMENTO	15
2.1	A ideia inicial	15
2.2	Ovitrap	17
2.2.1	Vaso maior	19
2.2.2	Vaso menor	19
2.2.3	Prato	20
2.2.4	Boleira	21
2.2.5	Tela mosquiteira	22
2.2.6	Breu	23
2.2.7	Palitos de churrasco e fitas hellerman	23
2.3	Hardware	24
2.3.1	NodeMCU	26
2.3.2	VL53L0X (Time-of-Flight)	26
2.4	Software	26
2.4.1	Banco de dados	27
2.4.2	API	28
2.4.3	Aplicação Cliente	29
2.5	Visão geral do sistema	32
2.6	Custos	33
3	CONCLUSÃO E TRABALHOS FUTUROS	35
	REFERÊNCIAS	36
	APENDICE A - DOCUMENTO DE REQUISITOS	37

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 (ISA et al., 2019). Uma ferramenta de controle que vem sendo utilizada desde a década de 70 são os ovitraps (LOK et al., 1977)

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 (LOK et al., 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 a fêmea do mosquito que deposita seus ovos na armadilha não seja morta, os filhotes que ecodem dos ovos ficam presos pela malha trançada e morrem afogados.

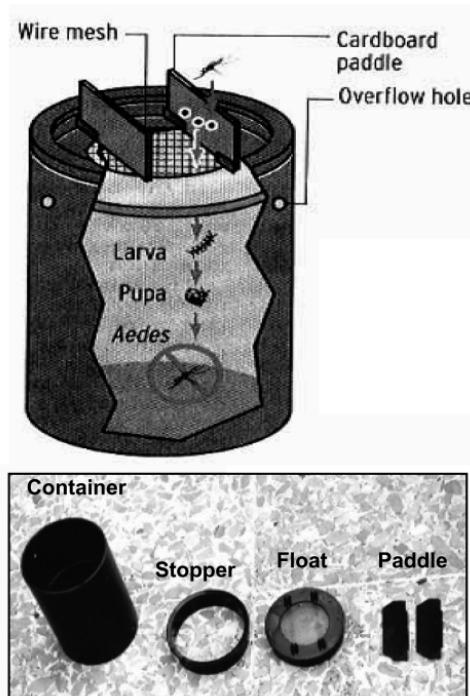


Figura 1 – Primeiro modelo de ovitrap, desenvolvido na década de 70

Disponível em: <<https://www.appropedia.org/Ovitrap>>. Acesso em: 23 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 (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.

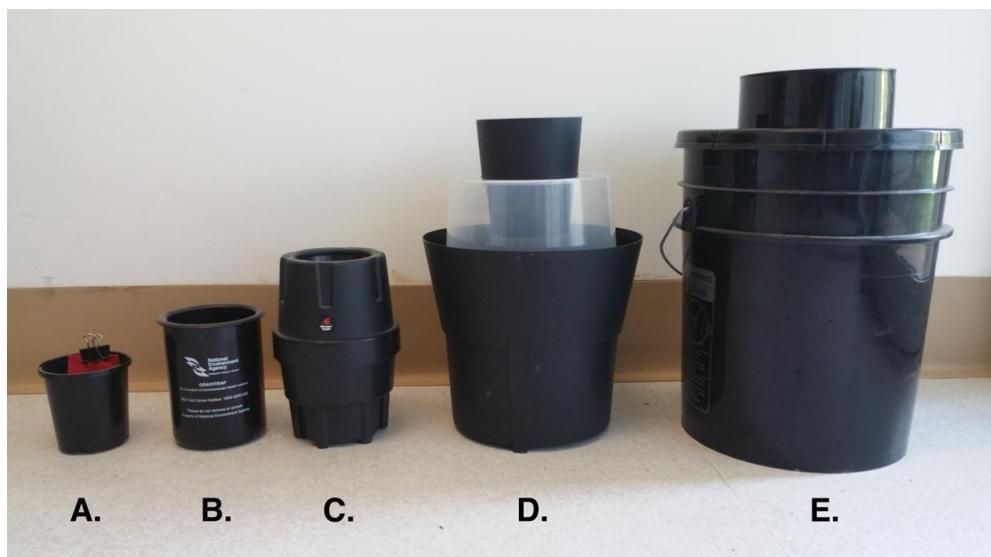


Figura 2 – (A) Standard Lethal Ovitrap (LO), (B) National Environmental Agency Singapore Sticky Ovitrap (SO), (C) MosquiTRAP Sticky Ovitrap (SO), (D) Biogents Gravid Aedes Trap (GAT), (E) Centers for Disease Control (CDC) Autocidal Gravid Ovi-trap(AGO)

Disponível em: <https://www.researchgate.net/publication/312158747_The_State_of_the_Art_of_Lethal_Oviposition_Trap-Based_Mass_Interventions_for_Arboviral_Control/figures>. Acesso em: 23 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 (JOHNSON et al., 2017), 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.

