

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
CURSO DE CIÊNCIA DA COMPUTAÇÃO

LUCAS RONCHETTI NUNES

**Aplicativo móvel para Mapeamento  
Colaborativo de Focos de *Aedes aegypti*  
Utilizando Técnicas de Machine Learning**

Monografia apresentada como requisito parcial  
para a obtenção do grau de Bacharel em Ciência  
da Computação

Orientador: Prof. Dr. Weverton Luis da Costa  
Cordeiro  
Co-orientador: Prof. Dr. Mariana  
Recamonde-Mendoza

Porto Alegre  
2020

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos André Bulhões

Vice-Reitora: Prof<sup>ª</sup>. Patricia Pranke

Pró-Reitor de Graduação: Prof. Vladimir Pinheiro do Nascimento

Diretora do Instituto de Informática: Prof<sup>ª</sup>. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Ciência de Computação: Prof. Sérgio Luis Cechin

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

## RESUMO

O mosquito *Aedes aegypti* é um vetor de transmissão de diversas doenças tropicais. A identificação de focos de mosquito ocorre atualmente de forma quase artesanal, através da instalação de armadilhas e análise em laboratório dos mosquitos capturados, bem como pela identificação de nascedouros da espécie. No intuito de fornecer à comunidade uma ferramenta para o mapeamento imediato da ocorrência de mosquitos em uma determinada região, o objetivo do presente trabalho é a pesquisa e o desenvolvimento de um aplicativo para *smartphones* que possa identificar *A. aegypti* a partir da análise do som gerado pelo bater de asas do mosquito. Utilizou-se a plataforma Android como base para a aplicação e um banco de dados na nuvem para a armazenagem dos dados pertinentes gerados na identificação positiva. Uma rede neural previamente treinada com um *dataset* de áudios de mosquitos foi implementada com sucesso no aplicativo móvel. Testes foram conduzidos no intuito de identificar o efeito de diferentes ruídos de fundo sobre a performance do aplicativo. Com ruído ambiente, o aplicativo identifica o mosquito com sucesso a uma distância de 40 cm. Sons gerados pelo compressor de um ar-condicionado e por uma televisão têm efeitos deletérios sobre a classificação. A aplicação desenvolvida demonstra potencial para se tornar uma valiosa ferramenta no combate à disseminação do mosquito *A. aegypti*, o que deve se traduzir em maior controle sobre as endemias das quais ele é vetor.

**Palavras-chave:** Aplicativo móvel. Android. *Machine learning*. *Aedes aegypti*.

## ABSTRACT

The *Aedes aegypti* mosquito is the vector of several tropical diseases. Currently, the geographical distribution of the mosquito is detected via a crude method based on the distribution of traps and the posterior analysis of the captured mosquitos, as well as by the identification of larvae incubation sites. In order to provide a tool for the swift mapping of the mosquito occurrences in a particular region, this work aimed to research and develop a smartphone application that would identify species of mosquitoes through the sound generated by their wingbeat. The Android platform was used as the basis for the application and a cloud database served as storage of the data generated after a positive identification. A neural network previously trained with a dataset of mosquito audios was successfully implemented in the mobile application. Tests were conducted in order to identify the effects of different background noises on the performance of the application. The system successfully identifies the mosquito at a distance of up to 40 cm in the absence of external sources of noise. The sounds generated by the compressor of an air conditioner and by a television have deleterious effects on the classification. The developed application has shown promise as a valuable tool in the fight against the spread of the mosquito, which should translate in greater control over the diseases transmitted by it.

**Keywords:** Mobile application. Android. Machine learning. *Aedes aegypti*.



## LISTA DE FIGURAS

Figura 1.1 Evolução do número de municípios do Rio Grande do Sul com registro de focos de <i>Aedes aegypti</i> . .....	11
Figura 2.1 Mosquito fêmea da espécie <i>Aedes aegypti</i> . .....	13
Figura 2.2 Agente de saúde realizando pesquisa em armadilha (larvitrapa). .....	15
Figura 2.3 Espectrograma do voo de um mosquito <i>Anopheles gambiae</i> capturado com um dispositivo móvel e sua distribuição de frequências. ....	16
Figura 2.4 Distribuição de frequências do batimento de asas de mosquito capturadas com sensor óptico. ....	16
Figura 2.5 Espectrograma do voo de mosquito <i>A. aegypti</i> . ....	17
Figura 2.6 Exemplo da operação de convolução. ....	20
Figura 2.7 Função sigmóide. ....	21
Figura 2.8 Função ReLU. ....	21
Figura 4.1 Componentes da plataforma Android. ....	25
Figura 4.2 Distribuição de mercado de sistemas operacionais móveis no Brasil. ....	26
Figura 4.3 Jornada de usuário 1: visualização do mapa. ....	29
Figura 4.4 Jornada de usuário 2: identificação e compartilhamento de ocorrência. ....	29
Figura 4.5 Diagrama do modelo de dados de uma denúncia. ....	31
Figura 4.6 Exemplo do padrão MVP. ....	33
Figura 5.1 Tela inicial. ....	35
Figura 5.2 Detalhe da ocorrência no mapa. ....	36
Figura 5.3 Mapa de calor das ocorrências do mosquito. ....	37
Figura 5.4 Tela de análise de áudio. ....	38
Figura 5.5 Alerta de detecção de <i>A. aegypti</i> . ....	39
Figura 5.6 Alerta de não detecção de <i>A. aegypti</i> . ....	40
Figura 5.7 Tela de relato da ocorrência de <i>A. aegypti</i> . ....	41
Figura 6.1 Esquema do suporte desenvolvido para testes. Todas as dimensões descritas estão em milímetros. ....	42
Figura 6.2 Espectro sonoro do ambiente sem fontes de ruído. ....	44
Figura 6.3 Espectro sonoro do ambiente com os ruídos do ar condicionado (à esquerda) e da televisão (à direita). ....	45

Figura 6.4 Intensidade sonora média no ambiente de testes. As linhas são meramente auxílios visuais. ....	46
Figura 6.5 Tempo de detecção positiva de <i>A. aegypti</i> em função da distância da fonte emissora de som. ....	47
Figura 6.6 Tempo de detecção positiva de <i>A. aegypti</i> em função da distância da posição da fonte emissora de som, com e sem ruído do ar condicionado. ....	48

## LISTA DE TABELAS

Tabela 4.1 <i>User stories</i> do aplicativo. ....	28
--	----

## **LISTA DE ABREVIATURAS E SIGLAS**

API	Application Programming Interface
ART	Android Runtime
AOT	Ahead-of-time
JIT	Just-in-time
ML	Machine Learning
MVP	Model-View-Presenter
RNA	Rede Neural Artificial
SMS	Secretaria Municipal de Saúde

## SUMÁRIO

<b>LISTA DE FIGURAS .....</b>	<b>5</b>
<b>LISTA DE TABELAS .....</b>	<b>7</b>
<b>1 INTRODUÇÃO .....</b>	<b>10</b>
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>13</b>
2.1 <i>Aedes aegypti</i> .....	13
2.2 Frequência do batimento das asas do mosquito .....	15
2.3 Espectrograma em escala mel .....	17
2.4 <i>Machine Learning</i> .....	17
2.4.1 Redes Neurais Artificiais .....	18
2.4.2 Redes Neurais Convolucionais .....	19
2.4.2.1 Função de ativação .....	20
<b>3 TRABALHOS RELACIONADOS .....</b>	<b>22</b>
3.1 Gravação de áudio de mosquitos .....	22
3.2 Classificação de áudio utilizando <i>Machine Learning</i> .....	22
3.2.1 Estratégia de classificação .....	23
<b>4 PROJETO DO APLICATIVO E TECNOLOGIAS UTILIZADAS.....</b>	<b>24</b>
<b>4.1 Android .....</b>	<b>24</b>
4.1.1 Arquitetura do sistema Android .....	24
4.1.2 Participação no mercado brasileiro .....	26
<b>4.2 TensorFlow Lite.....</b>	<b>27</b>
<b>4.3 Modelagem das funcionalidades.....</b>	<b>27</b>
4.3.1 Perfil do usuário .....	27
4.3.2 <i>User stories</i> .....	28
4.3.3 Jornada do usuário .....	28
<b>4.4 Soluções análogas .....</b>	<b>29</b>
4.4.1 Similares de função.....	30
4.4.2 Similares de produto .....	30
<b>4.5 Projeto da interface.....</b>	<b>31</b>
<b>4.6 Modelagem do armazenamento dos dados.....</b>	<b>31</b>
4.6.1 NoSQL .....	32
4.6.2 Firebase Cloud Firestore .....	32
<b>4.7 Modelagem da arquitetura.....</b>	<b>33</b>
<b>5 SOLUÇÃO DESENVOLVIDA .....</b>	<b>35</b>
<b>6 AVALIAÇÃO DO FUNCIONAMENTO .....</b>	<b>42</b>
6.1 Testes de detecção.....	43
<b>7 CONSIDERAÇÕES FINAIS .....</b>	<b>49</b>
7.1 Trabalhos futuros.....	49
<b>REFERÊNCIAS .....</b>	<b>51</b>

## 1 INTRODUÇÃO

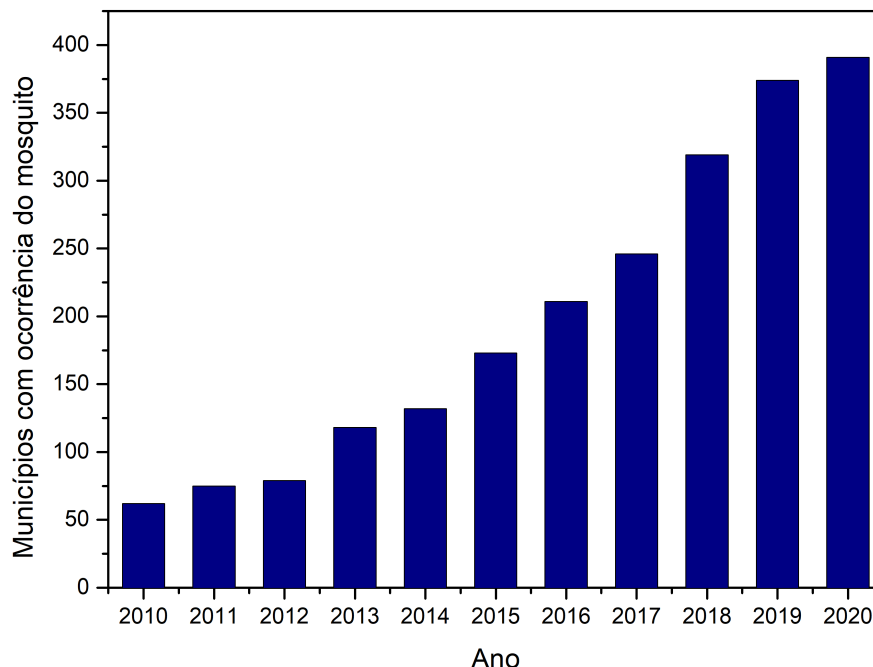
Desde o início de sua história, o Brasil sofre com uma série de doenças tropicais endêmicas. Dentre estas, podemos citar as transmitidas pelo mosquito *Aedes aegypti* como tendo grande importância na saúde pública brasileira. Dengue, chikungunya e zika são as três principais infecções transmitidas a humanos quase que exclusivamente por mosquitos dessa espécie. A zika ficou conhecida no ano de 2015 por, apesar de gerar sintomas relativamente leves, causar uma série de episódios de microcefalia em fetos gerados por mulheres infectadas durante a gestação. Este é um problema que incorrerá em um grande ônus para o sistema de saúde pública brasileiro por décadas. Por outro lado, estima-se que a dengue infectou mais de 800.000 pessoas no Brasil somente de janeiro a maio de 2020 (Ministério da Saúde, 2020). Esta doença, na sua forma hemorrágica, pode causar sintomas graves e levar à morte dos pacientes.

Apesar de ser historicamente concentrado nas regiões tropicais do país, o mosquito *A. aegypti* vem se espalhando com velocidade para as regiões mais austrais. O Rio Grande do Sul, particularmente, vem registrando um número crescente de focos deste mosquito (Centro Estadual de Vigilância em Saúde - RS, 2020). A Figura 1.1 mostra o número de municípios com focos confirmados do inseto. Nota-se uma tendência crescente no espalhamento geográfico do *A. aegypti*, o que gera séria preocupação aos gestores de saúde pública locais.

O método atual de identificação do mosquito consiste na observação das larvas presentes em armadilhas estrategicamente dispostas pela área urbana (Ministério da Saúde, 2001). Este sistema praticamente artesanal requer grande aporte de material, no caso das armadilhas, e de pessoal, já que é necessário o monitoramento individual das armadilhas pelos agentes de saúde. Entretanto, outras novas formas de monitoramento destes focos são atualmente consideradas. Dentre estas, encontra destaque a identificação a partir do zumbido emitido pelas asas do mosquito.

A presença ubíqua de *smartphones* – e de seus microfones – permitiria uma rede de detecção sonora que, através do mapeamento colaborativo, permitiria redirecionar recursos para otimizar as atuais estratégias que se utilizam de armadilhas manuais. Para atingir uma maior parcela dos usuários de *smartphone*, optou-se por desenvolver a aplicação para o sistema Android. Nota-se que, em julho de 2020 (último período para o qual obtivemos dados confiáveis), o Android estava presente em cerca de 86% dos dispositivos móveis brasileiros.

