



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ
CAMPUS CORRENTE
TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

JÚLIA GABRIELA MEDRADO FERREIRA

APLICAÇÃO MOBILE PARA CLASSIFICAÇÃO DA VARÍOLA DOS MACACOS

CORRENTE

2022

JÚLIA GABRIELA MEDRADO FERREIRA

APLICAÇÃO MOBILE PARA CLASSIFICAÇÃO DA VARÍOLA DOS MACACOS

Trabalho de Conclusão de Curso apresentado como exigência parcial para obtenção do diploma do Curso de Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Corrente.

Orientador: Prof^a. Me. Felipe Gonçalves dos Santos.

CORRENTE

2022

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD

F383a Ferreira, Júlia Gabriela Medrado
Aplicação mobile para classificação da varíola dos Macacos / Júlia
Gabriela Medrado Ferreira. - 2023.
25 p.: il. color.

Trabalho de Conclusão de Curso (Análise e Desenvolvimento de
Sistemas) - Instituto Federal de Educação, Ciência e Tecnologia do
Piauí, Campus Corrente, 2023.

Orientador : Prof Me. Felipe Gonçalves dos Santos.

1. Análise e Desenvolvimento de Sistemas - estudo e ensino. 2.
Aprendizado de máquina. 3. Varíola dos macacos. 4. Flutter. I. Título.

CDD - não encontrada

Elaborado por Tefischer Huanderson Soares e Sousa CRB 3/1251

LISTA DE ILUSTRAÇÕES

Figura 1	- Validação de imagens	12
Figura 2	- Print telas do app	12
Figura 3	- Processo de Treinamento e Implementação do modelo ao aplicativo	13
Figura 4	- Arquitetura e Tecnologias utilizadas no projeto	14
Figura 5	- Exemplos de conjuntos de imagens do dataset	15
Figura 6	- Treinamento do modelo	15
Figura 7	- Dependências do aplicativo	16
Figura 8	- Fluxo do aplicativo	17
Figura 9	- Telas iniciais	17
Figura 10	- Telas de classificação	18
Figura 11	- Telas de classificação 2	19
Figura 12	- Acurácia por classe	19
Figura 13	- Matriz de confusão	20
Figura 14	- Perda por época	20

LISTA DE SIGLAS

API Application Programming Interface

GTM Google Teachable Machine

IA Inteligência Artificial

MS Ministério da Saúde

OMS Organização Mundial da Saúde

PIC Personal Image Classifier

SVG Scalable Vector Graphics

SUMÁRIO

1 INTRODUÇÃO	8
2 REFERENCIAL TEÓRICO	9
2.1 VARÍOLA DOS MACACOS	9
2.2 APRENDIZADO DE MÁQUINA	9
2.3 TENSORFLOW	10
2.4 DESENVOLVIMENTO DE APLICATIVOS MÓVEIS	10
2.4.1 APLICAÇÕES HÍBRIDAS X NATIVAS	11
2.5 DART	11
2.6 TRABALHOS RELACIONADOS	11
3 METODOLOGIA	13
3.1 DESENVOLVIMENTO	13
3.2 FERRAMENTAS E TECNOLOGIAS	14
3.2.1 TENSORFLOW LITE E GOOGLE TEACHABLE MACHINE	14
3.2.2 DATASET	14
3.2.3 MODELO	15
3.2.4 FLUTTER	16
4 RESULTADOS E DISCUSSÕES	16
5 CONSIDERAÇÕES FINAIS	21
REFERÊNCIAS	22
GLOSSÁRIO	25

APLICAÇÃO MOBILE PARA CLASSIFICAÇÃO DA VARÍOLA DOS MACACOS

Júlia Gabriela Medrado Ferreira¹
Felipe Gonçalves dos Santos²

RESUMO

Enquanto o mundo ainda está se recuperando da pandemia do Coronavírus, surge um novo surto causado pelo vírus da varíola dos macacos, já relatado em 110 países. Os atributos clínicos da varíola dos macacos muitas vezes se assemelham às de outros tipos de varíola, a detecção da varíola do macaco pode facilitar o tratamento e a contenção da disseminação comunitária. O aprendizado de máquina tem mostrado grande potencial em imagens médicas e aplicações de detecção de doenças. Com a crescente popularidade das aplicações móveis e com os smartphones cada vez mais acessíveis para as pessoas, os aplicativos surgem para solucionar diversos tipos de problemas, inclusive na proposição de alternativas para questões da saúde. Com isso, o presente trabalho teve por objetivo desenvolver uma aplicação multiplataforma destinada a classificação da varíola dos macacos, tendo como base imagens de doenças que apresentam lesões de pele e erupções cutâneas causadas por vírus. Para o desenvolvimento do aplicativo foi utilizado o framework Flutter. Os resultados mostram que o aplicativo desenvolvido atendeu satisfatoriamente o objetivo proposto, apontando a aplicabilidade do sistema de classificação de imagens utilizando o TensorFlow Lite em desenvolvimento mobile.

Palavras-chaves: varíola dos macacos, aprendizado de máquina, tensorflow, flutter, dart.

ABSTRACT

While the world is still recovering from the Coronavirus pandemic, there is a new outbreak caused by the monkeypox virus, already reported in 110 countries. The clinical attributes of monkeypox often resemble those of other types of smallpox, detection of monkeypox can facilitate treatment and containment of community spread. Machine learning has shown great potential in medical imaging and disease detection applications. With the growing popularity of mobile applications and with smartphones becoming more accessible to people, applications emerge to solve different types of problems, including proposing alternatives to health issues. Therefore, the present work aimed to develop a multiplatform application for the classification of monkeypox, based on images of diseases that present skin lesions and rashes caused by viruses. For the development of the application, the Flutter framework was used. The results show that the developed application satisfactorily met the proposed objective, pointing out the applicability of the image classification system using TensorFlow Lite in mobile development.

Keywords: monkeypox, machine learning, tensorflow, flutter, dart.

¹ Graduando Análise e Desenvolvimento de Sistemas. Instituto Federal do Piauí. julia.jgmf@gmail.com.

² Professor Orientador. Instituto Federal do Piauí. felipe.santos@ifpi.edu.br.