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 Ismaliza e outros (ISA et al., 2019) (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>. Acesso em: 23 de novembro de 2021

O modelo desenvolvido por Ismaliza e outros 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 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.

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

2.1 A ideia inicial

A princípio a ideia inicial deste trabalho era desenvolver uma armadilha que pudesse além de capturar, também fotografar os mosquitos, para depois aplicar processamento digital de imagens e catalogar as espécies de mosquitos fotografadas. E se não bastasse isso, também havia uma câmera submersa na água da armadilha para que fosse possível fotografar as larvas do mosquito.

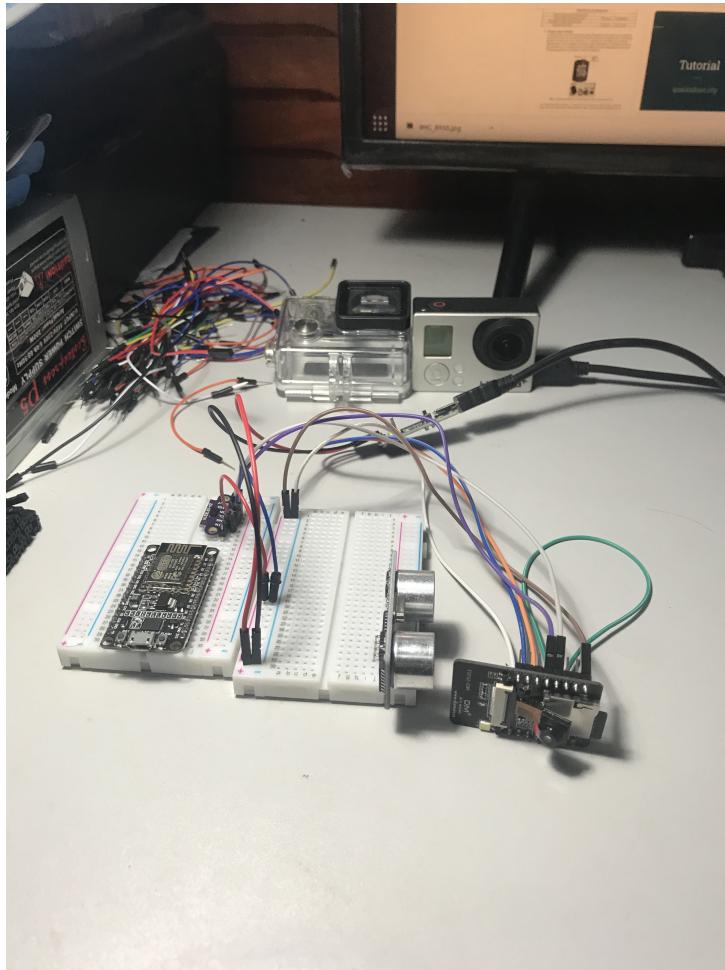


Figura 4 – Primeiros testes

FONTE: Autor

A figura acima é primeira foto do desenvolvimento deste trabalho, ela mostra um módulo NodeMCU, um sensor laser (o mesmo usado no protótipo final), um sensor ultrassônico, um módulo ESP32-CAM e uma câmera GoPro. A foto foi tirada no primeiro de vários testes.

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 câmera esp era

ativada pelo sensor laser, e a GoPro era ativada a cada 1 hora, lembre-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 modulo ESP32-CAM também se conectava com uma rede Wifi externa, através da qual enviava as fotos via requisição HTTP para API. O tempo de envio das fotos da GoPro para API era um pouco lento, devido a alta resolução das fotos, já o tempo do ESP32 era aceitável. Outro ponto era que essas fotos ficavam armazenadas em um cartão Micro-SD no módulo ESP32-CAM. O fato de ter que conectar em duas redes diferentes, ter de usar um cartão Micro-SD, além do custo do projeto ficar elevado devido a GoPro, foi desenvolvido um segundo teste sem a GoPro, apenas com o ESP32-CAM e o sensor laser.