Figura 1.1 – Evolução do número de municípios do Rio Grande do Sul com registro de focos de *Aedes aegypti*.



Fonte: Do Autor. Dados do Centro Estadual de Vigilância em Saúde - RS (2020).

Para a correta identificação de um áudio, as técnicas de *machine learning* demonstram excelente aplicabilidade. Esta ferramenta nos permite detectar padrões sonoros que não seriam classificados facilmente a partir de algoritmos tradicionais. Com um treinamento apropriado, pode-se exportar um modelo contendo a descrição da rede neural e seus pesos associados. Este modelo pode então ser transferido para diferentes plataformas, como, por exemplo, para aplicativos móveis.

O objetivo do presente trabalho é desenvolver um aplicativo móvel de identificação sonora e mapeamento de mosquitos *Aedes aegypti* através da utilização de uma rede neural. Este aplicativo tem potencial para ser uma ferramenta valiosa no combate ao mosquito *A. aegypti* pois ela será capaz de agregar dados que irão auxiliar no direcionamento de recursos para o combate ao mosquito.

Este trabalho está organizado em sete seções, incluindo a presente introdução. O Capítulo 2 apresenta a fundamentação teórica necessária para o desenvolvimento do aplicativo. O Capítulo 3 apresenta um conjunto de trabalhos envolvendo o estudo de sons de mosquitos *A. aegypti* e a sua classificação utilizando *Machine Learning*. O Capítulo

4 apresentada alguns conceitos das tecnologias utilizadas e o projeto do aplicativo, contendo uma breve pesquisa sobre algumas soluções análogas que serviram de base para o planejamento das funcionalidades elaboradas e sua modelagem. O Capítulo 5 detalha a solução desenvolvida, descrevendo as suas principais telas. No Capítulo 6 são descritos os testes realizados com o aplicativo e os seus resultados. Por fim, o Capítulo 7 apresenta as considerações finais sobre este trabalho, bem como algumas sugestões de trabalhos futuros.



## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados os conceitos utilizados como base para o desenvolvimento deste trabalho.

### 2.1 *Aedes aegypti*

O mosquito *Aedes aegypti* é um dos principais vetores de uma série de doenças tropicais, dentre as quais destacam-se a febre amarela, dengue, zika e chikungunya. Desde sua chegada ao Brasil, no final do século XIX, o mosquito vem se tornando um grave problema de saúde pública. Na época, o mosquito era responsável apenas pela transmissão da febre amarela. Em 1955, o mosquito foi erradicado no Brasil. Entretanto, após o relaxamento das medidas adotadas, o mesmo foi reintroduzido em território nacional, espalhando-se para todas as regiões do país (Instituto Oswaldo Cruz, 2020). O mosquito mede menos de 1 centímetro de comprimento, e possui listras brancas e uma marca característica no tórax, em forma de lira. Ele possui hábitos preferencialmente diurnos (RUEDA, 2020). Um exemplo de espécime pode ser visto na Figura 2.1.

Figura 2.1 – Mosquito fêmea da espécie *Aedes aegypti*.



Fonte: Centers for Disease Control and Prevention (CDC), Estados Unidos. Domínio público.

Por ser o principal vetor de diversas doenças, o mosquito é foco de preocupação de governantes e gestores de saúde pública. Destaca-se aqui o seu papel na cadeia de transmissão da dengue. De acordo com o Boletim Epidemiológico do monitoramento dos

casos de arboviroses transmitidas pelo *A. aegypti*, apenas nos cinco primeiros meses de 2020 foram contabilizados cerca de 800 mil prováveis casos de dengue (Ministério da Saúde, 2020). A transmissão se inicia quando a fêmea da espécie vetora se contamina ao picar um indivíduo infectado que se encontra na fase virêmica da doença. Após um período de 10 a 14 dias, o mosquito torna-se capaz de transmitir o vírus por toda sua vida através de suas picadas (Ministério da Saúde, 2001). Além da dengue, cabe ressaltar o papel do *A. aegypti* como vetor da zika. Em 2015, o Brasil sofreu com uma epidemia desta doença que, apesar de inócua para a maioria das pessoas, leva à má-formação cerebral de fetos em gestantes infectadas. Apenas no estado de Pernambuco foram registrados 141 casos de microcefalia no mês de novembro de 2015, juntamente com um excesso no número de casos em outros estados do Nordeste (OLIVEIRA; da Costa Vasconcelos, 2016).

Segundo o manual de instruções para Pessoal de Combate ao Vetor, é necessário um trabalho extenso de mapeamento de focos de mosquito para auxiliar no seu combate. Neste sentido, são cruciais o levantamento de índice de infestação<sup>1</sup> através de pesquisa larvária, a inspeção de pontos estratégicos, a pesquisa em armadilhas e a procura eventual de *A. aegypti* em função da denúncia da presença do mosquito em áreas supostamente não infestadas (Ministério da Saúde, 2001).

Uma das etapas necessárias para o mapeamento de focos do mosquito é a pesquisa em armadilhas de oviposição. As armadilhas são depósitos com água colocados em localidades negativas para a ocorrência do mosquito, a fim de atrair as fêmeas para a postura dos ovos. As armadilhas são divididas em ovitrampas, onde são depositados os ovos do mosquito, e larvitrapas, que coletam ovos, larvas, pupas e exúvias do mosquito. É necessária a pesquisa semanal nas armadilhas, pois estas podem se tornar um criadouro de mosquitos caso o período de verificação seja ampliado ou interrompido. O mosquito possui uma capacidade de dispersão pelo voo inferior a 100 metros. A fêmea do mosquito costuma passar toda a sua vida nas proximidades do local de onde eclodiu, sob condição de existência de hospedeiros próximos. Portanto, o mapeamento do mosquito em uma área pode indicar a presença de um foco adjacente. O Ministério da Saúde recomenda a pesquisa de focos em um raio de 300 metros a partir do foco inicial.

O processo de verificação das armadilhas é artesanal e trabalhoso (Figura 2.2), visto que os agentes de saúde precisam se deslocar a todas as armadilhas no período de uma semana, realizar a coleta de material, a limpeza e a manutenção dos dispositivos.

---

<sup>1</sup>O índice de infestação reflete o percentual de imóveis positivos para a presença do *A. aegypti*. É estimado pela razão entre o número de imóveis positivos e o número de imóveis pesquisados.

Além disso, temos o custo de implantação das armadilhas, o custo de manutenção do sistema (salário dos agentes de saúde), e a possibilidade de erro humano na classificação das ocorrências.

Figura 2.2 – Agente de saúde realizando pesquisa em armadilha (larvitampa).



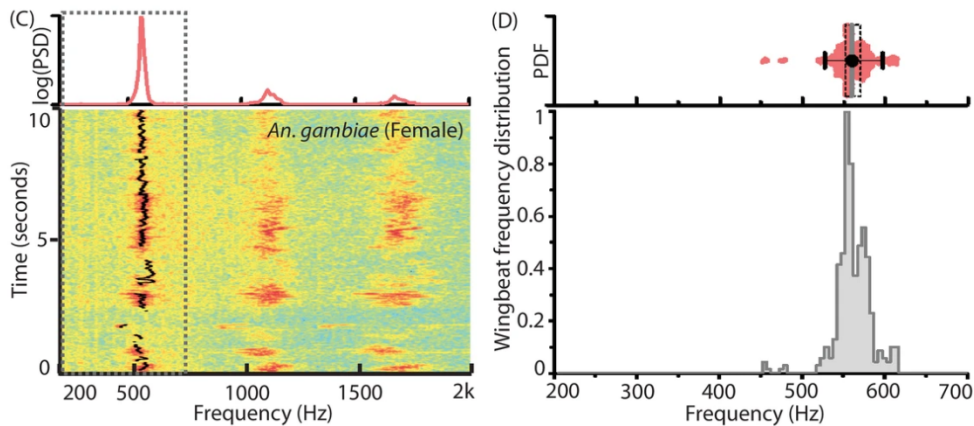
Fonte: Dengue, instruções para pessoal de combate ao vetor: manual de normas técnicas. - 3. ed., rev. - Brasília, Ministério da Saúde, Fundação Nacional de Saúde, 2001.

## 2.2 Frequência do batimento das asas do mosquito

Uma outra forma possível de identificação de mosquitos que não é baseada nos aspectos visuais é a sua classificação através da frequência do batimento de suas asas. Durante o voo, as diferentes espécies de mosquito batem as asas com frequências específicas. O intervalo de frequência do batimento das asas está compreendido entre 200 e 700 Hz (MUKUNDARAJAN et al., 2017). A frequência não é monótona e é virtualmente constante durante o voo. Um exemplo pode ser visto na Figura 2.3, que mostra o espectro sonoro das asas de uma fêmea do mosquito *Anopheles gambiae*. À esquerda na figura, podemos ver a constância do sinal emitido, bem como as harmônicas contidas no som. À

direita, observa-se a distribuição do espectro, demonstrando a largura do sinal principal.

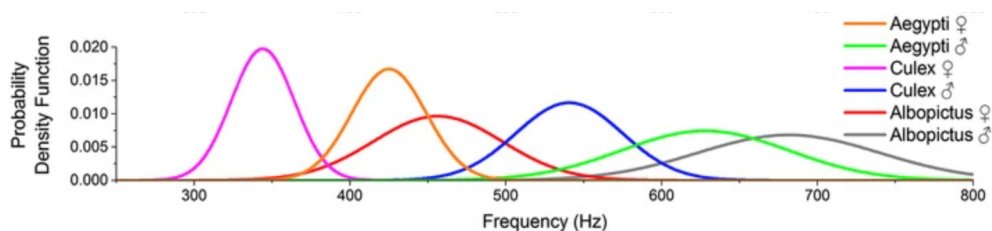
Figura 2.3 – Espectrograma do voo de um mosquito *Anopheles gambiae* capturado com um dispositivo móvel e sua distribuição de frequências.



Fonte: Adaptado de Mukundarajan et al. (2017).

Um aspecto interessante do batimento das asas do mosquito é que a seleção sexual levou à diferenciação deste som em diferentes espécies e também entre os sexos de uma mesma espécie. A Figura 2.4 demonstra esse fenômeno para as três espécies de mosquitos mais comumente encontradas no território brasileiro: *Aedes aegypti*, *Culex sp.* e *Aedes albopictus*. A variação sexual também pode ser claramente observada. Intraespecificamente, as fêmeas emitem frequências mais graves que os machos (GENOUD et al., 2018). De acordo com Genoud *et al.*, as frequências são de  $628 \pm 6$  Hz e  $425 \pm 2$  Hz para os machos e as fêmeas de *A. aegypti*, respectivamente.

Figura 2.4 – Distribuição de frequências do batimento de asas de mosquito capturadas com sensor óptico.

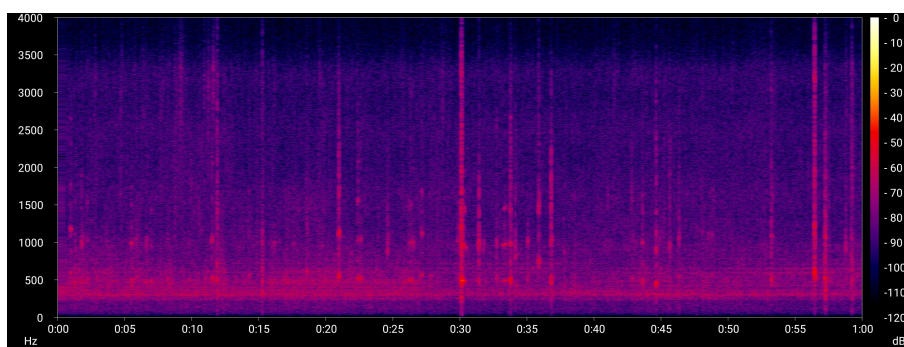


Fonte: Adaptado de Genoud et al. (2018).

## 2.3 Espectrograma em escala mel

Um espectrograma é uma representação visual da intensidade do espectro de frequências de um sinal conforme ele varia ao longo do tempo. Um exemplo de espectrograma do voo de um mosquito pode ser visto na Figura 2.5. Podemos ver que o áudio emitido pelo mosquito está situado próximo à frequência de 440 Hz, pois o espectrograma apresenta uma cor mais intensa nesta faixa.

Figura 2.5 – Espectrograma do voo de mosquito *A. aegypti*.



Fonte: Do Autor.

No âmbito de *Machine Learning*, é comum utilizarmos espectrogramas em escala mel para a classificação de áudios (DIELEMAN; SCHRAUWEN, 2014), pois esta se assemelha à percepção humana, sendo mais discriminativa nas frequências mais baixas e menos discriminativa nas frequências mais altas (STEVENS; VOLKMANN, 1940). A escala mel é subjetiva, e seu nome vem do inglês *melody*, indicando que a escala é baseada na altura do som. Nesta escala, toma-se como ponto de referência inicial o valor de 1000 Hz, que equivalem a 1000 *mels*. A partir disso, Stevens and Volkman (1940) realizaram um estudo onde 5 observadores externos classificaram 10 frequências diferentes, e a partir disso foi construída uma escala numérica proporcional à magnitude da altura percebida. O'Shaughnessy (2000) definiu a conversão de Hertz para mel através da Equação 2.1.

$$mel = 2595 \log_{10}(1 + f/700) \quad (2.1)$$

## 2.4 Machine Learning

De acordo com o dicionário *Aurélio*, um algoritmo é um conjunto de regras e operações e procedimentos definidos e ordenados, usados na solução de um problema ou

de classe de problemas, em um número finito de etapas. Para determinados problemas, pode não existir um algoritmo que os resolva, ou então a sua complexidade pode ser tamanha que ainda não exista um equipamento capaz de executar tal algoritmo. Portanto, são necessárias formas de resolução destes problemas que fujam deste tradicional modelo de resolução.