1 INTRODUÇÃO

Enquanto o mundo ainda está se recuperando da pandemia devido ao Coronavírus, surgem relatos de um novo surto causado pelo vírus da varíola dos macacos como mais um vírus global. O vírus vem se espalhando pelo mundo em um ritmo acelerado. De acordo com a OMS, até o momento, o surto de varíola dos macacos foi relatado em 110 países, com 81.766 casos confirmados por laboratório e 1.507 casos prováveis, incluindo 60 mortes (OMS, 2022). No dia 23 de julho, Tedros Adhanom Ghebreyesus, o diretor-geral da Organização Mundial da Saúde, declarou que o atual surto de monkeypox (varíola dos macacos) constitui uma Emergência de Saúde Pública de Importância Internacional (OPAS, 2022). Com os dados obtidos pela OMS na data do presente trabalho os 5 países mais afetados globalmente são: Estados Unidos da América (n = 29.169), Brasil (n = 10.111), Espanha (n = 7.408), França (n = 4.110) e Colômbia (n = 3.852) (OMS, 2022).

No monitoramento do cenário epidemiológico nacional, foram registradas 10.111 casos confirmados e 3.917 casos suspeitos para monkeypox até 5 de dezembro de 2022. Ao analisar a distribuição espacial dos casos confirmados e prováveis por região de residência, observam-se maiores concentrações nas regiões Sudeste com 5.920 casos, representando 62,7% e o Centro-Oeste com 1.143 e 12,1% (MS, 2022). De acordo com a Organização Pan-Americana da Saúde, o Brasil registrou 14 mortes por varíola dos macacos até 5º de dezembro de 2022 (OPAS, 2022).

Nesse contexto, o desenvolvimento de sistemas computacionais, em termos de capacidade de processamento e armazenamento, permite que a IA seja amplamente aplicável em diversos campos. Podendo ser aplicada nesses ambientes o uso de softwares de classificação para ajudar a detectar e identificar doenças por meio de imagens. O aprendizado de máquina é um campo crescente da inteligência artificial que tem mostrado grande potencial em uma variedade de configurações, desde assistência à decisão e aplicações industriais até imagens médicas e aplicações de detecção de doenças.

Na contemporaneidade atual, a tecnologia avança rápido e continuamente, modificando os hábitos e formas do indivíduo atuar em sociedade, integrando-se cada vez mais no cotidiano das pessoas. Segundo Carvalho (2021, p.5), “A tecnologia da informação e suas subáreas estão transformando a realidade de muitos setores da sociedade com as incontáveis possibilidades de aplicações que conseguem auxiliar as pessoas durante o cotidiano”.

Em consequência disso, as aplicações móveis vêm ganhando popularidade e com os smartphones cada vez mais acessíveis para as pessoas, naturalmente surge uma grande demanda por desenvolvimento de aplicativos para solucionar diversos tipos de problemas. De acordo com o site de estatísticas STATISTA, o número de smartphones no mundo hoje já ultrapassou 6 bilhões, e espera-se que cresça em centenas de milhões nos próximos anos. China, Índia e Estados Unidos são os países com maior número de usuários de smartphones. Em 2021, 75,6% da população brasileira usava smartphone.

Este projeto partiu da necessidade de assumir uma postura responsável diante dos processos de mudança do cenário social. Até o momento da escrita deste artigo, não foi identificado nenhum aplicativo móvel que execute o processamento de imagens para a classificação da varíola dos macacos. Devido a falta de aplicativos para tal funcionalidade, chegou-se a uma conclusão significativa: há a necessidade de desenvolver um aplicativo para preencher essa lacuna. Para preenchê-lo, este trabalho propõe o desenvolvimento de uma aplicação classificadora utilizando flutter. Com isso, o presente trabalho teve por objetivo desenvolver uma aplicação multiplataforma destinada a classificação da varíola dos macacos, tendo como base imagens de doenças que apresentam lesões de pele e erupções cutâneas causadas por vírus.

2 REFERENCIAL TEÓRICO

Nesta seção, será discutido o contexto utilizado para desenvolver e executar o projeto proposto, conceitos básicos serão apresentados para melhor compreensão da aplicação proposta.

2.1 VARÍOLA DOS MACACOS

A varíola dos macacos é transmitida pelo vírus monkeypox, o nome se originou a partir da descoberta inicial do vírus em macacos em um laboratório dinamarquês em 1958. É considerada uma zoonose viral, isto é, é transmitido aos seres humanos a partir de animais. O primeiro caso humano foi identificado em 1970 na República Democrática do Congo. O vírus é pertencente ao gênero orthopoxvirus da família Poxviridae (BUTANTAN, 2022).

A família Poxviridae consiste em duas subfamílias, a subfamília Chordopoxvirinae e a subfamília Entomopoxvirinae. A Chordopoxvirinae contém 18 gêneros, incluindo o gênero Orthopoxvirus, este inclui os vírus que causam varíola humana, varíola bovina, varíola dos macacos, varíola equina e outros. A subfamília Entomopoxvirinae inclui o vírus da varíola encontrado em invertebrados (KREUTZ; REZENDE; MATÉ, 2022).

Os atributos clínicos da varíola dos macacos, lesões cutâneas e erupções cutâneas, muitas vezes se assemelham às de outros tipos de varíola, como catapora e varíola bovina (ISLAM et al., 2022). Enquanto a varíola foi erradicada em 1980, a varíola dos macacos continua a ocorrer em países da África Central e Ocidental.

A varíola dos macacos apresenta sintomas como febre, uma extensa erupção característica é geralmente gânglios linfáticos inchados. É possível ser transmitida pelo contato com fluidos corporais, lesões na pele ou em superfícies mucosas internas, como boca ou garganta, gotículas respiratórias e objetos contaminados. É importante distinguir a varíola dos macacos de outras doenças, como catapora, sarampo, infecções bacterianas da pele, sarna, sífilis e alergias associadas a medicamentos (OMS, 2022).

2.2 APRENDIZADO DE MÁQUINA

O aprendizado de máquina é a ciência e a arte da programação de computadores capaz de aprender com dados. É apto a compreender problemas complexos e grandes quantidades de dados. O aprendizado de máquina é ideal para problemas em que as soluções existentes exigem configuração manual ou uma grande lista de regras. Para problemas complexos para os quais as abordagens tradicionais não têm solução adequada. E em ambientes flutuantes onde há necessidade de se adaptar a novos dados (GÉRON, 2019).