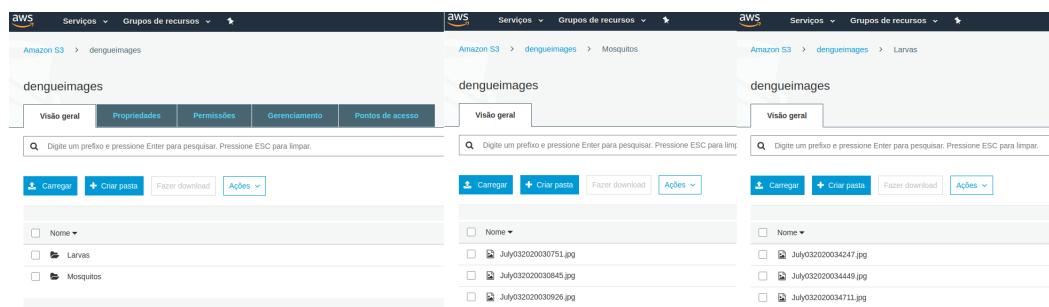


Figura 5 – Esquema de pastas de fotos para larvas e mosquitos no Amazon S3

FONTE: Autor

Neste segundo teste havia apenas uma câmera ESP32-CAM, utilizando um cartão Micro-SD e fixada no teto da armadilha, o fato de usar apenas uma câmera, eliminou a necessidade de se conectar em duas redes diferentes, e o tempo de upload de arquivos para API era menor que do primeiro teste, uma vez que as fotos tiradas pelo ESP32-CAM tem no máximo 2MB de resolução. Se por um lado isso é bom para comunicação com a API, por outro lado é ruim, pois é 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

FONTE: Autor

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 da armadilha de maneira segura para uma API.
- 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.

2.2 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 JOSH YUKICH et al., 2016) constataram que os mosquitos fêmeas são atraídos por ovitraps totalmente pretas.

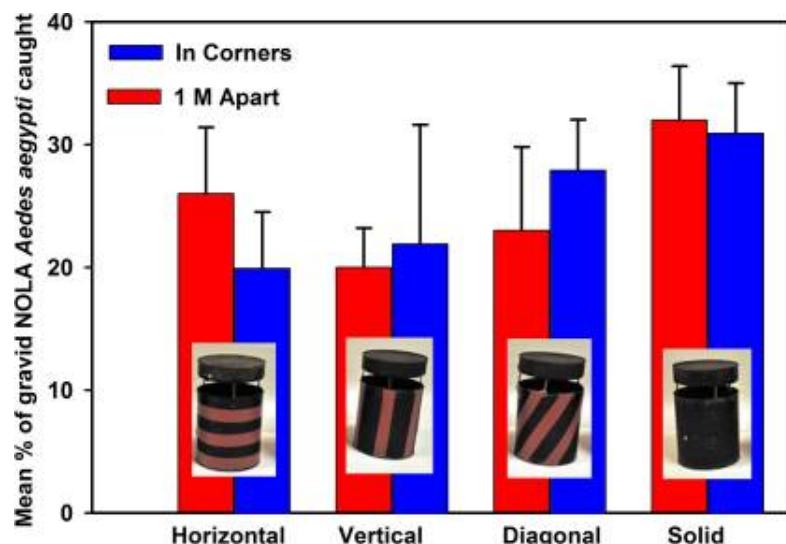


Figura 7 – Padrões em ovitraps

Disponível em: <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4988764/figure/pone.0160386.g009/>>. Acesso em: 23 de novembro de 2021

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 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 protótipo foi inspirado no 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:

2.2.1 Vaso maior



Figura 9 – Vaso maior

FONTE: Autor

Dimensões:

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

2.2.2 Vaso menor



Figura 10 – Vaso menor

FONTE: Autor

Dimensões:

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

2.2.3 Prato



Figura 11 – Prato

FONTE: Autor

Dimensões:

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

2.2.4 Boleira



Figura 12 – Boleira

FONTE: Autor



Figura 13 – 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

2.2.5 Tela mosquiteira



Figura 14 – Tela mosquiteira

FONTE: Autor



Figura 15 – Tela mosquiteira após o recorte

FONTE: Autor

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

2.2.6 Breu



Figura 16 – Breu

FONTE: Autor

Breu é uma resina natural utilizada para afinar violinos, 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.

2.2.7 Palitos de churrasco e fitas hellerman



Figura 17 – 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.

2.3 Hardware

Para que fosse possível contar os mosquitos que adentrassem na armadilha foram colocados peças de hardware nesta. 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

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.