*Machile Learning* (ML) – ou, em português, Aprendizado de Máquina – pode ser definido como a capacidade de um programa qualquer de computador de melhorar seu desempenho em uma tarefa através da experiência (MITCHELL, 1997). Segundo Alpaydin (2010), utilizamos o aprendizado de máquina para otimizar os parâmetros de um modelo pré-definido utilizando dados de treinamento ou experiências passadas. Existem três classes de algoritmos aplicados em ML:

- **Aprendizado por reforço:** O algoritmo de aprendizado tenta aprender qual é a melhor ação a ser tomada dentro de um contexto baseado no feedback, positivo ou negativo, de ações passadas.
- **Aprendizado não supervisionado:** Parte de um conjunto de dados não rotulados, onde o algoritmo busca identificar nestes regularidades que permitam realizar a sua classificação.
- **Aprendizado supervisionado:** Utiliza um conjunto de dados de treinamento contendo respostas rotuladas, ou classes, e então o algoritmo aprende padrões a partir destes. A partir dos padrões aprendidos pelo algoritmo, ele deve ser capaz de formular regras e aplicá-las para identificar novas instâncias da classe de dados de treinamento.

Uma forma particular de sistemas que utilizam o aprendizado supervisionado como forma de treinamento são as Redes Neurais Artificiais.

#### **2.4.1 Redes Neurais Artificiais**

Uma Rede Neural Artificial (RNA) é um sistema vagamente inspirado por neurônios reais. Ela é baseada em uma coleção de nodos interconectados, que representam neurônios, e, em cada conexão, transmite dados entre os elementos da rede. Em cada neurônio, estes dados são processados através de uma soma ponderada de todas as suas entradas. Estes neurônios são divididos em camadas, devendo conter obrigatoriamente uma camada de entrada e uma camada de saída, e entre estas, podem existir diversas ca-

madas intermediárias. Cada neurônio de uma camada é totalmente conectado aos neurônios da próxima e da anterior.

O treinamento destas redes ocorre através do processamento de conjuntos de dados que contêm entrada e resultado conhecidos, formando associações com probabilidade ponderada entre os dois, que são armazenados na estrutura interna da rede. Sucessivos treinamentos ocorrem onde a rede ajusta esses valores, de modo que a sua saída seja sucessivamente mais parecida com a saída esperada.

Redes neurais artificiais são adequadas para algumas classes de problemas onde os dados de treinamento são complexos e ruidosos, como dados oriundos de câmeras e microfones (MITCHELL, 1997).

#### 2.4.2 Redes Neurais Convolucionais

Uma classe de redes neurais que tem se popularizado nos últimos anos é a de Redes Neurais Convolucionais (do inglês *Convolutional Neural Network*). Esta classe é especializada em processar dados que possuem uma topologia em forma de rede, como, por exemplo, imagens (GOODFELLOW; BENGIO; COURVILLE, 2016). Os neurônios deste tipo de rede neural não são totalmente conectados entre as camadas; os neurônios de uma camada se conectam apenas aos neurônios de parte de uma camada anterior. O termo convolucional indica que a rede emprega uma operação matemática de convolução no lugar de multiplicações de matrizes em pelo menos uma de suas camadas. Um exemplo da operação de convolução pode ser visto na Figura 2.6. Nesta operação, foi utilizado um *kernel*<sup>2</sup> 2 x 2 em uma imagem 4 x 3.

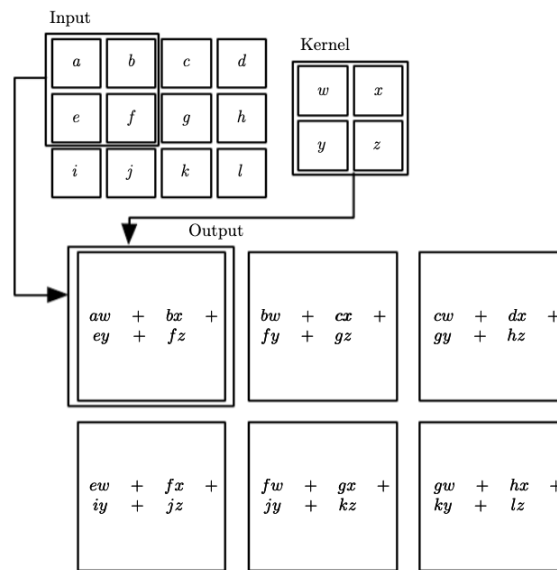
As camadas de uma rede neural convolucional possuem comportamentos distintos das camadas de uma rede neural artificial. Entre os tipos de camadas utilizadas, as mais comuns são:

- Camada convolucional: esta camada é composta por um conjunto de *kernels* que realizam a convolução na imagem de entrada, obtendo um mapa de características que corresponde à resposta do *kernel* sobre todas as regiões da imagem. Isto faz com que a rede neural reconheça padrões locais do mesmo tamanho que o *kernel* utilizado, e a partir disso é possível reconhecer estes mesmos padrões em outras

---

<sup>2</sup>Um *kernel* é uma matriz que atua como filtro, utilizado para extrair características, ou features, de uma imagem. Para cada ponto da imagem de entrada, é realizado o produto escalar naquele ponto com o *kernel*, obtendo-se como saída uma matriz de produtos escalares.

Figura 2.6 – Exemplo da operação de convolução.



Fonte: Goodfellow, Bengio and Courville (2016).

áreas da imagem a ser analisada. Camadas subsequentes conseguem reconhecer combinações de padrões reconhecidos em camadas anteriores. Por exemplo: a primeira camada convolutiva reconhece padrões simples, como arestas e cores. Uma camada subsequente consegue realizar combinações destas arestas e cores e reconhecer texturas na imagem. Uma terceira camada pode combinar essas texturas e reconhecer elementos mais complexos encontrados em imagens reais, como objetos (CHOLLET, 2017).

- Camada de *pooling*: esta camada é geralmente utilizada imediatamente após as camadas convolucionais e tem como objetivo simplificar as informações da camada anterior, a fim de reduzir a complexidade para camadas posteriores (ALBAWI; MOHAMMED; AL-ZAWI, 2017).
- Camada de *dropout*: *dropout* é uma técnica de regularização muito efetiva e frequentemente utilizada em redes neurais (CHOLLET, 2017). Consiste em eliminar aleatoriamente um número de saídas de uma camada durante o treinamento, com o intuito de inserir ruído para que a rede quebre padrões que não sejam significativos.

#### 2.4.2.1 Função de ativação

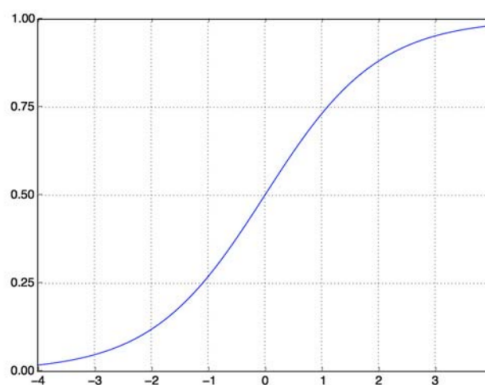
Após realizar a soma ponderada de suas entradas, os neurônios de uma rede neural aplicam uma função de ativação. Essa função de ativação permite que pequenas mudanças



nos pesos dos neurônios causem apenas uma pequena alteração na saída destes. Este fato permite que uma rede neural aprenda; caso contrário, ela seria apenas uma série de transformações lineares incapazes de atuar sobre dados complexos (CHOLLET, 2017). Entre as funções de ativação mais utilizadas em literatura, podemos citar as seguintes:

- Função Sigmóide: esta função varia entre 0 e 1, e possui o formato de S. Podemos ver um exemplo desta Figura 2.7. Ela força os valores de Y para os extremos, sendo esta uma qualidade desejável ao tentar realizar a classificação para uma classe específica.

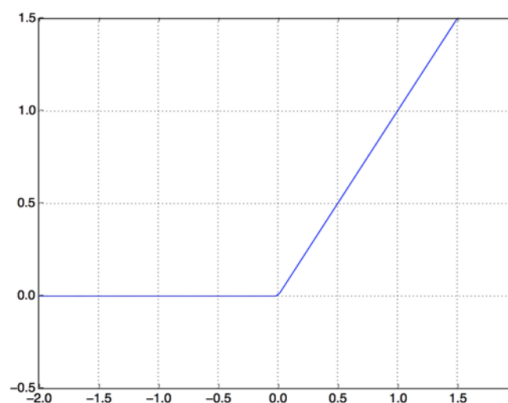
Figura 2.7 – Função sigmóide.



Fonte: Chollet (2017)

- Função ReLU: Do inglês *Rectified Linear Unit*, é uma função de baixo custo computacional que limita valores menores que zero em zero (ALBAWI; MOHAMMED; AL-ZAWI, 2017). Um exemplo da função pode ser visto na Figura 2.8.

Figura 2.8 – Função ReLU.



Fonte: Chollet (2017)

### 3 TRABALHOS RELACIONADOS

Este capítulo apresenta trabalhos relacionados com o tema abordado.

#### 3.1 Gravação de áudio de mosquitos

Em Staunton et al. (2019), os autores propõem uma nova metodologia para detectar alterações nas frequências do bater de asas de machos e fêmeas da espécie *A. aegypti*. Os autores realizaram a gravação de áudios em alta resolução dos mosquitos, juntamente com alguns metadados como data, hora e localização.

No trabalho de Arthur et al. (2014), é realizada uma análise detalhada do bater de asas de machos e fêmeas do mosquito *A. aegypti* utilizando gravações realizadas por um conjunto de sensores acústicos. Os microfones foram dispostos a uma distância de até 19 cm dos mosquitos em vários experimentos. Cator et al. (2011) realizam a gravação do áudio do mosquito *A. aegypti* em campo, observando suas frequências e padrões comportamentais.

Por fim, Mukundarajan et al. (2017) demonstram que é possível realizar a captura do áudio do batimento das asas de mosquitos utilizando o microfone de *smartphones*, desde que o mosquito esteja a 10 cm de distância do aparelho. Além disso, é demonstrado que a classificação das diferentes espécies de mosquitos é factível, pois elas possuem variações perceptíveis. Essa variação interespecífica do zumbido dos mosquitos pode ser utilizada como uma ferramenta poderosa na substituição dos métodos atuais de identificação de mosquitos, crucialmente no caso do *A. aegypti*.

#### 3.2 Classificação de áudio utilizando *Machine Learning*

Já foi descrita em literatura uma série de instâncias onde é realizada a classificação de áudio utilizando *machine learning*. Em Moore (1991) foi criada uma rede neural capaz de identificar o voo de mosquitos utilizando sensores ópticos que gravavam o batimento de asas. Seguindo os mesmos passos, Zhang and Luo (2010) conseguiram realizar a classificação de cinco espécies diferentes de mosquitos. Por fim, Fernandes, Cordeiro and Recamonde-Mendoza (2020) realizam a classificação do *A. aegypti* utilizando um espectrograma em escala mel do voo do mosquito e uma rede neural convolucional.

### 3.2.1 Estratégia de classificação

Uma das estratégias de classificação adotada no trabalho de Fernandes, Cordeiro and Recamonde-Mendoza (2020) foi a utilização de um classificador binário. Este classificador foi treinado para distinguir entre duas classes de mosquito, onde uma das classes é o *A. aegypti* e a outra classe é um grupo representando as outras 23 espécies de mosquito. Ou seja, o classificador identifica se o mosquito é *A. aegypti* ou não. Como *feature*, usou-se o espectrograma das gravações, de forma que a frequência de voo do mosquito seja representada visualmente ao longo do tempo.

A arquitetura da rede neural do classificador binário é composta de duas camadas convolucionais, duas camadas de *pooling*, uma camada de *dropout* e duas camadas totalmente conectadas. A primeira camada é composta por 32 filtros com um *kernel* 3 x 3 e utiliza uma função de ativação ReLU. A segunda e quinta camadas são camadas de *pooling* de tamanho 2 x 2 e *stride* 1. A terceira camada é uma camada de *dropout*, com taxa definida em 0,2. Por fim, as duas camadas finais são totalmente conectadas e compostas na seguinte ordem: a primeira contém 1334 neurônios utilizando a função de ativação ReLU. A última camada da rede (camada de classificação) é composta de dois neurônios de saída que utilizam uma função de ativação sigmóide.

O classificador binário apresentou um desempenho satisfatório, com uma acurácia de  $97,78 \pm 0,73\%$ . Portanto, optou-se pela escolha deste como base para realização do trabalho.

Os estudos previamente citados mostraram que é possível realizar a análise do áudio dos mosquitos utilizando *machine learning*. Porém, este tipo de trabalho ainda não foi implementado em uma plataforma móvel que permita a sua utilização em campo. Mukundarajan et al. (2017) demonstraram que é possível realizar a gravação do áudio do mosquito com qualidade satisfatória utilizando o microfone de *smartphones*. Logo, há evidências de que seria possível criar um sistema ubíquo de monitoramento de *A. aegypti* utilizando os *smartphones* da população.