De acordo com Géron (2019), os sistemas de aprendizado de máquina podem ser classificados com base na quantidade e o tipo de supervisão recebida durante o treinamento. Possuem quatro tipos principais de aprendizado: supervisionado, não supervisionado, semi supervisionado e por reforço.

No aprendizado supervisionado, os dados de treinamento são fornecidos ao algoritmo e incluem rótulos. Uma tarefa característica do aprendizado supervisionado é a classificação. Outra tarefa característica é a de regressão, fazendo a previsão a partir de um conjunto de características. Alguns dos algoritmos mais importantes de aprendizado supervisionado são o k-Nearest Neighbours, Regressão Linear, Regressão Logística, Máquinas de Vetores de Suporte, Árvores de Decisão e Florestas Aleatórias, Redes Neurais. Já no aprendizado não supervisionado, o sistema tenta aprender sem orientação, os dados fornecidos não são rotulados. Alguns dos algoritmos mais importantes do aprendizado não supervisionado são o Clustering, Visualização e redução da dimensionalidade e o Aprendizado da regra da associação (GÉRON, 2019).

O aprendizado semi-supervisionado lida com uma grande quantidade de dados não rotulados e um pouco de dados rotulados. A maior parte dos algoritmos de aprendizado semi-supervisionado é de combinações de algoritmos supervisionados e não supervisionados. Já o aprendizado por reforço é diferente, o sistema de aprendizado aprende por si só qual é a melhor estratégia através da observação do ambiente, ele receberá recompensas troca ou penalidades em troca (GÉRON, 2019).

2.3 TENSORFLOW

TensorFlow é uma plataforma de código aberto, desenvolvida pelo Google Brain Team com o propósito de conduzir aprendizado de máquina e pesquisa de redes neurais profundas. Por trás desses dois nomes está um poderoso conjunto de algoritmos que compartilham um único desafio. Trata-se de permitir que os computadores reconheçam automaticamente padrões complexos e aprendam a tomar as decisões mais inteligentes (ZACCONE, 2016).

O TensorFlow foi projetado com ponto central em portabilidade, permitindo cálculos gráficos a serem executados em uma ampla variedade de ambientes e plataformas de hardware. Outro aspecto importante do TensorFlow é sua flexibilidade, isso o torna relativamente fácil para pesquisadores e cientistas de dados expressarem seus modelos. Os principais facilitadores são as bibliotecas de abstração de nível (HOPE; RESHEFF; LIEDER, 2017).

Uma tarefa fundamental em visão computacional é a classificação de imagens, construir algoritmos e sistemas que recebem imagens como entrada e retornam um conjunto de categorias que melhor as descrevem (HOPE; RESHEFF; LIEDER, 2017). O aprendizado profundo visa aproximar o aprendizado de máquina dos objetivos da inteligência artificial. Isso significa que os algoritmos de aprendizado profundo estão tentando se comportar como o cérebro humano. Para realizar pesquisas nessas áreas, a equipe do Google desenvolveu o TensorFlow (ZACCONE, 2016).

2.4 DESENVOLVIMENTO DE APLICATIVOS MÓVEIS

Segundo El-Kassas et al. (2017), o desenvolvimento de aplicativos móveis tornou-se uma plataforma importante para produtos e serviços de software, diferenciando-se de outros tipos de desenvolvimento de software por possuir particularidades e restrições. Envolvendo uma vasta gama de aplicativos: comunicação, entretenimento, negócios, mídia e entre outros.

O desenvolvimento de software nesse ambiente precisa levar em consideração modelagem, implantação e necessidades de marketing, e também exige lidar com um relacionamento complexo entre as partes interessadas.

2.4.1 Aplicações híbridas x Nativas

Aplicativos nativos são desenvolvidos especificamente para uma plataforma, usando ferramentas de desenvolvimento e linguagens de programação suportadas, para poder executar e explorar todas as funções que tal plataforma possui. Posto isto, esses aplicativos são criados especificamente para um sistema operacional (iOS ou Android), e uma vez que o código é desenvolvido para uma plataforma, ele não pode ser reutilizado em outra. Podemos entender qualquer programa customizado para uma plataforma como um aplicativo nativo, de forma a trabalhar em harmonia com o dispositivo e todas as suas características, capaz de executar e explorar todas as funções da plataforma a que se destina.

De acordo com, Montan e Santos (2018), as aplicações híbridas possuem o código compartilhado entre as plataformas, sendo desenvolvido apenas uma vez. Dispõe de três abordagens de desenvolvimento: Tempo de

Execução, Tradutor de Código e Web-para-Nativo. Um aplicativo híbrido é uma ponte entre as web apps e um aplicativo nativo, parte nativo e parte da Web. Os aplicativos híbridos são projetados para funcionar em qualquer tipo de dispositivo e em diferentes plataformas usando o mesmo código-fonte.

2.5 DART

Dart é uma linguagem otimizada para o cliente para o desenvolvimento de aplicativos rápidos em qualquer plataforma. Seu objetivo é oferecer a linguagem de programação mais produtiva para desenvolvimento multiplataforma, combinada com uma plataforma de execução flexível para frameworks de aplicativos. O Dart foi projetado para um envelope técnico particularmente adequado ao desenvolvimento do cliente, priorizando tanto o desenvolvimento quanto experiências de produção de alta qualidade em uma ampla variedade de destinos de compilação (web, dispositivos móveis e desktop).

Dart também forma a base do Flutter, fornecendo a linguagem e os tempos de execução que impulsionam os aplicativos Flutter, também oferece suporte a muitas tarefas principais do desenvolvedor, como formatação, análise e teste de código. A linguagem Dart usa a verificação de tipo estático para garantir que o valor de uma variável sempre corresponda ao tipo estático da variável. O sistema de tipagem Dart é flexível, permitindo o uso de um tipo dinâmico combinado com verificações em tempo de execução.

2.6 TRABALHOS RELACIONADOS

Kreutz, Rezende e Maté (2022) fazem uma revisão acerca do vírus MonkeyPox. Foram notados em estudos filogenéticos com o Monkeypox isolado diferenças genéticas entre da África Ocidental e da Bacia do Congo que poderiam se refletir na patogenicidade do Monkeypox. A clado da África Ocidental apresenta menor patogenicidade e da Bacia do Congo maior patogenicidade. Aponta que as características epidemiológicas do surto global diferem das observadas no continente Africano, indicando a possibilidade de alterações na biologia do vírus associadas ao comportamento humano. Ressalta a comprovação da transmissão do vírus monkeypox de humanos infectados para cães, restringindo o contato de pessoas infectadas com seus pets.