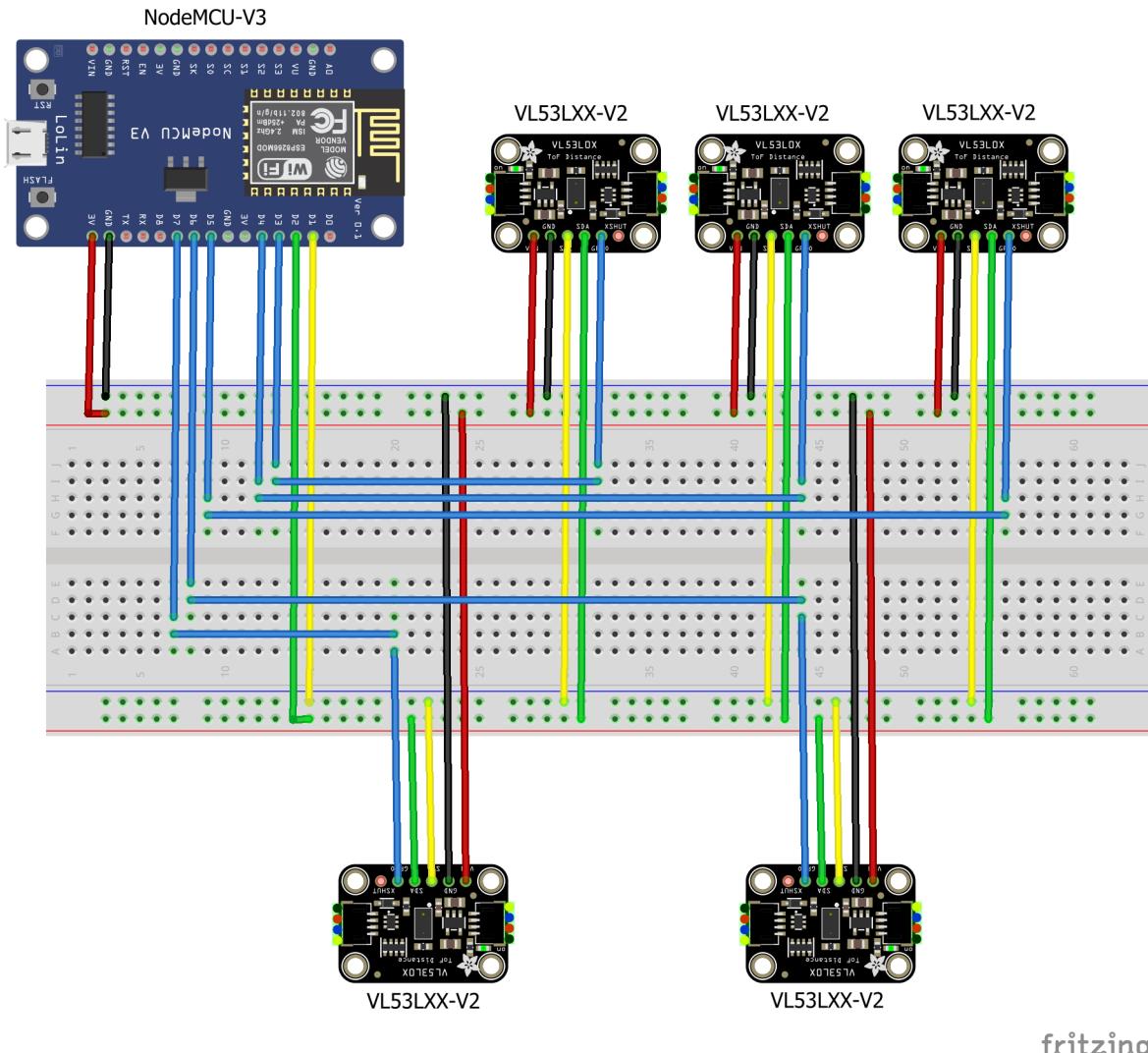


Figura 19 – Esquema de ligação dos sensores

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:

2.3.1 NodeMCU

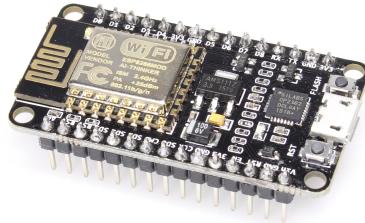


Figura 20 – NodeMCU

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.

2.3.2 VL53L0X (Time-of-Flight)



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

FONTE: Autor

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.

2.4 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.

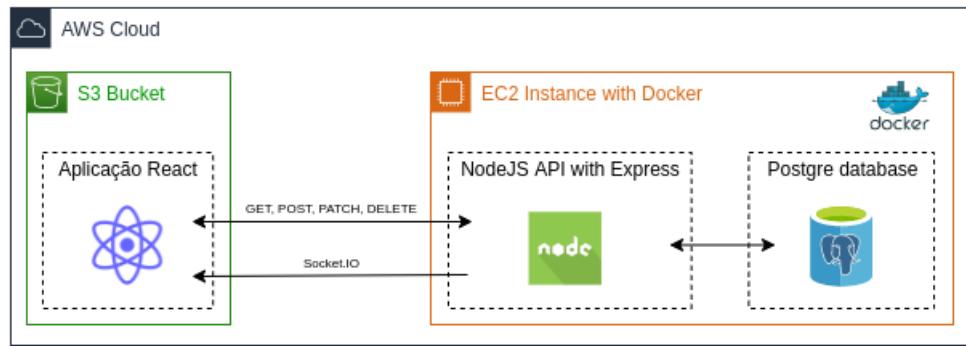


Figura 22 – Diagrama de infraestrutura

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 Amazon S3. A API bem como o banco de dados estão rodando em containers Docker dentro de uma instância do Amazon EC2. A comunicação entre a aplicação cliente e a API é feita utilizando o protocolo HTTP e Web Socket.