## 4 PROJETO DO APLICATIVO E TECNOLOGIAS UTILIZADAS

Este capítulo trata da modelagem utilizada na elaboração do aplicativo proposto e das tecnologias utilizadas durante o seu desenvolvimento.

### 4.1 Android

O Android é uma plataforma de software de código aberto baseada em uma versão modificada do Linux, criado em 2003 pela empresa Android Inc. Originalmente concebido para melhorar a experiência do usuário em câmeras digitais, o projeto acabou sendo reorientado para o mercado de dispositivos móveis. Em 2005, a empresa foi adquirida pelo Google e, em 2008, o sistema operacional foi lançado comercialmente ao público (CALLAHAM, 2020).

#### 4.1.1 Arquitetura do sistema Android

A arquitetura da plataforma é composta de uma pilha dividida em cinco camadas, contendo seis elementos principais. Um esquema da arquitetura da plataforma Android pode ser visto na Figura 4.1. As diferentes camadas serão discutidas a seguir.

- *Kernel Linux*

O kernel do Linux é responsável pelas funcionalidades básicas do sistema operacional, como gerenciamento de memória e processos.

- *Camada de abstração de hardware*

Camada responsável pela comunicação entre o hardware e o *framework* Java, consistindo de diversos módulos e bibliotecas que implementam uma interface para cada elemento do hardware do dispositivo, como a câmera ou o Bluetooth.

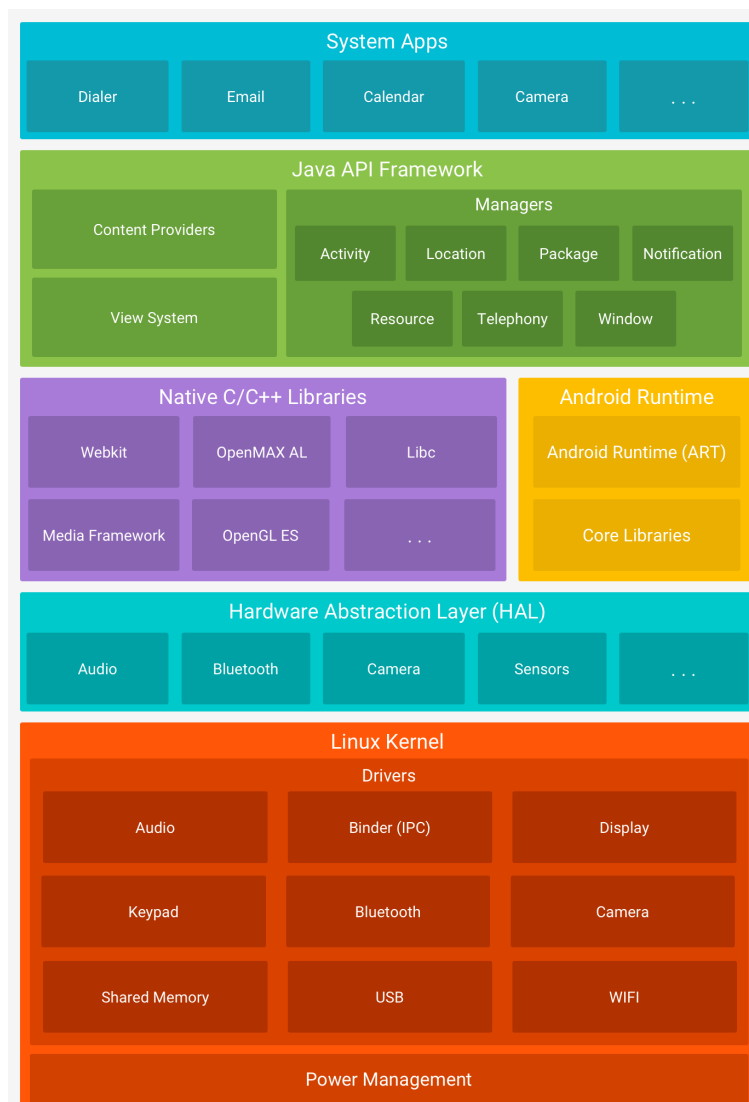
- *Bibliotecas nativas*

Esta camada expõe os elementos do núcleo de componentes e serviços do Android que são desenvolvidos utilizando C e C++, como, por exemplo, o Android Runtime (ART) e o OpenGL. Suas funcionalidades são expostas através de uma Interface de Programação de Aplicação<sup>1</sup> Java.

---

<sup>1</sup>Do inglês *Application Programming Interface* (API).

Figura 4.1 – Componentes da plataforma Android.



Fonte: <https://developer.android.com/guide/platform>. Acessado em: 09/11/2020.

- *Android Runtime*

O Android Runtime é o ambiente de execução utilizado desde a versão 5.0, substituindo a máquina virtual Dalvik. Ele foi projetado para executar múltiplas máquinas virtuais em dispositivos com pouca memória através da execução de um *bytecode* otimizado para reduzir o consumo de memória. Suas principais características são compilação ahead-of-time (AOT) e just-in-time (JIT), um coletor de lixo aprimorado e melhores ferramentas de diagnóstico.

- *Framework Java*

O *Framework Java* é responsável por disponibilizar aos desenvolvedores as APIs de desenvolvimento necessárias para a construção de aplicações, simplificando o reuso

de componentes modulares do sistema, como elementos de interface, notificações e pilha de navegação.

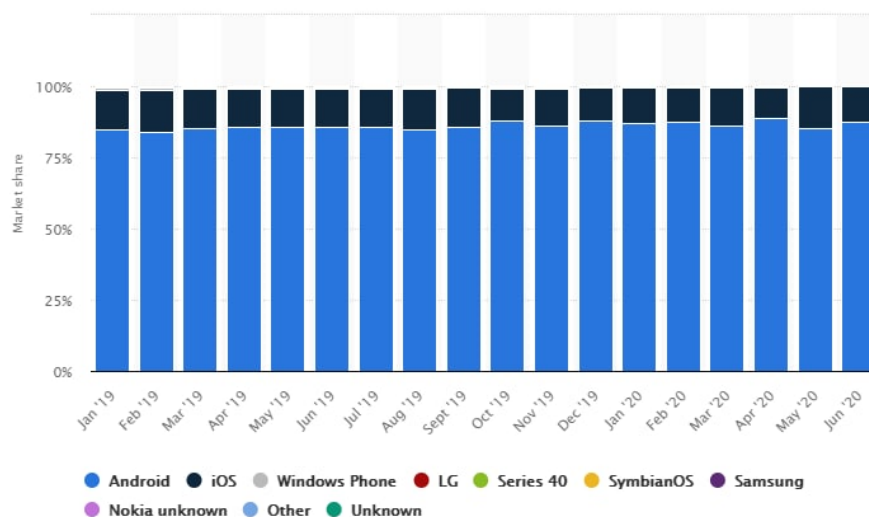
- *Aplicações do sistema*

A camada superior consiste nas aplicações disponibilizadas ao usuário, como email, calendário e browser. Estas aplicações podem ser substituídas pelo usuário, caso o mesmo deseje. Além disso, os desenvolvedores também podem utilizar estas aplicações para fornecer algumas funcionalidades específicas, como o envio de mensagens.

#### 4.1.2 Participação no mercado brasileiro

Após chegar ao Brasil em 2009, o sistema acabou se tornando o mais popular entre os dispositivos móveis. Isto se deve ao comprometimento do Google em manter o projeto com o código aberto, o que causou uma intensa adoção do sistema por diversas fabricantes. Tais fabricantes conseguem projetar dispositivos de diversos níveis de custo, desde celulares simples de entrada (*low-end*) a celulares topo-de-linha (*high-end*).

Figura 4.2 – Distribuição de mercado de sistemas operacionais móveis no Brasil.



Fonte: <https://www.statista.com/statistics/262167/market-share-held-by-mobile-operating-systems-in-brazil/>. Acessado em: 09/11/2020.

A Figura 4.2 mostra a distribuição de mercado dos sistemas operacionais utilizados nos dispositivos móveis brasileiros entre janeiro de 2019 e junho de 2020. Nota-se

que, em julho de 2020 (último período para o qual obtivemos dados confiáveis), o Android estava presente em cerca de 86% dos dispositivos móveis brasileiros. Também pode-se observar através da figura que a presença do Android se manteve estável, sem grandes perdas ou ganhos de mercado. Seu concorrente mais próximo, o iOS da Apple, está presente em uma fração muito menor de dispositivos. Assim, para atingir uma maior parcela de mercado no Brasil, é necessário que o desenvolvimento de novos projetos seja feito para o sistema Android.

## 4.2 TensorFlow Lite

O TensorFlow (ABADI et al., 2016) é uma biblioteca de software aberto criada para descrever, implementar e executar algoritmos de aprendizado de máquina. Um modelo nele descrito pode ser executado em diversos tipos de dispositivos, desde dispositivos móveis até grandes sistemas distribuídos, realizando poucas ou nenhuma mudança. Uma versão compatível com dispositivos móveis chamada TensorFlow Lite<sup>2</sup> foi disponibilizada para os sistemas Android, iOS<sup>3</sup> e microcontroladores.

Para ser utilizado no ambiente móvel, o modelo empregue por Fernandes, Cordeiro and Recamonde-Mendoza (2020), que foi criado sobre o TensorFlow, teve de ser convertido para um formato compatível com o TensorFlow Lite. É importante salientar que a rede neural é executada no smartphone; o consumo energético e o consumo de memória estão sendo avaliados em um trabalho paralelo do mesmo grupo de pesquisa.

## 4.3 Modelagem das funcionalidades

### 4.3.1 Perfil do usuário

Durante o desenvolvimento desta aplicação, imaginamos os seguintes perfis de usuário:

- Um usuário residente de áreas onde são encontrados mosquitos, sendo estes da espécie *A. aegypti* ou não.
- Um usuário que tem interesse em visitar alguma área onde a presença de mosquitos

---

<sup>2</sup>Disponível em <https://www.tensorflow.org/lite>.

<sup>3</sup>iOS é o sistema operacional de dispositivos móveis da Apple. Mais informações em <https://www.apple.com/ios>.

*A. aegypti* já é conhecida, e gostaria de verificar a magnitude de sua ocorrência.

- Um usuário membro de órgão de saúde público que gostaria de verificar os dados disponíveis da ocorrência de *A. aegypti* a fim de otimizar as políticas públicas aplicadas.

#### 4.3.2 *User stories*

Para realizar a modelagem das funcionalidades do aplicativo, foi decidida a utilização de *user stories*. Segundo Cohn (2004), uma *user story* descreve uma funcionalidade que será valiosa para o usuário do sistema. O seu formato segue o estilo *Eu como um <papel> quero <função> para que <motivo>*. As *user stories* definidas para o desenvolvimento deste trabalho foram construídas com base na jornada que o usuário tomaria em nosso aplicativo e são mostradas na Tabela 4.1.

Tabela 4.1 – *User stories* do aplicativo.

<i>Como um usuário</i>
<i>eu quero visualizar as ocorrências de A. aegypti ao meu redor.</i>

<i>Como um usuário</i>
<i>eu quero saber se um mosquito próximo a mim é A. aegypti</i>

<i>Como um usuário</i>
<i>eu quero relatar uma ocorrência de A. aegypti</i>

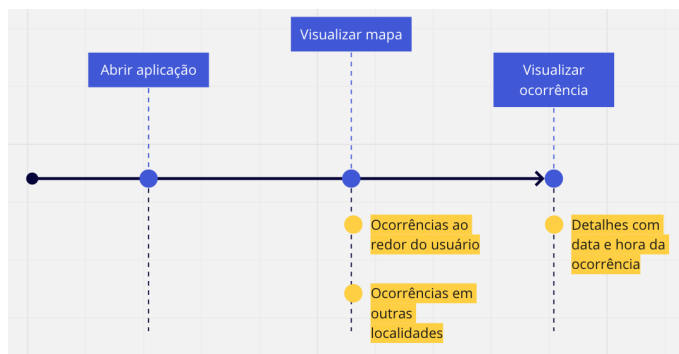
#### 4.3.3 Jornada do usuário

O termo "jornada do usuário" tem sido amplamente adotado no projeto de aplicações. Ele aborda os aspectos processuais e experimentais de serviços vistos pelo ponto de vista do usuário (FØLSTAD; KVALE, 2018). Com base nas *user stories*, extraímos as seguintes jornadas de usuário:

Na Figura 4.3 podemos ver a sequência de passos que o usuário deve tomar para visualizar as ocorrências de *A. aegypti* no mapa.

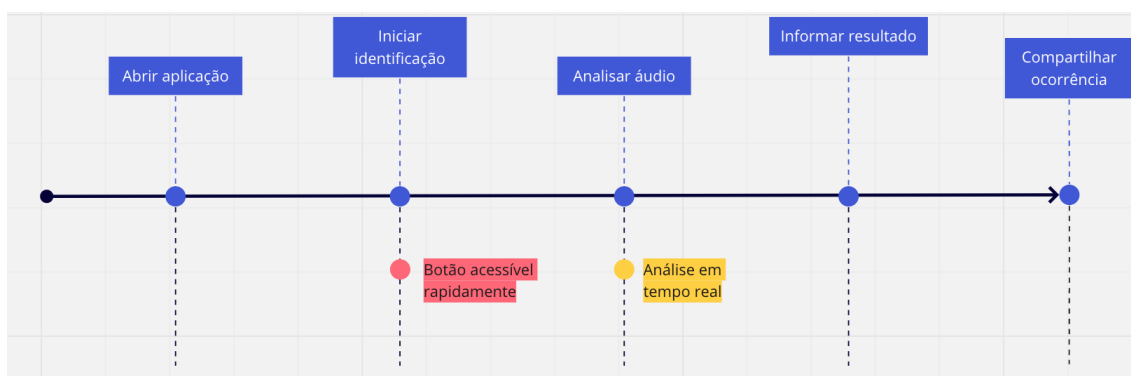


Figura 4.3 – Jornada de usuário 1: visualização do mapa.