Beltran, Felix e Thielo (2017) fazem a classificação de quatro espécies de plantas medicinais, sendo elas: Aloe Vera, Matricaria chamomilla, Peumus boldus e Achyrocline satureioides. No treinamento foi utilizado o script mobilenet e TensorFlow. Para a validação, as imagens foram carregadas através de linhas de comandos e através da câmera em um app Android, a classificação apontou a planta correta em 99% das ocorrências.

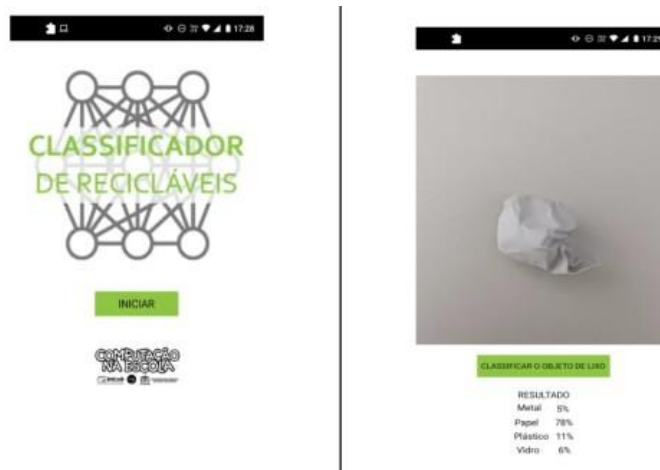
Figura 1 – Validação de imagens



Fonte: Beltran, Felix e Thielo, 2017.

No estudo realizado por Oliveira et al. (2022) foi desenvolvido a extensão Teachable Machine Image Classifier para classificação de imagens utilizando modelos de aprendizado de máquina exportado em aplicativos desenvolvidos na plataforma App Inventor. Durante o desenvolvimento da extensão foi utilizado como base o código da extensão PIC no App Inventor.

Figura 2 – Print telas do app



Fonte: Oliveira et al, 2022.

O artigo intitulado: “Prisma: Um classificador de imagens para aprendizado de idiomas” propõe criar uma solução que complemente o aprendizado de linguagens usando a classificação de imagens. Para a implementação dessa solução foi utilizado o TensorFlow Lite para fazer a classificação das imagens com React Native. O modelo pré-treinado utilizado foi o MobileNet que é ideal para dispositivos mobile.

Coutinho et al. (2019) faz uma revisão bibliográfica acerca de como ocorre o processo de desenvolvimento de projetos dos aplicativos para dispositivos móveis. É destacado que o desenvolvimento de aplicativos para dispositivos móveis possuem desafios e complicações, como lidar com recursos físicos limitados. Notou-se que a maioria dos aplicativos visam atender a grande massa de usuários, recomendando o desenvolvimento de aplicativo multiplataforma para abranger as principais plataformas do mercado. A análise leva ao entendimento que o sistema Android representa a plataforma mais utilizada no desenvolvimento para dispositivos móveis, deixando o sistema IOS em segundo lugar.

Machiaveli, Marciano e Dos Santos demonstraram um aplicativo móvel que usa inteligência artificial para classificar serpentes peçonhentas, auxiliando a identificar corretamente as espécies em caso de acidente e evitar o manuseio incorreto desses animais. O modelo de rede neural convolucional implementado no aplicativo foi desenvolvido usando técnicas de transfer learning e treinado com um banco de imagens de serpentes peçonhentas e não peçonhentas. Desenvolvido no framework Flutter, o app utiliza a câmera do aparelho para capturar imagens e realizar a classificação correta da espécie, fornecendo informações relevantes sobre a mesma. O aplicativo apresentou uma taxa de acerto de 88,41% no teste com o aplicativo mobile e 90,22% no nível de acurácia

3 METODOLOGIA

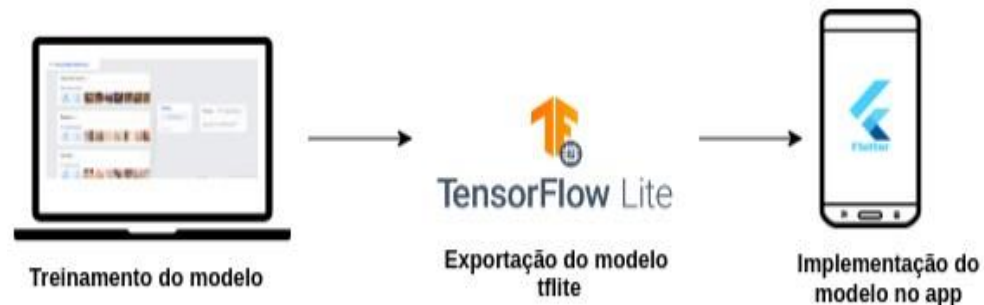
Em termos de objetivos, este estudo foi classificado como exploratório, pois proporciona um estudo para familiarizar-se com o tema, com vistas a torná-lo mais explícito. Sendo bastante flexível, para que possam ser considerados os mais diversos aspectos relacionados aos fatos estudados. (GIL, 2002). Quanto à natureza, trata-se de um estudo aplicado, de produção tecnológica, pois trata-se de uma aplicação para gerar conhecimento para aplicações práticas voltadas à resolução de problemas específicos.

Dada a abordagem qualitativa deste estudo, as técnicas utilizadas para coleta de dados foram por meio de leitura das publicações e análise do conteúdo dos artigos, observação de vídeos selecionados, e análise dos dados obtidos no desenvolvimento.

3.1 DESENVOLVIMENTO

Para a realização deste estudo foi utilizado as tecnologias de desenvolvimento como TensorFlow e Teachable Machine para treinar um modelo (Figura 3). Para a implementação, foi escolhida a linguagem de programação Dart e o framework Flutter (Figura 3). Na última etapa, foi utilizado o plugin do flutter tflite para acessar a API do Tensorflow Lite. Desta forma, o modelo irá prever a possibilidade de cada uma das quatro classes com base em qual probabilidade é maior.