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

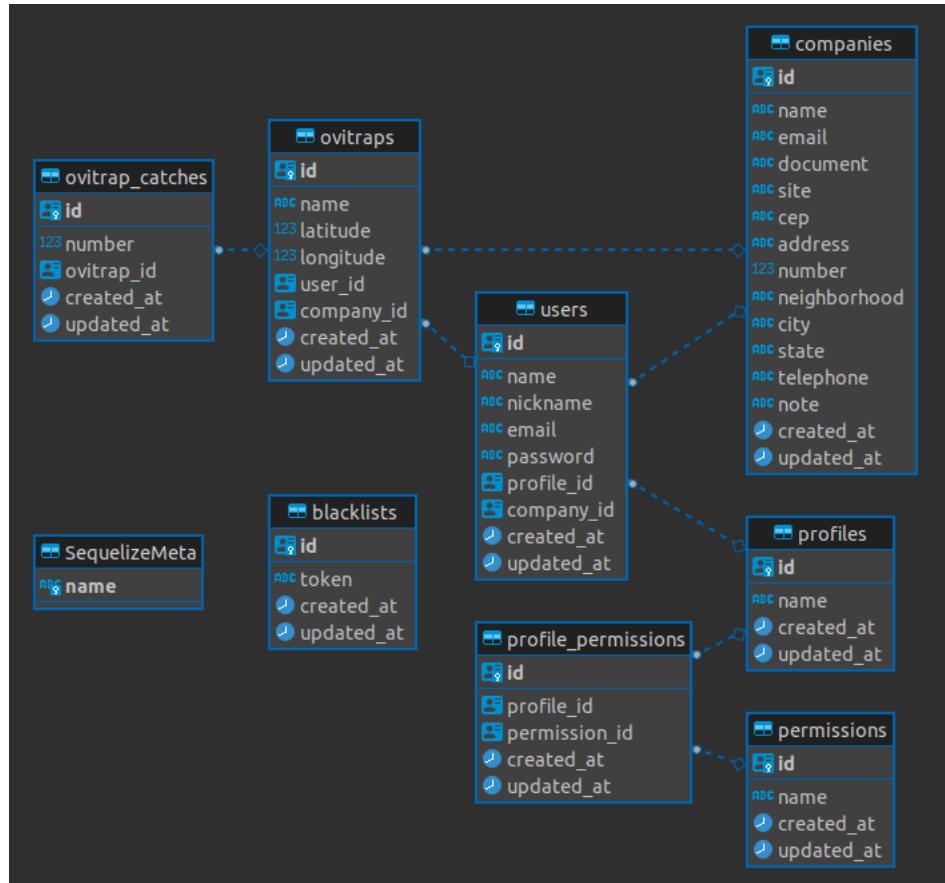


Figura 23 – Diagrama do banco

FONTE: Autor

É inserido um novo registro na tabela ovitrap_catches a cada captura feita por qualquer armadilha, isso possibilita gerar um gráfico 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 este.

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.

2.4.2 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 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 a aplicação cliente receba 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.

2.4.3 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 e conta com dois temas, claro e escuro, dois idiomas, português e inglês.