Fonte: Do Autor.

Figura 4.4 – Jornada de usuário 2: identificação e compartilhamento de ocorrência.



Fonte: Do Autor.

Na Figura 4.4 é indicado o fluxo de ações que o usuário realiza para identificar o áudio do mosquito e relatar a sua ocorrência no mapa.

#### 4.4 Soluções análogas

Durante o desenvolvimento de um produto é muito importante analisar as alternativas já existentes no mercado que cumpram parcial ou totalmente as funcionalidades buscadas. Neste trabalho analisamos duas categorias de produtos análogos, similares de função e similares de produto.

#### 4.4.1 Similares de função

Na área de Design de Produto, um produto similar de função é aquele que compartilha a função principal do produto a ser desenvolvido, mesmo que não sejam destinados à mesma aplicação (PLATCHECK, 2003). Nesta categoria, dois produtos foram selecionados no presente projeto: o Shazam e o SoundHound.

O Shazam é uma aplicação móvel que realiza a identificação de músicas, filmes, propagandas e programas de televisão através da análise de áudio capturada pelo dispositivo. Originalmente lançado como um serviço por telefone, em 2006 foi disponibilizado nas principais lojas de aplicativos do mercado. Em funcionamento, o aplicativo analisa continuamente o áudio até encontrar em sua base uma ocorrência para o som captado.

Assim como o Shazam, o SoundHound também realiza a identificação de músicas, além de ser um processador de linguagem natural. Sua estreia no mercado deu-se em 2009, com o nome *Midomi*.

#### 4.4.2 Similares de produto

Os similares de produto englobam aqueles que compartilham as mesmas funções, características e atendem parcial ou totalmente aos requisitos de projeto elencados para o produto em desenvolvimento (PLATCHECK, 2003). Neste trabalho, elencamos três similares de produto relacionados ao objetivo deste projeto.

*Onde está o aedes* é um website disponibilizado pela Secretaria Municipal da Saúde de Porto Alegre (SMS) que possui um mapa contendo as ocorrências de *A. aegypti* nas armadilhas de mosquito espalhadas pela cidade. Os dados são atualizados conforme os agentes de saúde realizam a verificação das armadilhas.

O Abuzz é um aplicativo que está em desenvolvimento pela Universidade de Stanford, proposto por Mukundarajan e colaboradores em 2017. Com ele, o usuário pode gravar os sons de mosquitos e enviar os dados para uma base centralizada, onde estes são analisados, classificados e disponibilizados em um mapa interativo. Até o presente momento, a aplicação não foi disponibilizada para o público.

O aplicativo Sem Dengue permite realizar denúncias de focos de dengue em uma base de dados que é consultada por diversos órgãos de saúde municipais do Brasil. Também é possível acompanhar o estado da denúncia, obter informações sobre a dengue e visualizar as denúncias em um mapa colaborativo.

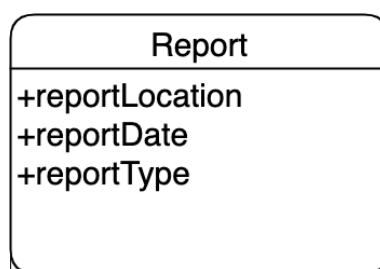
## 4.5 Projeto da interface

A partir do conjunto de funcionalidades modelados através das *user stories* e das jornadas de usuário projetamos uma versão inicial da aplicação contendo duas áreas principais: a área de análise de áudio, contendo um grande botão para iniciar o processo de identificação do mosquito; e uma área contendo o mapa das ocorrências. Durante o processo de desenvolvimento da aplicação, notamos que a alternância entre as duas áreas principais poderia prejudicar a usabilidade da aplicação. Portanto, decidimos unir as duas áreas em uma única, exibindo o mapa de ocorrências e o botão de análise na mesma tela.

## 4.6 Modelagem do armazenamento dos dados

Através das *user stories* percebemos que o único tipo de dado que teríamos de armazenar seria a ocorrência de *A. aegypti* realizada pelo usuário. Com base nestes requisitos, chegamos ao modelo descrito na Figura 4.5.

Figura 4.5 – Diagrama do modelo de dados de uma denúncia.



Fonte: Do Autor.

Armazenou-se a localização estimada da ocorrência, sua data e o tipo de detecção, com o valor 1 indicando a detecção por gravação e 2 o carregamento prévio do áudio. Como o modelo de dados é muito simples e as denúncias não possuem nenhuma relação umas com as outras, decidiu-se pela escolha de um banco de dados não relacional. Além disso, as ocorrências deveriam ser armazenadas em um servidor remoto para que pudessem ser acessadas pelos outros usuários do aplicativo.

### 4.6.1 NoSQL

Os tradicionais bancos de dados relacionais não conseguiram acompanhar a crescente complexidade dos sistemas computacionais em termos de escalabilidade horizontal. Com isso, foi necessário projetar novos sistemas de banco de dados com esta característica. Entre eles, estão os sistemas chamados *NoSQL*<sup>4</sup>. De acordo com Cattell (2010), um banco de dados *NoSQL* tem as seguintes características:

- Escalabilidade de operações simples em diversos servidores;
- Replicação e particionamento de dados;
- Chamadas simples;
- Modelo simples de concorrência;
- Indexação eficiente;
- Capacidade de adicionar dinamicamente novos atributos aos registros de dados.

Apesar da simplicidade do modelo de dados do projeto proposto, considera-se que uma solução escalável seria necessária. Em um cenário futuro, espera-se que o número de ocorrências a serem armazenadas tornar-se-á extenso, possibilitando realizar o acompanhamento do histórico de infestações do mosquito em diversas regiões do Brasil. Além disso, um grande número de usuários poderá acessar a nossa base de dados e consultar os focos de *A. aegypti*; logo, optou-se por uma solução que utiliza *NoSQL*.

### 4.6.2 Firebase Cloud Firestore

Dentre os serviços disponíveis para armazenamento, decidiu-se utilizar o *Firebase Cloud Firestore*<sup>5</sup>, um serviço de banco de dados *NoSQL* hospedado na nuvem, disponibilizado pelo Google. Projetado para ser utilizado por aplicativos móveis por meio de kits de desenvolvimento de software nativos, este serviço é um banco de dados orientado a documentos, onde estes são armazenados em coleções, que servem para organizar os dados e possibilitar consultas. Além de tipos primitivos de dados, os documentos podem conter subcoleções, permitindo a criação de estruturas de dados hierárquicas e escalonáveis.

As consultas ao banco permitem a recuperação dos dados através de métodos ou

---

<sup>4</sup>Segundo Cattell (2010) o nome *NoSQL* vem do inglês para "Não Apenas", ou então "Não Relacional".

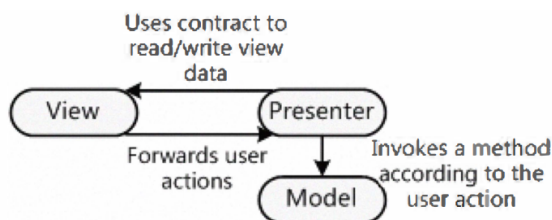
<sup>5</sup>Disponível em <https://firebase.google.com/docs/firestore>.

de *listeners*<sup>6</sup>. Também é oferecido suporte offline para que a aplicação possa funcionar mesmo quando não há conexão com a Internet, armazenando os dados em cache até que a conexão seja reestabelecida.

#### 4.7 Modelagem da arquitetura

Um ponto essencial durante a modelagem da aplicação foi manter uma baixa dependência dos serviços de hospedagem de dados do Google, pois os mesmos passam a cobrar taxas após atingir-se uma determinada quota<sup>7</sup>. Dentre as arquiteturas mais populares para o desenvolvimento de aplicações móveis, decidiu-se pelo uso do padrão Model-View-Presenter (MVP). Desenvolvido por uma subsidiária da IBM nos anos 90 (POTEL, 1996), o MVP é uma generalização do padrão Model-View-Controller (KRASNER; POPE, 1988) e oferece uma metodologia de design de fácil compreensão para uma ampla gama de aplicações. Segundo Zhang and Luo (2010), um exemplo da arquitetura pode ser visto na Figura 4.6.

Figura 4.6 – Exemplo do padrão MVP.



Fonte: (ZHANG; LUO, 2010).

Seus elementos são os seguintes:

- *Model*: componente responsável pela persistência de dados, estado e lógica de negócio.
- *View*: é a interface encarregada de receber a entrada do usuário e exibir os resultados.
- *Presenter*: recebe as entradas do usuário através da *View* e encaminha os respectivos

<sup>6</sup>Um *listener* é um objeto que é notificado após a ocorrência de um evento. No caso do *Cloud Firestore*, o aplicativo recebe uma cópia dos dados sempre que ocorre uma atualização na base remota.

<sup>7</sup>Para o nível gratuito, são disponibilizadas diariamente 50.000 operações de leitura e 20.000 operações de escrita. Após este limite, é cobrada uma taxa de 18 centavos de dólar a cada 100.000 operações. Mais informações em <https://firebase.google.com/pricing>.

comandos para o *Model*. Após receber os resultados do *Model*, é responsável por atualizar a *View*.

Uma das principais vantagens do MVP é o baixo acoplamento entre as três camadas. Esta característica é fundamental para diminuir a quantidade de trabalho necessária caso seja exigida a troca do serviço de banco de dados. Este baixo acoplamento permite que se altere a interface do usuário sem grandes impactos no restante da aplicação (ZHANG; LUO, 2010).

## 5 SOLUÇÃO DESENVOLVIDA

A aplicação desenvolvida é composta por uma tela principal (Figura 5.1), onde é possível interagir com o mapa colaborativo e suas ocorrências. Ao clicarmos em uma ocorrência, é exibida a sua data (Figura 5.2). Foi criado um *overlay*<sup>1</sup> contendo o botão que inicia a captura de áudio.

Figura 5.1 – Tela inicial.

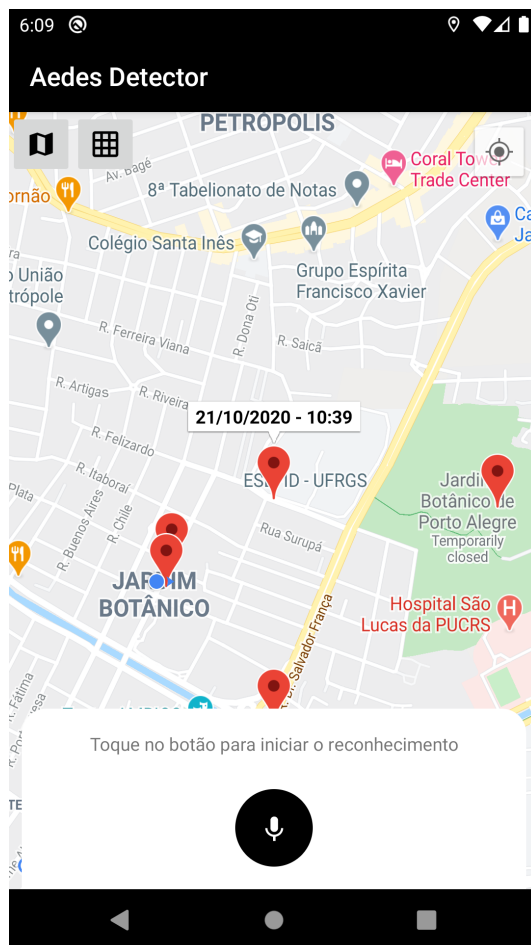


Fonte: do Autor.

Por questões de privacidade, ao inicializarmos o aplicativo pela primeira vez, é requisitado o consentimento do usuário para o uso de sua localização durante a execução da aplicação. Em caso negativo, a aplicação ainda pode ser utilizada; porém, não será possível informar automaticamente a localização de possíveis ocorrências de mosquito. Também por questões de privacidade, ao clicar-se no botão de gravação pela primeira

<sup>1</sup>Um *overlay* é um elemento gráfico que fornece informações e opções de ação na forma de uma camada adicional que se sobrepõe ao estado atual da interface.

Figura 5.2 – Detalhe da ocorrência no mapa.



Fonte: do Autor.

vez, é requisitado o consentimento do usuário para o uso do microfone. Se o usuário não fornecer a permissão de acesso, não é possível realizar a análise do áudio.