Figura 3 – Processo de Treinamento e Implementação do modelo ao aplicativo.

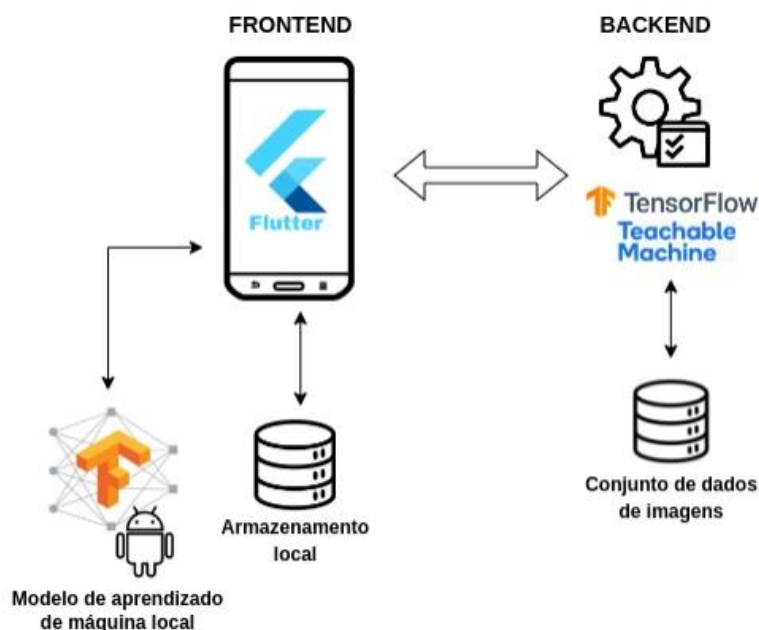


Fonte: Autoria própria.

3.2 FERRAMENTAS E TECNOLOGIAS

A Figura 4 mostra as tecnologias contidas no desenvolvimento. Sendo dividido em frontend, responsável pela experiência do usuário, e o backend para fazer a integração das informações.

Figura 4 – Arquitetura e Tecnologias utilizadas no projeto.



Fonte: Autoria própria.

3.2.1 TENSORFLOW LITE E GOOGLE TEACHABLE MACHINE

O TensorFlow Lite fornece um conjunto de ferramentas que permite o aprendizado de máquina no dispositivo, ajudando os desenvolvedores a executarem modelos em dispositivos móveis, incorporados e de IoT (TENSORFLOW, 2022).

GTM é uma ferramenta capaz de segmentar objetos, reconhecer padrões através de imagens, utilizando o TensorFlow para fazer treinamentos de reconhecimentos. Suporta o upload de conjuntos de dados, treinamento de modelos e avaliação das classes. Possibilita a implantação de modelos treinados exportados como TensorFlow.tflite, formato aplicado no desenvolvimento deste trabalho.

3.2.2 DATASET

Os experimentos conduzidos neste trabalho foram baseados em conjuntos de dados disponíveis no Kaggle³. O dataset montado consiste em 455 imagens de casos de varíola dos macacos, 107 imagens de Catapora, 91 imagens de Sarampo e 326 imagens de casos normais, totalizando 979 imagens. Consistindo em quatro classes: “Varíola dos macacos”, “Catapora”, “Sarampo” e “Normal” (Figura 5).

³ <https://www.kaggle.com/datasets/jyotidabas/monkeypox-dataset>
<https://www.kaggle.com/datasets/dipuiucse/monkeypoxskinimagedataset?resource=download>

Figura 5 – Exemplos de conjuntos de imagens do dataset. Varíola dos Macacos(A), Catapora(B), Sarampo(C) e Normal(D).

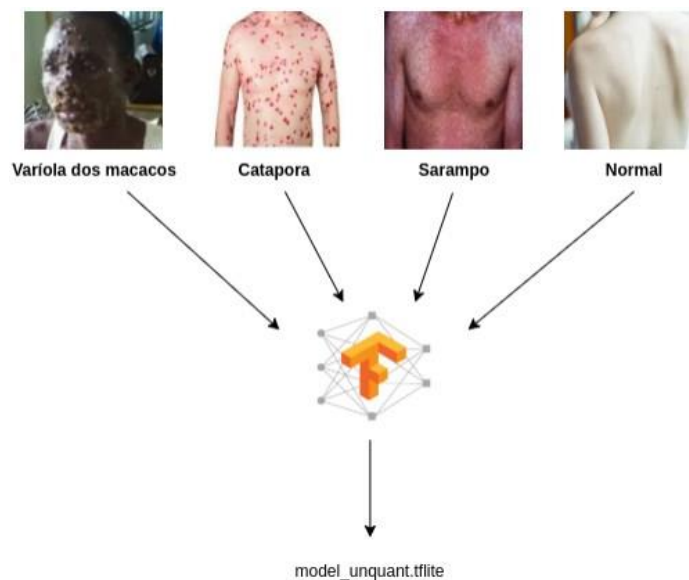


Fonte: Autoria própria.

3.2.3 MODELO

O treinamento foi definido para 100 épocas de aprendizagem (vezes cada amostra do conjunto de dados é alimentada pelo modelo de treinamento), 16 batch size (número de exemplos de treinamento em uma passagem de época) e uma taxa de learning rate de 0,001 (taxa de aprendizagem que a rede neural atualiza os conceitos que aprendeu). O modelo foi dividido em quatro classes e exportado como tflite como mostra a Figura 6.

Figura 6 – Treinamento do modelo.



Fonte: Autoria própria.

3.2.4 FLUTTER

Criado pelo Google e lançado em maio de 2017, o Flutter é um Framework open source para o desenvolvimento de aplicativos mobile para Android e iOS. A versão beta do Flutter foi lançada em 13 de março de 2018 e foi a primeira a ser lançada em 4 de dezembro de 2018. Flutter foi lançado como um projeto chamado Sky, que no início funcionava apenas no Android.

De acordo com a documentação: “Flutter é uma estrutura de código aberto do Google para criar aplicativos multiplataforma bonitos e compilados nativamente a partir de uma única base de código.” (FLUTTER, 2022). O objetivo do Flutter é permitir que os desenvolvedores compilem para cada plataforma usando sua própria camada gráfica renderizada pelo mecanismo Skia. Ao criar um aplicativo com o Flutter, o código é compilado para a linguagem base do dispositivo.