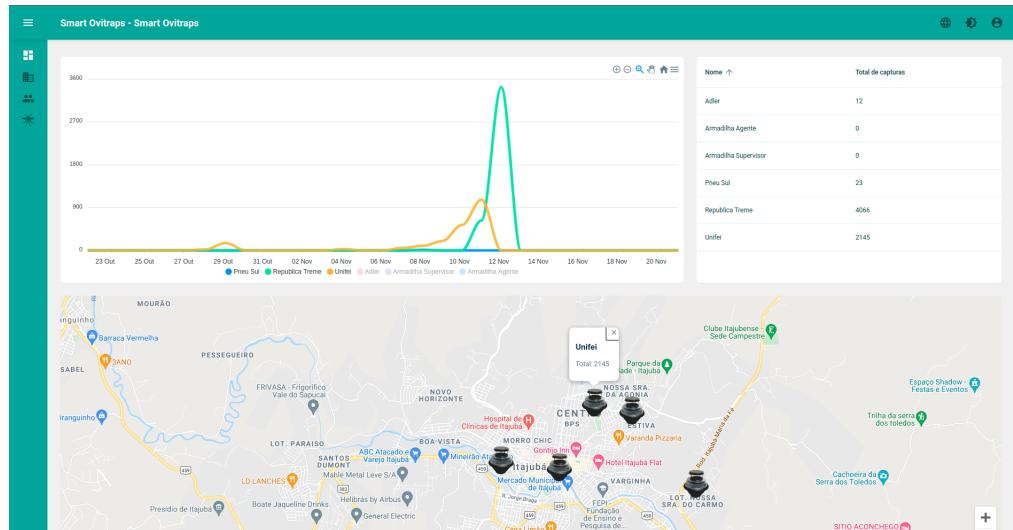


Figura 24 – Tema claro

FONTE: Autor

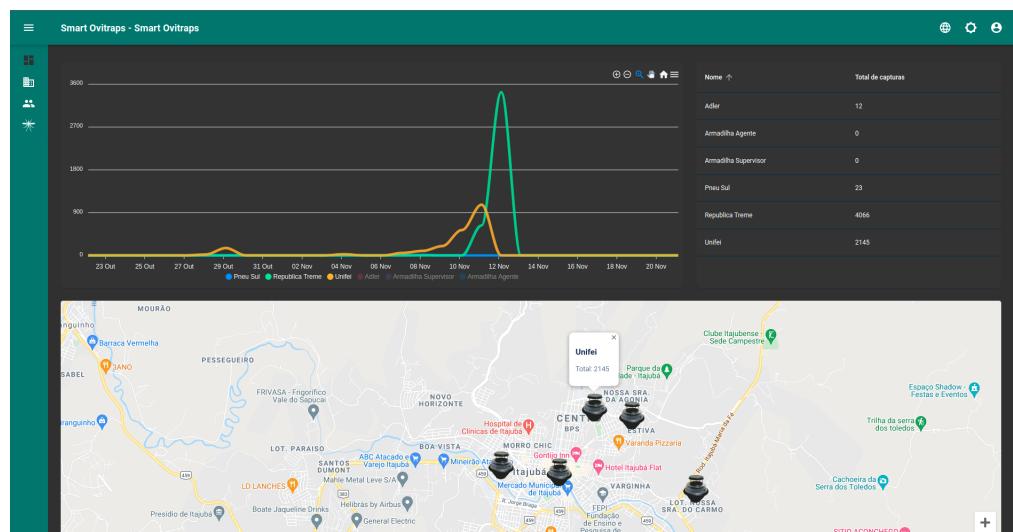


Figura 25 – Tema escuro

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.

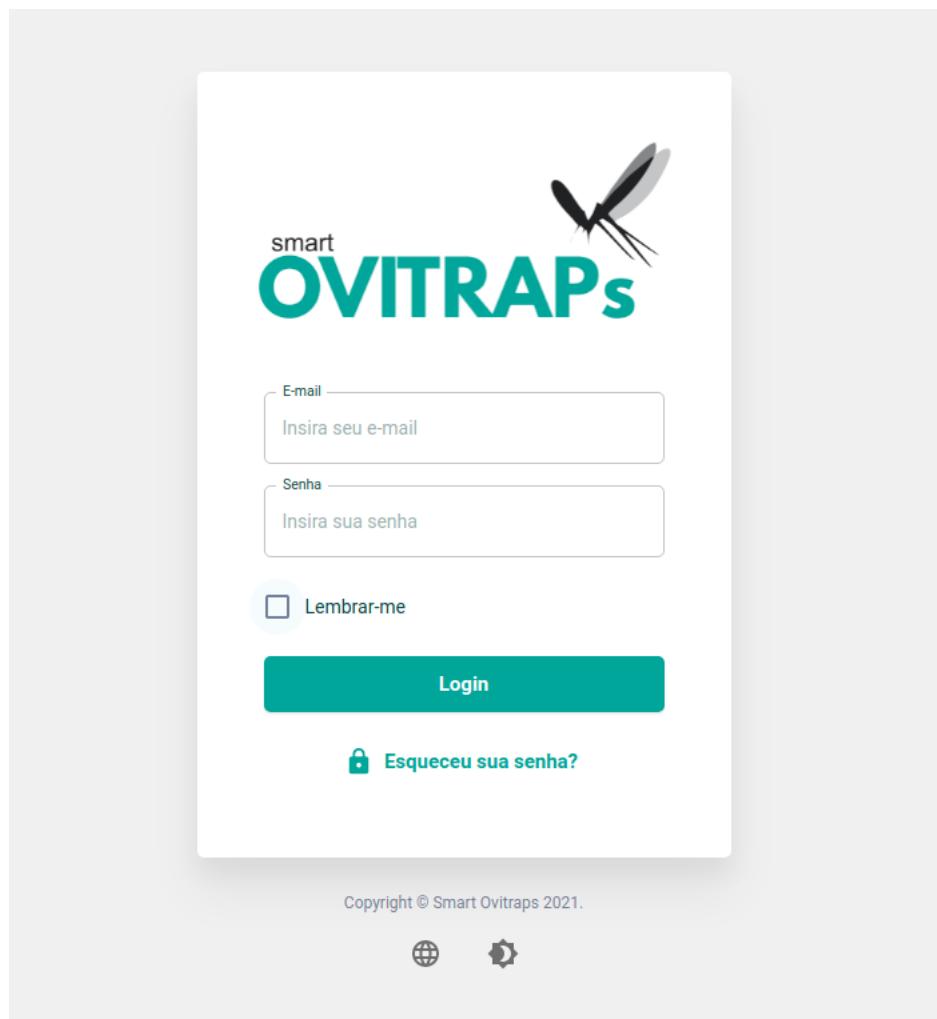


Figura 26 – Tela de login

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 Ovitrap - Smart Ovitrap		
Ovitrap	Usuário	Empresa
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 27 – 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 própria empresa, cadastrar novas armadilhas, alterar e deletar apenas suas próprias armadilhas.