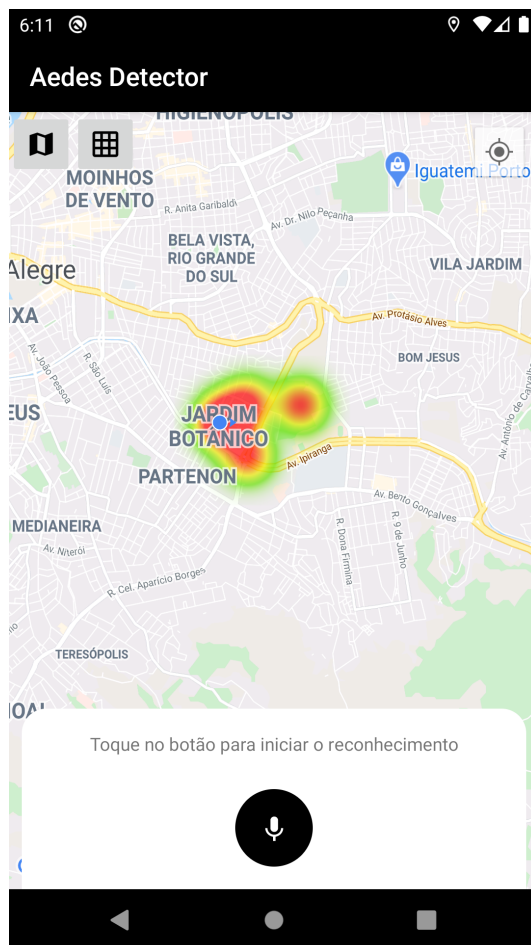
Há dois botões no canto superior esquerdo da interface principal. Com eles, é possível alternar-se o modo de exibição das ocorrências no mapa. Além da visualização independente das ocorrências, é possível exibir um mapa de calor (Figura 5.3), que facilita a observação de regiões onde o mosquito está mais presente.

Ao pressionar-se o botão em destaque, o *overlay* passa a ocupar a tela inteira enquanto realiza a análise do áudio capturado (Figura 5.4). Em caso positivo para *A. aegypti*, é exibido um alerta informando tal resultado e as opções de compartilhamento da ocorrência no mapa, conforme demonstrado na Figura 5.5 da página 39.

Se a aplicação não detectar o mosquito no tempo limite, ou se o usuário cancelar a ação apertando o botão novamente, é exibido um alerta informando a não-detecção (Figura 5.6). O tempo limite configurado é de 15 segundos, o que se mostrou suficiente



Figura 5.3 – Mapa de calor das ocorrências do mosquito.



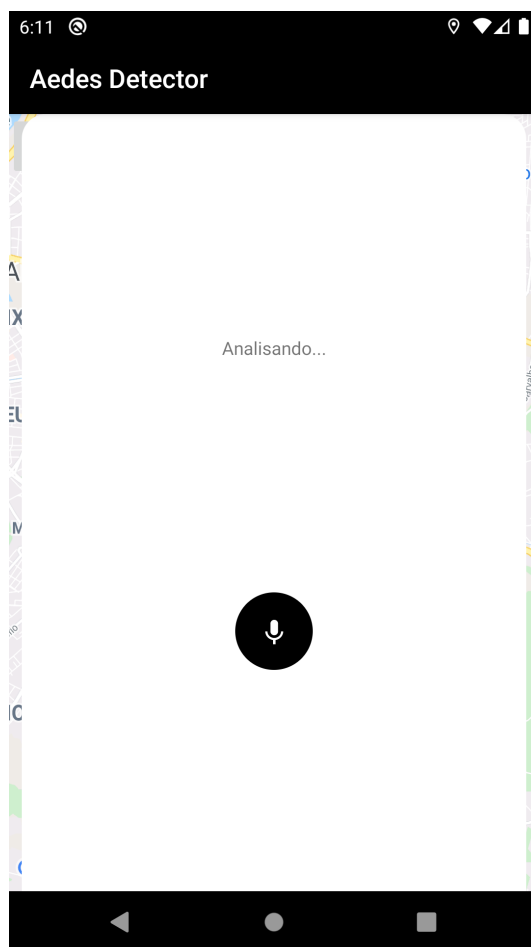
Fonte: do Autor.

para a detecção positiva durante a maioria dos testes, conforme será demonstrado no Capítulo 6. Todavia, este tempo de análise pode ser futuramente alterado no código-fonte sem maiores dificuldades.

Caso o usuário aceite compartilhar a sua ocorrência, o aplicativo exibe uma tela contendo o mapa da localização do usuário (Figura 5.7). Para não compartilhar a sua localização exata, é adicionado um pequeno erro à localização atual e exibido um raio de 20 metros, que será enviado para o servidor e compartilhado com os outros usuários. Se o usuário não tiver consentido com o compartilhamento de sua localização, é possível selecionar no mapa o ponto da sua ocorrência.

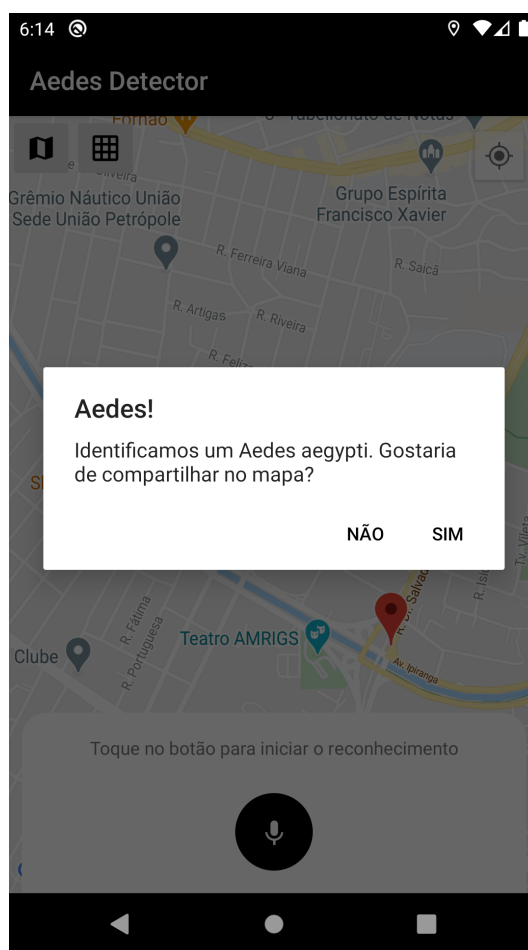
Além do modo de gravação e análise simultâneas, também foi desenvolvida uma funcionalidade que permite o carregamento de áudios gravados previamente. Originalmente desenvolvido para a validação do funcionamento da aplicação com os *datasets* originais, esta função encontra-se oculta na versão atual. Em versões iniciais do apli-

Figura 5.4 – Tela de análise de áudio.

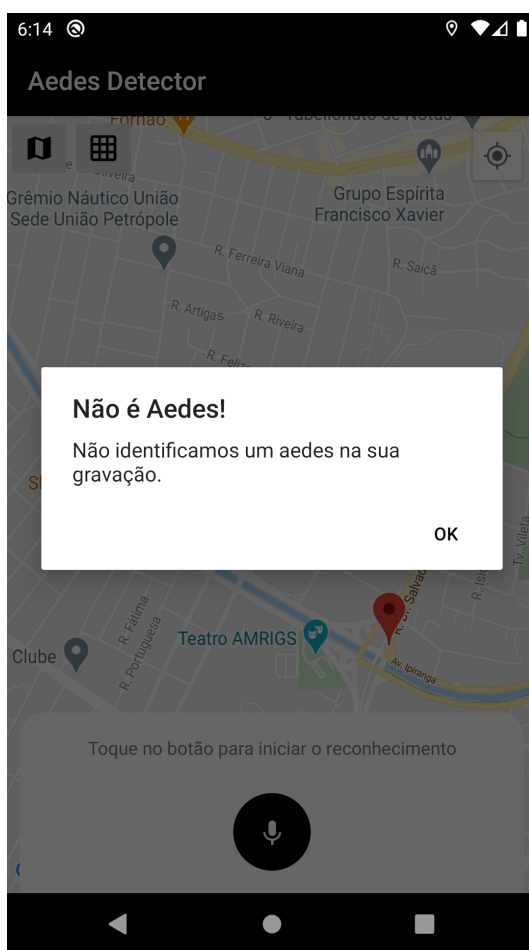


Fonte: do Autor.

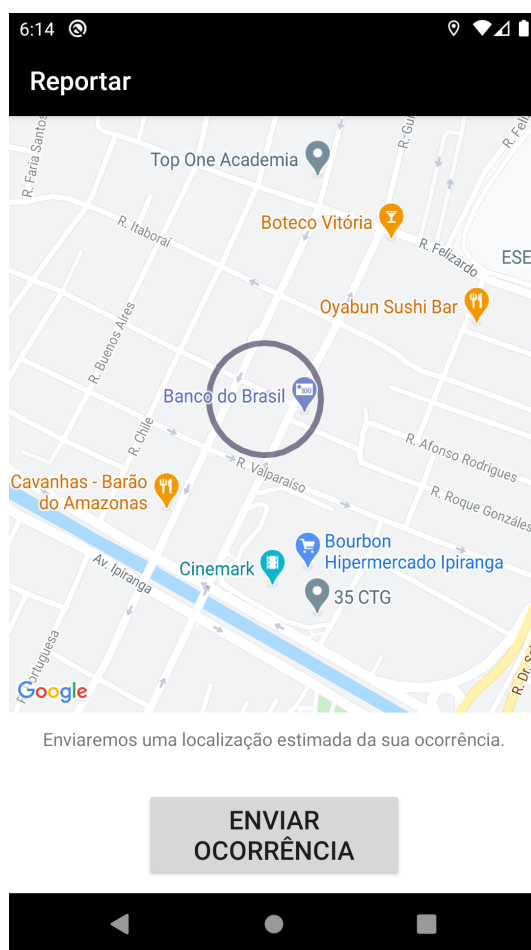
cativo, esta ferramenta foi mantida considerando-se situações em que o usuário receba o áudio do mosquito através de aplicativos de mensagem ou caso tenha utilizado outros meios de gravação que não o aplicativo em questão. Caso o aplicativo acuse a presença de *A. aegypti* através de um áudio carregado, o usuário tem a opção de selecionar a localização da ocorrência no mapa. Para evitar abusos de usuários maliciosos, estas ocorrências são marcadas com um identificador de origem do áudio (se foi gravado pelo aplicativo ou enviado pelo usuário).

Figura 5.5 – Alerta de detecção de *A. aegypti*.

Fonte: do Autor.

Figura 5.6 – Alerta de não detecção de *A. aegypti*.

Fonte: do Autor.

Figura 5.7 – Tela de relato da ocorrência de *A. aegypti*.

Fonte: do Autor.

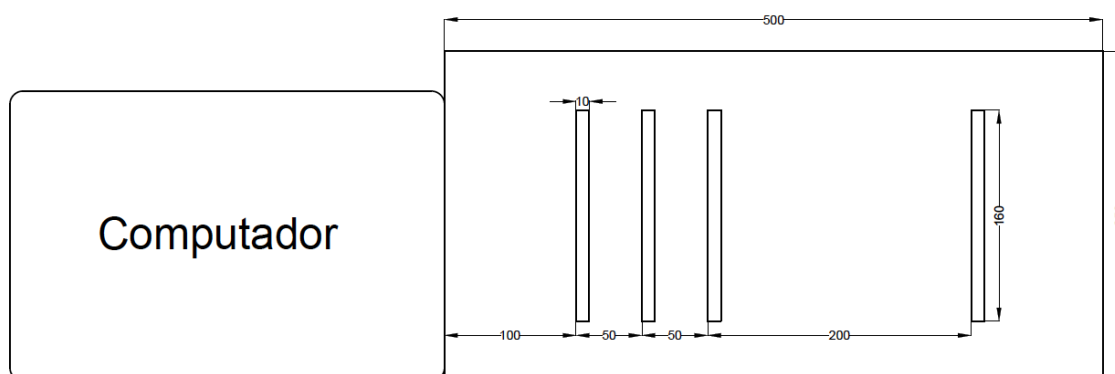
## 6 AVALIAÇÃO DO FUNCIONAMENTO

No intuito de avaliar o funcionamento do aplicativo e de compreender os efeitos de fatores ambientais sobre a detecção, uma série de testes foi realizada. Em uma primeira iteração sobre esse problema, utilizamos o *dataset* disponibilizado por Mukundarajan et al. (2017) e adaptado por Fernandes, Cordeiro and Recamonde-Mendoza (2020) para a avaliação da eficácia do aplicativo na monitoração de ruídos de mosquitos com som de fundo ambiente. Utilizou-se como fonte emissora de áudio um notebook Macbook Pro 13'. Como trabalhos futuros, pretende-se experimentar a aplicação desenvolvida com zumbidos reais de *A. aegypti* em um laboratório seguro e controlado, com todas as medidas de segurança e aprovações de comitê ético que um estudo dessa natureza requer.

O dispositivo no qual a aplicação foi instalada para os testes foi um Samsung Galaxy S10+ (2019). Embora este seja um dispositivo *high-end*, ou seja, a qualidade do *hardware* do aparelho seja superior à media, Mukundarajan et al. (2017) indicam que *smartphones* comuns, e até *feature phones*<sup>1</sup> podem realizar a captura do áudio do mosquito sem grandes perdas na sua qualidade. É importante salientar novamente que a captura do áudio do mosquito deve ser possível por aparelhos de baixo custo.

Para os testes de distância da fonte emissora, desenvolveu-se um suporte em poliestireno expandido com encaixes em diferentes distâncias para o aparelho móvel. O esquema do suporte desenvolvido encontra-se exposto na Figura 6.1. Quatro distâncias da fonte emissora são possíveis neste suporte: 10, 15, 20 e 40 cm.

Figura 6.1 – Esquema do suporte desenvolvido para testes. Todas as dimensões descritas estão em milímetros.