O Flutter suporta o uso de Packages(pacotes) compartilhados contribuídos por outros desenvolvedores para os ecossistemas Flutter e Dart. Permitindo construir rapidamente um aplicativo sem ter que desenvolver tudo do zero. Um pacote pode conter dependências (listadas no pubspec), bibliotecas Dart, aplicativos, recursos, testes, imagens e exemplos (FLUTTER, 2022). Para adicionar um Package a um aplicativo é necessário abrir o arquivo pubspec.yaml localizado dentro da pasta do aplicativo e adicionar em dependências. Os packages utilizados no projeto foram (Figura 7):

- **flutter_svg**: biblioteca de widgets e renderização SVG para Flutter.
- **image_picker**: plugin do Flutter para selecionar imagens da galeria e tirar fotos com a câmera.
- **tflite**: plugin do Flutter para acessar a API do Tensorflow Lite. Suporta classificação de imagens, detecção de objetos, Pix2Pix e Deeplab e PoseNet em iOS e Android.

Figura 7 – Dependências do aplicativo.

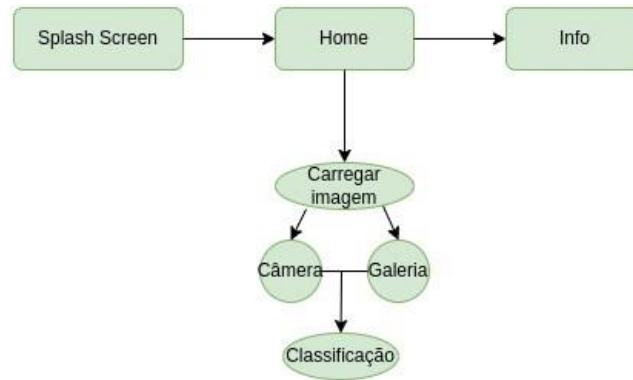
```
dependencies:  
  flutter:  
    sdk: flutter  
  
  # The following adds the Cupertino Icons font to your application.  
  # Use with the CupertinoIcons class for iOS style icons.  
  cupertino_icons: ^1.0.2  
  image_picker: ^0.8.4+4  
  tflite: ^1.1.2  
  flutter_svg: ^1.1.6
```

Fonte: Autoria própria.

4 RESULTADOS E DISCUSSÕES

A Figura 8 a seguir mostra o mapeamento do fluxo das telas do aplicativo. As setas indicam a passagem de tela e conectam os pontos do fluxo.¹⁷

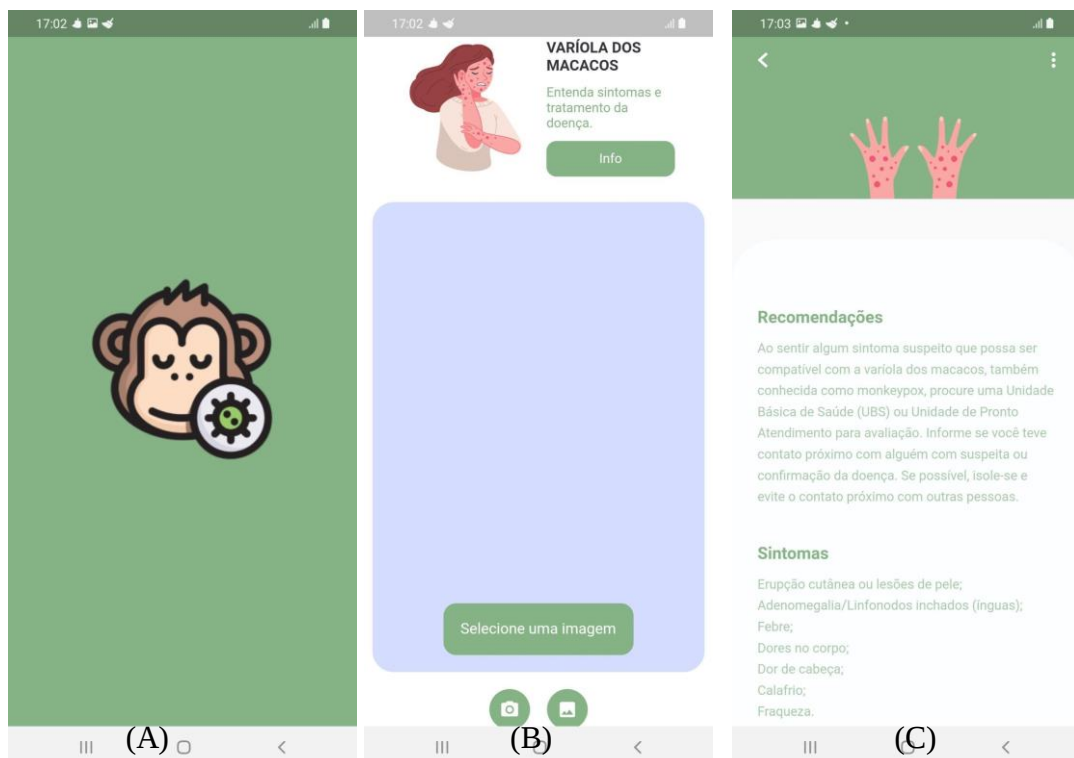
Figura 8 – Fluxo do aplicativo.



Fonte: Autoria própria.

A Splash Screen ou tela de abertura, é a primeira tela ao abrir o aplicativo e permanece visível por 3 segundos mostrando o ícone do aplicativo. Figura 9 (A). A tela Home, mostrada na Figura 9 (B), consiste na tela principal do aplicativo. Possui um card com texto e um botão que redireciona para a página de informações. Abaixo, existem um box para fazer a classificação e dois ícones na parte inferior, um para abrir a câmera e outro para ter acesso à galeria. A Figura 9 (C) é a página de informações sobre a varíola dos macacos.