The screenshot shows the 'Smart Ovitrap - Smart Ovitrap' application. At the top, there's a header with a menu icon, user profile, and settings. Below the header is a table listing 'Ovitrap' entries with columns for 'Nome', 'Usuário', and 'Empresa'. A green button labeled 'Adicionar' is at the top right of the table. The main area contains a form for creating a new trap. The 'Id' field is populated with a unique identifier. The 'Nome' field is a text input with placeholder 'Escreva o nome'. The 'Usuário' field is a dropdown menu. The 'Empresa' field is also a dropdown menu with 'Escolha a empresa' as the placeholder. Below the form is a map of Itajubá, Brazil, showing various neighborhoods like Centro, BPS, Varginha, and Serra dos Toledo. Numerous trap locations are marked with orange dots across the city. At the bottom of the map, there are buttons for 'Cancelar' and 'Confirmar'.

Figura 28 – Tela de cadastro de armadilhas

FONTE: Autor

2.5 Visão geral do sistema

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

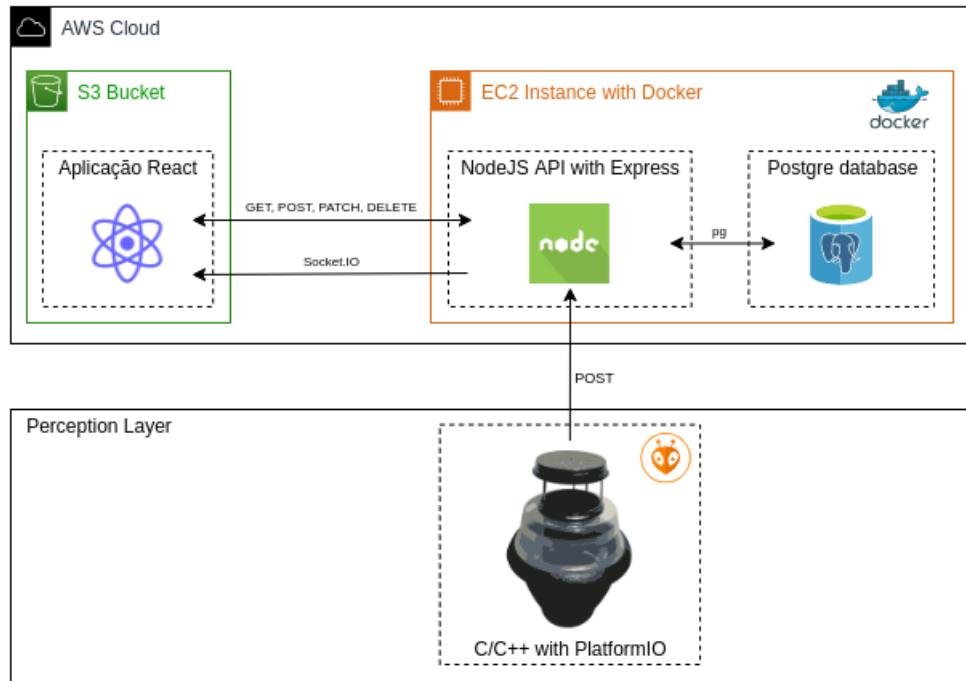


Figura 29 – Tela de cadastro de armadilhas

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.

O código de cada parte do sistema está disponível no link:

<https://gitlab.com/users/danielreis58/projects>

Os nomes de cada repositório são:

A Cloud IoT-Ovitrap System - ESP32, A Cloud IoT-Ovitrap System - API e A Cloud IoT-Ovitrap System - Dashboard



Figura 30 – Protótipo final da armadilha em funcionamento

FONTE: Autor

2.6 Custos

Para construção do protótipo foram utilizados materiais que podem ser adquiridos facilmente na internet, lojas de materiais domésticos ou agrícolas. Abaixo temos a tabela completa de todos os materiais e valores, cotados na data de publicação deste trabalho, na moeda brasileira real.

Tabela 1 – 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,
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 herllerman 120mm	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
VL53X-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	166,05

Observamos a partir da tabela acima que o custo com componentes eletrônicos foi de R\$133,904 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.

3 CONCLUSÃO E TRABALHOS FUTUROS

Baseado nos testes realizados com o protótipo em campo aberto, observou-se que a maior dificuldade é atrair mosquitos para armadilha, sendo que esta foi deixada por uma semana em no jardim, e não foi capturado nenhum mosquito. Houve alguns registros do sensor, porém nenhum mosquito foi capturado pela cola interna da armadilha. A cola desenvolvida com breu, após alguns dias derreteu e caiu na tela mosquiteira da armadilha devido ao calor do sol. 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 consistentes e pode ser replicada para qualquer outra rede de sensores com pouquíssimas modificações, a estrutura do banco está bem definida e todo o sistema de perfis e permissões podem ser customizados de acordo com a necessidade do cliente.