Fonte: Do Autor.

<sup>1</sup>Um *feature phone* é um aparelho celular simples, cujas funcionalidades se resumem ao envio de mensagens de texto e ligações.

## 6.1 Testes de detecção

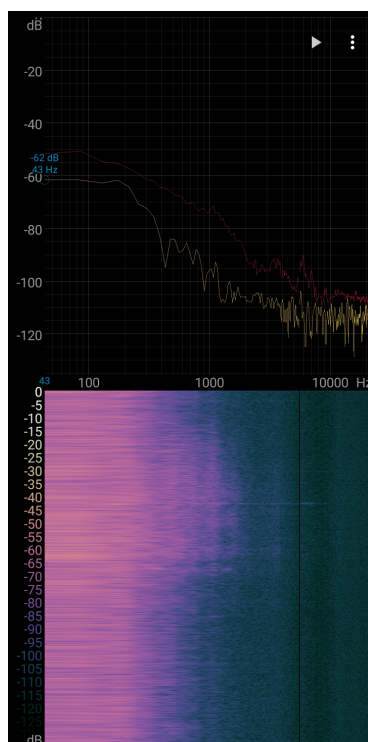
Para os testes dos efeitos ambientais, utilizou-se de um aparelho de ar-condicionado portátil Elgin MA-9000-2 a uma distância de 3 m. O suporte para testes foi posicionado na linha espacial entre o computador e o ar-condicionado. Três áudios padrão de *A. aegypti* do *dataset* disponibilizado por Mukundarajan et al. (2017) foram escolhidos para os testes devido à sua maior qualidade sonora. Estes áudios foram nomeados A1, A2 e A3. O efeito do som de uma televisão ligada também foi analisado. O equipamento foi posicionado em uma distância semelhante à dos testes anteriores. O vídeo utilizado na simulação foi a chamada do Jornal Nacional (Rede Globo) do dia 14/11/2020; o mesmo trecho do vídeo foi utilizado para todas as medições.

Os espectrogramas nas três condições de *background* foram medidos através do aplicativo Spectroid (disponível em versão grátis na loja Google Play). A Figura 6.2 mostra os resultados para o espectro de fundo na condição de silêncio máximo. Observa-se que, na maior parte do tempo, as componentes do som ambiente se situam em valores baixos de frequência (menores que 1 kHz). Também nesta faixa observam-se as maiores intensidades sonoras, denotadas por cores mais claras no espectrograma.

Os espectros sob influência de ar-condicionado e de som de televisão estão demonstrados na Figura 6.3. Quanto ao primeiro, observa-se comportamento semelhante ao do som ambiente, porém com maior distribuição em frequências mais altas. Quando à sua dispersão, pode-se dizer que o som produzido pelo ar condicionado é homogêneo, apesar de conter frequências mais altas. O espectro também demonstra ser temporalmente constante. O contrário pode ser dito do som da televisão. Seu espectro também contém frequências mais altas; porém, estas são mais dispersas com relação à dimensão temporal. Tal fenômeno está relacionado às vozes contidas na gravação, que apresentam aspecto de grande variação de entonação e volume.

Para fins de comparação, registrou-se a intensidade sonora nas quatro posições do suporte de testes enquanto um áudio teste (A1) foi tocado; os resultados estão demonstrados na Figura 6.4. Observa-se um pequeno crescimento inicial conforme o dispositivo móvel contendo o aplicativo foi afastado da fonte emissora. Uma queda acentuada na intensidade foi observada para a maior distância. Os valores medidos ficaram na faixa entre 43 e 47,5 dB. A intensidade nas diferentes posições tem um forte componente espacial, o

Figura 6.2 – Espectro sonoro do ambiente sem fontes de ruído.



Fonte: Do Autor.

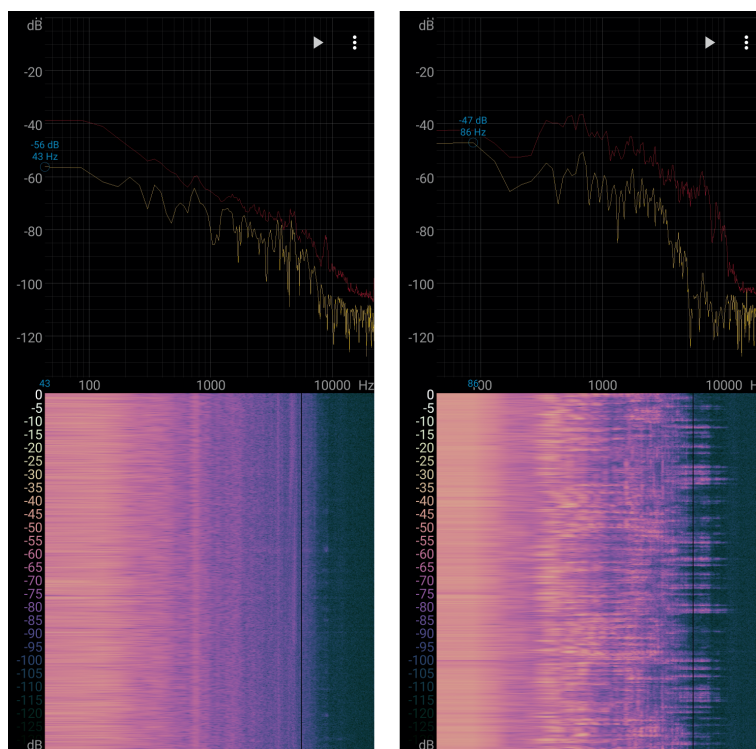
que dificilmente poderia ser controlado sem um ambiente específico para testes.

Repetindo-se a análise com o ar-condicionado ligado, podemos observar que a intensidade sonora aumenta aproximadamente 4 dB em todas as distâncias analisadas. Este fenômeno é claramente relacionado ao som produzido pelo compressor do equipamento. Testes similares foram feitos com uma televisão ligada. A intensidade sonora em todas as condições analisadas teve valores na faixa entre 53,3 e 54,1 dB. Uma menor dependência da posição foi observada nestas medições.

Procedeu-se ao teste de funcionamento do aplicativo. Os três áudios do mosquito foram analisados nas quatro distâncias estabelecidas, sem a influência de outras fontes sonoras. Todos os doze testes foram positivos para a "presença" do mosquito em questão. A Figura 6.5 exibe o tempo necessário para a identificação em função da distância para os três diferentes áudios. Pode-se inferir que o aplicativo funciona mais apropriadamente a até 20 cm de distância do som. Apesar de parecer uma distância curta, esta já é o dobro da normalmente relatada em literatura como distância máxima a gravação dos sons de mosquitos: Mukundarajan et al. (2017) indicam 10 cm como a maior separação possível entre mosquito e microfone para uma identificação satisfatória. Na maior distância analisada, o tempo de detecção médio aumenta significativamente em comparação com as outras três



Figura 6.3 – Espectro sonoro do ambiente com os ruídos do ar condicionado (à esquerda) e da televisão (à direita).



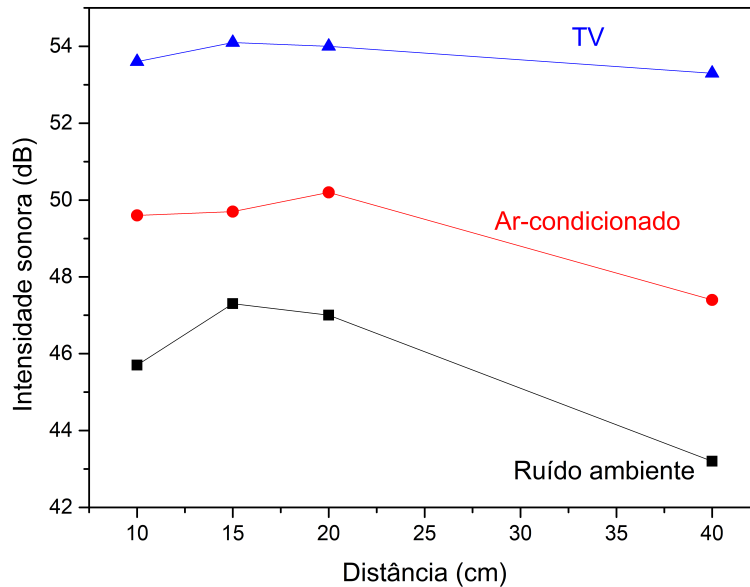
Fonte: Do Autor.

posições. O resultado positivo para o áudio A3 a 40 cm leva quase todo o tempo disponível no aplicativo. Entretanto, podemos observar que o tempo alocado para a identificação (15 s) foi suficiente para gerar resultados satisfatórios em todos os testes. O tempo médio de detecção foi de 6401 ms, com desvio padrão de 4726 ms, considerando-se uma amostra com 12 testes.

Quando os resultados foram repetidos com o ar-condicionado ligado, apenas 7 das instâncias de teste geraram identificação positiva: todas do áudio A1, duas do A2 e apenas uma das distâncias analisadas no áudio A3. Portanto, para compararmos o efeito do ar-condicionado sobre o tempo de detecção, analisou-se apenas o áudio A1, que forneceu número de dados relevante para uma análise significativa.

O tempo de detecção para este áudio em diferentes distâncias da fonte sob as duas condições de ruído de fundo estão dispostos na Figura 6.6. Surpreendentemente, observa-se que, nas duas menores distâncias, a presença de ruído de fundo do ar-condicionado levou a menores tempos para a detecção. Entretanto, em testes nas distâncias maiores, o aplicativo levou quase todo o tempo disponível para alcançar um resultado positivo, o

Figura 6.4 – Intensidade sonora média no ambiente de testes. As linhas são meramente auxílios visuais.

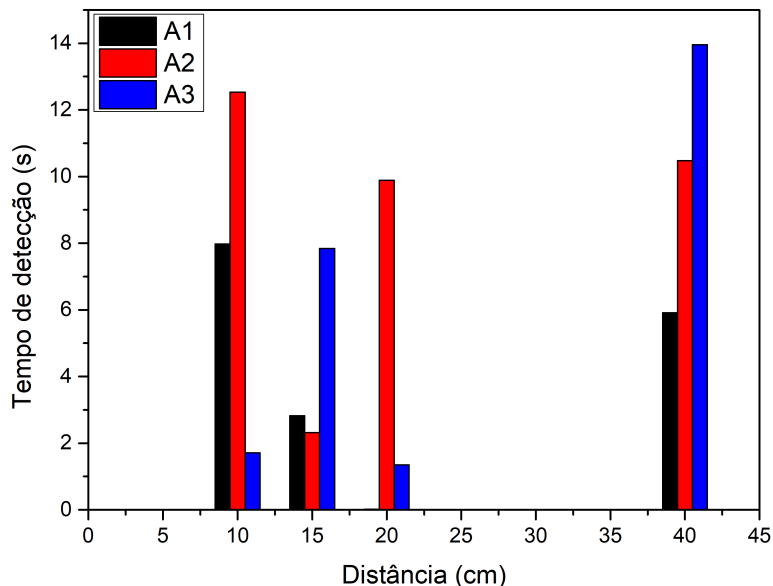


Fonte: do Autor.

que demonstra o efeito deletério do ruído de fundo sobre o pleno funcionamento da rede neural. Considerando-se a fraca performance observada para os áudios A2 e A3 sob as mesmas condições, pode-se supor que que os menores tempos observados para as distâncias de 10 e 15 cm foram flutuações estatísticas. Para os 7 testes positivos, o tempo médio de detecção foi de 8326 ms, com desvio padrão de 5782 ms, valores ligeiramente superiores aos obtidos na ausência de fontes sonoras externas. Em suma, conclui-se que o ruído constante do ar-condicionado tem um efeito levemente prejudicial sobre o desempenho da aplicação.

O mesmo não pode ser dito sobre o efeito de som de televisão. Nenhum dos doze testes simulando uma televisão ligada resultou em identificação do mosquito. Este resultado negativo pode ser relacionado ao espectro emitido pelo vídeo, como foi demonstrado na Figura 6.3. De maneira diferente ao que é emitido pelo ar condicionado, o som da televisão é inerentemente inconstante, o que pode levar à confusão da rede neural. Além disso, o espectro possui intensas componentes na região de emissão sonora do mosquito *A. aegypti*, entre 400 e 700 Hz. A sobreposição de sons nesta região também é nociva para um desempenho eficiente da rede neural utilizada. Mukundarajan et al. (2017) também

Figura 6.5 – Tempo de detecção positiva de *A. aegypti* em função da distância da fonte emissora de som.

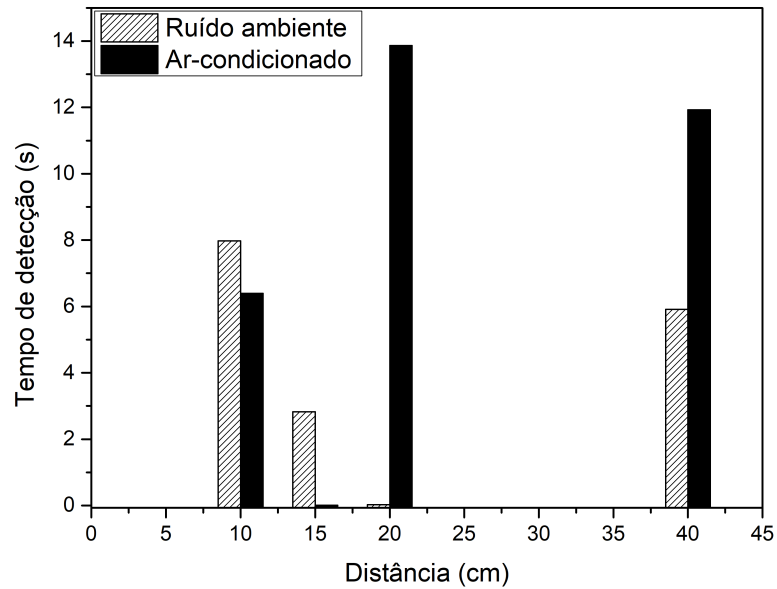


Fonte: do Autor.

indicam que a qualidade da detecção cai drasticamente em ambientes com intensidade sonora acima de 50 dB, como foi o caso destes experimentos.