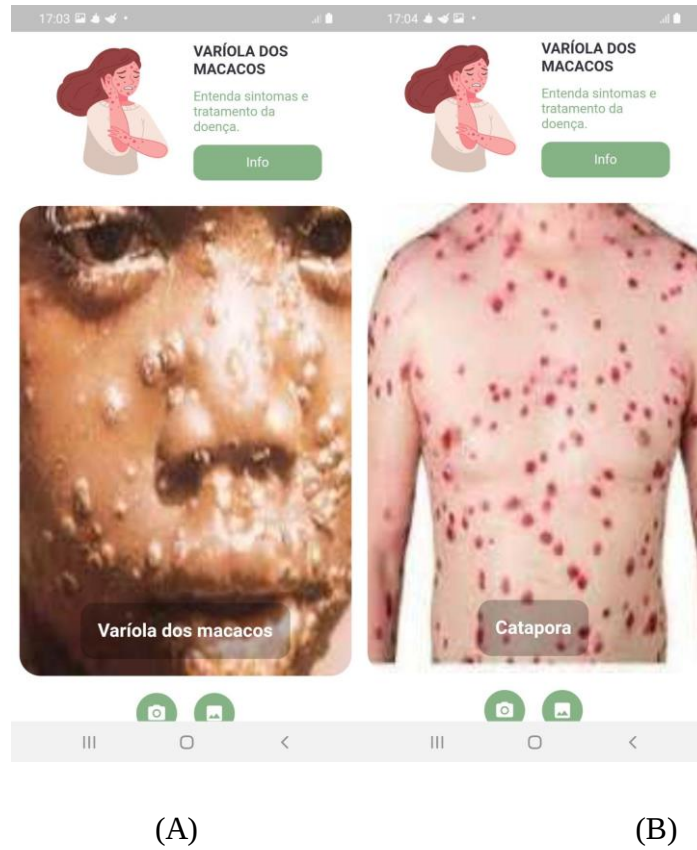
Figura 9 – Telas iniciais. Splash Screen (A), Tela Home (B), Tela Info (C).



Fonte: Autoria própria.

A Figura 10 mostra a classificação, na Figura A foi identificado a Varíola dos macacos, já na Figura B foi classificado Catapora.

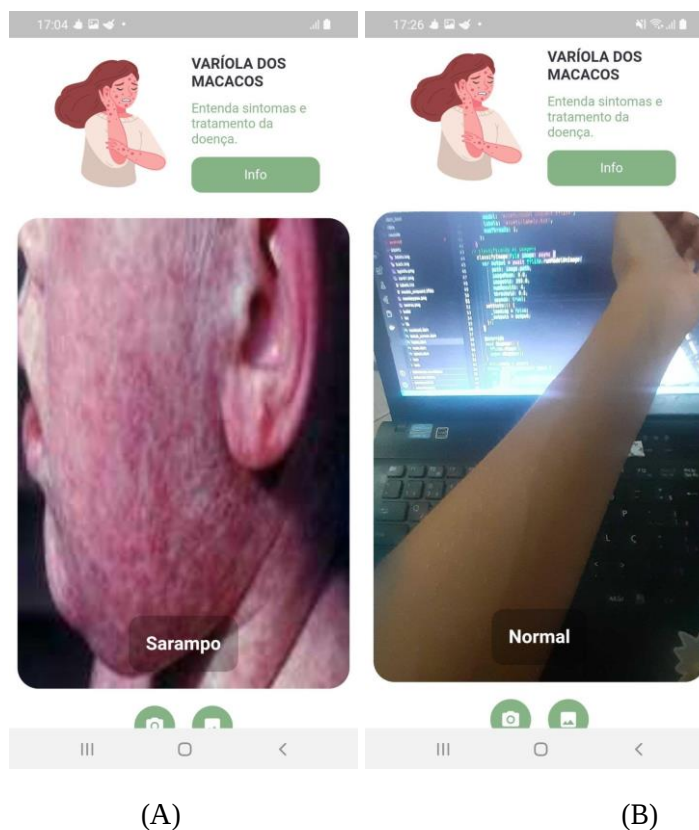
Figura 10 – Telas de classificação. Varíola dos macacos (A), Catapora (B).



Fonte: Autoria própria.

Na Figura 11 (A) foi identificado Sarampo através de imagens da galeria. Já na Figura 11 (B) foi feito o uso da câmera para fazer a identificação da classe Normal.

Figura 11 – Telas de classificação 2. Sarampo (A), Normal (B).



Fonte: Autoria própria.

No conjunto de testes, a precisão de detecção de Varíola dos macacos é de 0.99, para Catapora foi 0.82, já para Sarampo foi 0.93 e a identificação Normal é de 1.00 (Figura 12). A accuracy é a porcentagem da pontuação que o modelo obtém durante o treinamento. Se o modelo classifica diretamente 70 de 100 amostras, a precisão é $70/100 = 0,7$. Se a previsão do modelo é perfeita, a precisão é 1, caso contrário, a precisão é inferior a 1. É possível ver a precisão das classes ao longo de 100 épocas nos gráficos.

Figura 12 – Acurácia por classe.

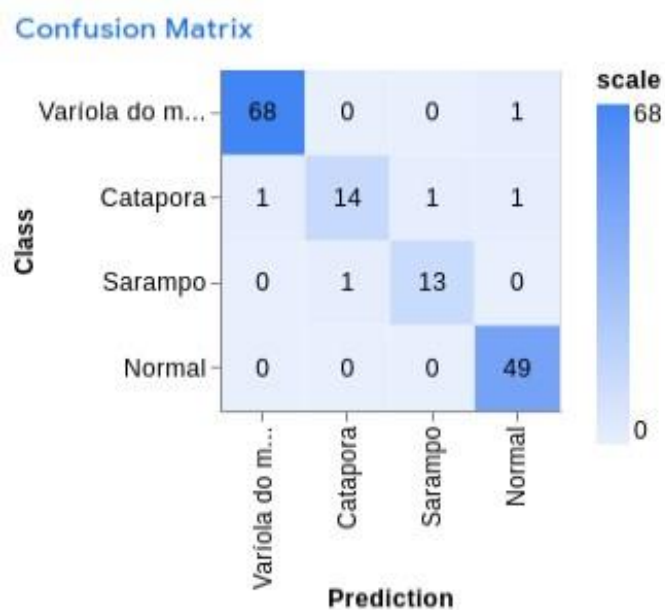
Accuracy per class ?

CLASS	ACCURACY	# SAMPLES
Varíola do macaco	0.99	69
Catapora	0.82	17
Sarampo	0.93	14
Normal	1.00	49

Fonte: Autoria própria.