Como 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.
 - Procurar alternativas melhores para cola no interior da armadilha como: fitas pregarato 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 wifi de maneira dinâmica.
 - Melhorar a aplicação para que seja possível cadastrar uma foto de perfil para o usuário.

REFERÊNCIAS

- 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.
- 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.
- 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.
- PAZ-SOLDAN JOSH YUKICH, A. S. 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.

APENDICE A - DOCUMENTO DE REQUISITOS

DOCUMENTO DE REQUISITOS ESPECIFICAÇÃO

PARA

SMART OVITRAPS

Version 2.0

Preparado por : Daniel Pinheiro dos Reis

23 de novembro 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

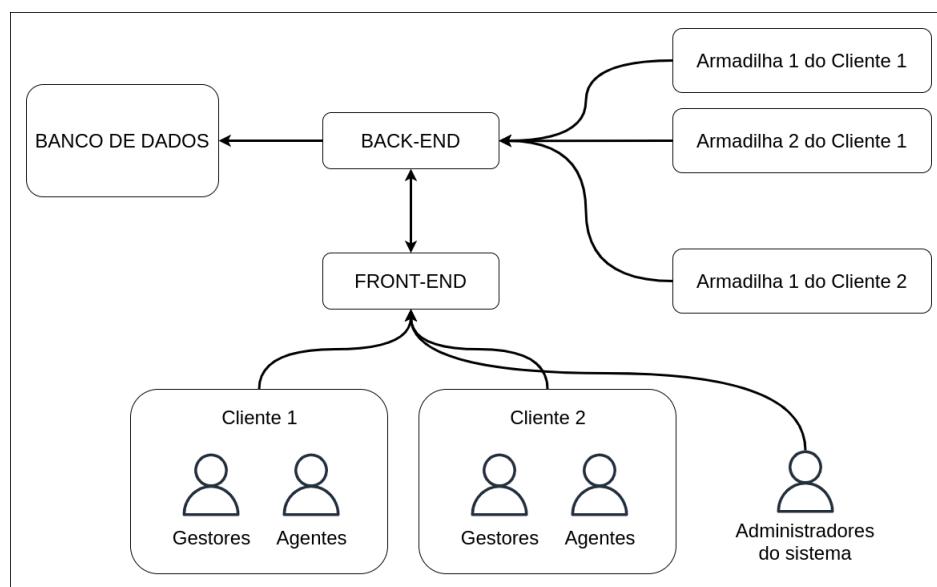


Figura 31 – Diagrama do sistema

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

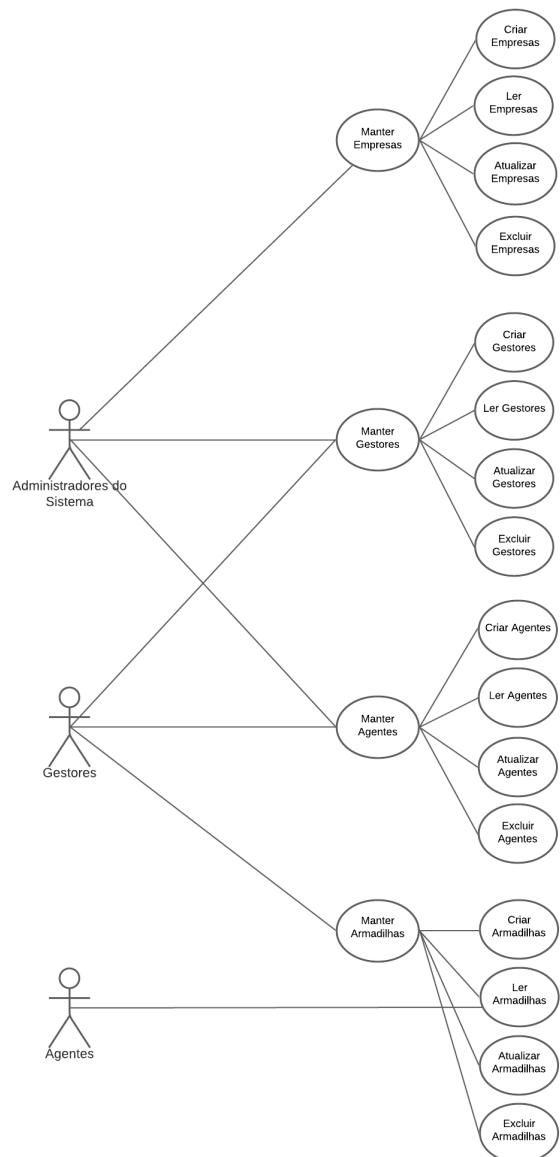


Figura 32 – Diagrama de Casos de Uso

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.