Finalmente, também tentou-se gerar falsos positivos a partir dos sons de mosquitos de três outras espécies: *Culex quinquefasciatus*, *Anopheles sp.* e *Aedes albopictus*, escolhidas por estarem entre as mais prevalentes em território brasileiro. Os testes foram repetidos nas mesmas quatro distâncias anteriores e sem a influência de ruídos de fundo. Nenhum resultado positivo foi gerado na análise das duas primeiras espécies. Entretanto, uma das quatro análises do *A. albopictus* foi indicada como positiva para a presença de *A. aegypti*. O fato de estas duas espécies compartilharem posições no mesmo gênero (*Aedes*), bem como as frequências semelhantes emitidas por ambos mosquitos durante o voo – vide Figura 2.4 –, podem explicar este falso positivo. Em literatura, este problema da sobreposição entre frequências dos mosquitos vem sendo extensivamente estudado. Brogdon (1994) descreve o efeito da taxa de amostragem sobre a separação desta classe de sinais. O autor relata grande sobreposição com taxas de 1000 Hz, enquanto taxas acima de 5000 Hz resolvem consideravelmente este problema. No caso do nosso aplicativo, o microfone faz a gravação a uma taxa de 44000 Hz; entretanto, a taxa de amostragem da transformada de Fourier utilizada para a geração do espectrograma é de apenas 8000 Hz.

Figura 6.6 – Tempo de detecção positiva de *A. aegypti* em função da distância da posição da fonte emissora de som, com e sem ruído do ar condicionado.



Fonte: do Autor.

Esta frequência mais baixa poderia ter gerado dificuldade na separação dos sinais das diferentes espécies dentro do gênero *Aedes*.

## 7 CONSIDERAÇÕES FINAIS

Neste trabalho discutiu-se a pesquisa, o desenvolvimento e a avaliação de um aplicativo móvel capaz de identificar o mosquito *A. aegypti* através da captura do áudio do seu voo e a posterior análise em uma rede neural previamente treinada. Além disso, o aplicativo permitiu o compartilhamento das ocorrências positivas do mosquito em um mapa colaborativo que foi disponibilizado na própria aplicação. A aplicação foi desenvolvida para o sistema Android, e o Firebase Cloud Firestore foi utilizado como banco de dados remoto.

Foram armazenados metadados das ocorrências positivas de *A. aegypti*, como localização e horário da detecção. Estes dados alimentaram um mapa que pode ser utilizado para conscientizar o público em geral para as ocorrências de *A. aegypti* ao seu redor. Tais dados também poderão ser utilizados para treinar e melhorar o modelo da rede neural aplicado neste trabalho. Além disso, os dados disponibilizados ao público são de grande valia para os sistemas de saúde no aprimoramento de seus processos de mapeamento de surtos de doenças relacionadas.

Testes de funcionamento em diferentes condições ambientais foram realizados. O aplicativo mostra seu melhor desempenho em condições de baixo ruído. Neste caso, todos os testes realizados a uma distância de até 40 cm foram satisfatórios na identificação do mosquito *A. aegypti*. A influência sonora de um ar-condicionado mostrou-se deletéria para o pleno funcionamento do sistema. Apenas sete dos doze testes conseguiram detectar o mosquito com sucesso. Ruídos de televisão se mostraram particularmente prejudiciais; nenhum dos ensaios teve sucesso. Tal fenômeno pode ser atribuído ao volume do ruído de fundo, bem como ao caráter do espectrograma da televisão, com sua variabilidade temporal e dispersão de frequências.

Conforme demonstrado através dos testes, o aplicativo desenvolvido pôde ser utilizado com sucesso na identificação do mosquito em uma variedade de condições ambientais. O aplicativo poderá servir como base para a criação de uma ferramenta responsável por auxiliar os órgãos de saúde pública a otimizar o uso de recursos no combate ao mosquito *A. aegypti*, além de engajar a população a auxiliar nesta batalha.

### 7.1 Trabalhos futuros

Sugerem-se os seguintes pontos para análise em trabalhos futuros:

- Estudar a possibilidade de classificação de outros tipos de mosquitos transmissores de doenças.
- Aprimorar o modelo utilizado com metadados de localização e horário que contemplem os ciclos circadianos e a distribuição geoespacial de cada espécie.
- Aplicar algoritmos de filtragem de som, no intuito de minimizar influências de fontes sonoras externas.
- Exportar o algoritmo desenvolvido de análise de áudio para outros dispositivos e criar armadilhas portáteis (Arduino, Raspberry PI).
- Testar o funcionamento do aplicativo com amostras reais de mosquitos *A. aegypti*.

## REFERÊNCIAS

- ABADI, M. et al. TensorFlow: A system for large-scale machine learning. In: **Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016**. [S.l.: s.n.], 2016. ISBN 9781931971331.
- ALBAWI, S.; MOHAMMED, T. A.; AL-ZAWI, S. Understanding of a convolutional neural network. In: **2017 International Conference on Engineering and Technology (ICET)**. IEEE, 2017. p. 1–6. ISBN 978-1-5386-1949-0. Available from Internet: <<https://ieeexplore.ieee.org/document/8308186/>>.
- ALPAYDIN, E. **Introduction to Machine Learning**. 2nd. ed. [S.l.]: The MIT Press, 2010. ISBN 026201243X.
- ARTHUR, B. J. et al. Mosquito ( *Aedes aegypti* ) flight tones: Frequency, harmonicity, spherical spreading, and phase relationships . **The Journal of the Acoustical Society of America**, 2014. ISSN 0001-4966.
- BROGDON, W. G. Measurement of Flight Tone Differences Between Female *Aedes aegypti* and *A. albopictus* (Diptera: Culicidae). **Journal of Medical Entomology**, v. 31, n. 5, p. 700–703, sep 1994. ISSN 1938-2928. Available from Internet: <<https://academic.oup.com/jme/article-lookup/doi/10.1093/jmedent/31.5.700>>.
- CALLAHAM, J. **The history of Android: The evolution of the biggest mobile OS in the world**. 2020. Available from Internet: <<https://www.androidauthority.com/history-android-os-name-789433/>>.
- CATOR, L. J. et al. Behavioral Observations and Sound Recordings of Free-Flight Mating Swarms of *Ae. aegypti* (Diptera: Culicidae) in Thailand. **Journal of Medical Entomology**, v. 48, n. 4, p. 941–946, jul 2011. ISSN 00222585. Available from Internet: <<https://doi.org/10.1603/ME11019><https://academic.oup.com/jme/article-lookup/doi/10.1603/ME11019>>.
- CATTELL, R. Scalable SQL and NoSQL data stores. **SIGMOD Record**, 2010. ISSN 01635808.
- Centro Estadual de Vigilância em Saúde - RS. Informativo Epidemiológico de Arboviroses - Março de 2020 - Semana Epidemiológica 10 (01/03 a 07/03). 2020. Available from Internet: <<https://www.cevs.rs.gov.br/upload/arquivos/202003/12165347-informativo-epidemiologico-dengue-chik-zika-e-fa-se-10-2020.pdf>>.
- CHOLLET, F. **Deep Learning with Python**. [S.l.]: Manning, 2017. ISBN 9781617294433.
- COHN, M. **User Stories Applied: For Agile Software Development (Addison Wesley Signature Series)**. [S.l.: s.n.], 2004. ISBN 0321205685.
- DIELEMAN, S.; SCHRAUWEN, B. End-to-end learning for music audio. In: **2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. IEEE, 2014. p. 6964–6968. ISBN 978-1-4799-2893-4. ISSN 15206149. Available from Internet: <<http://ieeexplore.ieee.org/document/6854950/>>.

FERNANDES, M. S.; CORDEIRO, W.; RECAMONDE-MENDOZA, M. Detecting *Aedes Aegypti* Mosquitoes through Audio Classification with Convolutional Neural Networks. aug 2020. Available from Internet: <<https://arxiv.org/abs/2008.09024><https://arxiv.org/abs/2008.09024>>.

FØLSTAD, A.; KVALE, K. Customer journeys: a systematic literature review. **Journal of Service Theory and Practice**, v. 28, n. 2, p. 196–227, mar 2018. ISSN 2055-6225. Available from Internet: <<https://www.emerald.com/insight/content/doi/10.1108/JSTP-11-2014-0261/full/html>>.

GENOUD, A. P. et al. Optical remote sensing for monitoring flying mosquitoes, gender identification and discussion on species identification. **Applied Physics B: Lasers and Optics**, 2018. ISSN 09462171.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016.

Instituto Oswaldo Cruz. **O mosquito *Aedes aegypti* faz parte da história e vem se espalhando pelo mundo desde o período das colonizações**. 2020. Available from Internet: <<http://www.ioc.fiocruz.br/dengue/textos/longatraje.html>>.

KRASNER, G.; POPE, S. A Description of the Model-View-Controller User Interface Paradigm in the Smalltalk80 System. **Journal of Object-oriented Programming - JOOP**, v. 1, 1988.

Ministério da Saúde. **Dengue - Instruções para Pessoal de Combate ao Vetor - Manual de Normas Técnicas**. Brasília, 2001. Available from Internet: <[http://bvsmms.saude.gov.br/bvs/publicacoes/funasa/man{\\\_}dengue.>](http://bvsmms.saude.gov.br/bvs/publicacoes/funasa/man{\_}dengue.>)

Ministério da Saúde. **Boletim Epidemiológico 23 - Monitoramento dos casos de arboviroses urbanas transmitidas pelo *Aedes* (dengue, chikungunya e zika), Semanas Epidemiológicas 1 a 22, 2020**. Brasília, 2020. Available from Internet: <<https://www.conasems.org.br/wp-content/uploads/2020/06/Boletim-epidemiologico-SVS-23.pdf>>.

MITCHELL, T. M. **Machine Learning**. 1. ed. USA: McGraw-Hill, Inc., 1997. ISBN 0070428077.

MOORE, A. Artificial neural network trained to identify mosquitoes in flight. **Journal of Insect Behavior**, 1991. ISSN 08927553.

MUKUNDARAJAN, H. et al. Using mobile phones as acoustic sensors for high-throughput mosquito surveillance. **eLife**, 2017. ISSN 2050084X.

OLIVEIRA, C. S. de; da Costa Vasconcelos, P. F. Microcephaly and Zika virus. **Jornal de Pediatria**, v. 92, n. 2, p. 103–105, mar 2016. ISSN 00217557. Available from Internet: <<http://www.sciencedirect.com/science/article/pii/S0021755716000395><https://linkinghub.elsevier.com/retrieve/pii/S0021755716000395>>.

O'SHAUGHNESSY, D. Speech communications - human and machine, 2nd Edition. In: . [S.l.: s.n.], 2000.



PLATCHECK, E. R. METODOLOGIA DE ECODESIGN PARA O DESENVOLVIMENTO DE PRODUTOS SUSTENTÁVEIS. **Universidade Federal do Rio Grande do Sul Escola de Engenharia Mestrado**, 2003.

POTEL, M. MVP: Model-View-Presenter The Taligent Programming Model for C++ and Java. **Taligent Inc**, 1996. ISSN <null>.

RUEDA, L. M. Key to the Species of Adult Female Mosquitoes of Texas. In: DEBBOUN, M.; NAVA, M. R.; Rueda Communities, and Public Health in Texas, L. M. B. T. M. (Ed.). **Mosquitoes, Communities, and Public Health in Texas**. Elsevier, 2020. p. 177–192. ISBN 978-0-12-814545-6. Available from Internet: <<http://www.sciencedirect.com/science/article/pii/B9780128145456000043><https://linkinghub.elsevier.com/retrieve/pii/B9780128145456000043>>.

STAUNTON, K. M. et al. A Novel Methodology For Recording Wing Beat Frequencies of Untethered Male and Female *Aedes aegypti*. **Journal of the American Mosquito Control Association**, v. 35, n. 3, p. 169–177, sep 2019. ISSN 8756-971X. Available from Internet: <<https://doi.org/10.2987/18-6799>.1<https://meridian.allenpress.com/jamca/article/35/3/169/438794/A-Novel-Methodology-For-Recording-Wing-Beat>>.

STEVENS, S. S.; VOLKMANN, J. The Relation of Pitch to Frequency: A Revised Scale. **The American Journal of Psychology**, University of Illinois Press, v. 53, n. 3, p. 329, jul 1940. ISSN 00029556. Available from Internet: <<http://www.jstor.org/stable/1417526><https://www.jstor.org/stable/1417526?origin=crossref>>.

ZHANG, Y.; LUO, Y. An architecture and implement model for model-view-presenter pattern. In: **Proceedings - 2010 3rd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2010**. [S.l.: s.n.], 2010. ISBN 9781424455386.