A matriz de confusão resume a acurácia da previsão do modelo. Essa matriz ajuda a descobrir com quais classes o modelo fica confuso. Na matriz da Figura 13, a classe das amostras é representada no eixo y e a previsão no eixo x. Para a variola dos macacos foram utilizadas 69 amostras (Figura 12). Das 69 amostras o modelo classificou 68 como variola dos macacos e em 1 o modelo classificou incorretamente (Figura 13).

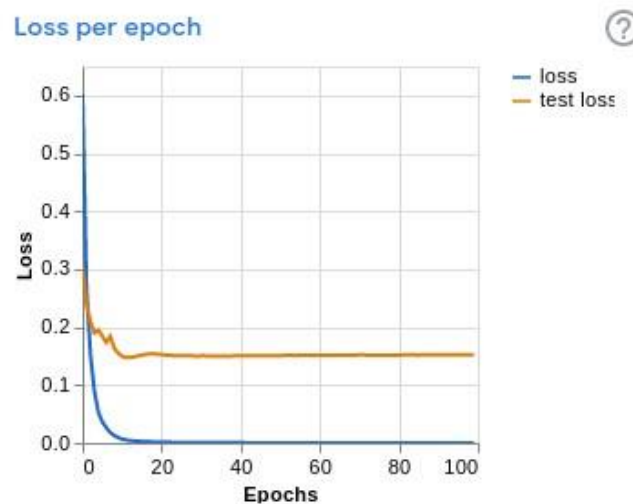
Figura 13 – Matriz de confusão.



Fonte: Autoria própria.

A perda por época avalia quão bem um modelo aprendeu a prever as classificações corretas para um determinado conjunto de amostras (Figura 14). Quando as previsões do modelo são perfeitas, a perda é zero, caso contrário, a perda é maior que zero.

Figura 14 – Perda por época.



onte: Autoria própria.

5 CONSIDERAÇÕES FINAIS

Neste trabalho foi apresentada uma aplicação desenvolvida em Flutter para executar um modelo de aprendizado de máquina treinado com o TensorFlow Lite, que pode ser utilizado para detecção ou predição a partir de imagens. Por meio desse aplicativo, foi possível construir uma tecnologia informacional e classificatória, item fundamental na segurança e qualidade da assistência à saúde. Diante dos resultados obtidos, o método se mostra promissor em cenários onde são necessárias respostas rápidas e com acurácia.

Dada a falta de conjuntos de dados de imagens relacionados aos vírus usados neste estudo, uma restrição encontrada foi a quantidade limitada de dados de imagem para classificar as lesões/erupções cutâneas utilizadas. Para trabalhos futuros, visa verificar o conteúdo e a aplicabilidade da aplicação junto a especialistas para garantir uma técnica mais adequada e confiável. Pretende-se também uma expansão do conjunto de dados de imagens para amplificar a precisão e o desempenho do modelo e a implementação de novos módulos para a detecção e predição de outras doenças.

REFERÊNCIAS

BELTRAN, Rafael; FELIX, Eduardo Bianchin; THIELO, Marcelo Resende. TENSOR FLOW FOR POETS: CLASSIFICAÇÃO DE ERVAS CULTIVADAS EM CASA. Anais do Salão Internacional de Ensino, Pesquisa e Extensão, v. 9, n. 2, 2017.

BUTANTAN. **Variola dos macacos: sintomas, transmissão, origem e número de casos são atualizados pela OMS.** Disponível em:

<https://butantan.gov.br/noticias/variola-dos-macacos-sintomas-transmissao-origem-e-numero-de-casos-sao-atualizados-pela-oms>. Acesso em: 22 nov. 2022.

CAMARGO FILHO, Stênio Marcos Rodrigues. **Aplicativo mobile para gestão de agendamento, prontuário e receituário.** 2021. Trabalho de Conclusão de Curso. Universidade Tecnológica Federal do Paraná.

CARVALHO, NICOLE DA SILVA. APLICATIVO MÓVEL PARA AUXÍLIO NO DIAGNÓSTICO DO PACIENTE DURANTE O ATENDIMENTO MÉDICO REMOTO. 2021.

COUTINHO, Helio Pereira et al. OS APLICATIVOS PARA DISPOSITIVOS MÓVEIS: breves considerações conceituais do processo de desenvolvimento. In: **10ª JICE-JORNADA DE INICIAÇÃO CIENTÍFICA E EXTENSÃO.** 2019.

DOS SANTOS MONTAN, Jardel et al. Avaliação de Plataformas Híbridas para Desenvolvimento de Aplicações para Dispositivos Móveis. **Seminários de Trabalho de Conclusão de Curso do Bacharelado em Sistemas de Informação**, v. 2, n. 1, 2018.

EL-KASSAS, Wafaa S. et al. Taxonomy of cross-platform mobile applications development approaches. **Ain Shams Engineering Journal**, v. 8, n. 2, p. 163-190, 2017.

FLUTTER. **Adding interactivity to your Flutter app**. Disponível em: <https://docs.flutter.dev/development/ui/interactive>. Acesso em: 13 jun. 2022.

FLUTTER. **Build apps for any screen**. Disponível em: <https://flutter.dev/>. Acesso em: 13 jun. 2022.

FLUTTER. **Using packages**. Disponível em: <https://docs.flutter.dev/development/packages-and-plugins/using-packages>. Acesso em: 11 nov. 2022.

GÉRON, Aurélien. **Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow**. Alta Books, 2019.

GIL, Antonio Carlos et al. **Como elaborar projetos de pesquisa**. São Paulo: Atlas, 2002.

GUIMARÃES FILHO, Marcos G.; LOURENÇO, Rafael P.; DO NASCIMENTO, Anderson S. Prisma: Um classificador de imagens para aprendizado de idiomas. In: **Anais da VI Escola Regional de Sistemas de Informação do Rio de Janeiro**. SBC, 2020.

HOPE, Tom; RESHEFF, Yehezkel S.; LIEDER, Itay. **Learning tensorflow: A guide to building deep learning systems**. "O'Reilly Media, Inc.", 2017.

ISLAM, Towhidul et al. A Web-scrapped Skin Image Database of Monkeypox, Chickenpox, Smallpox, Cowpox, and Measles. **bioRxiv**, 2022.

KREUTZ, L. C.; REZENDE, M. A.; MATÉ, Y. A. VARÍOLA DOS MACACOS (MONKEYPOX VIRUS-POXVIRIDAE): UMA BREVE REVISÃO. **Ars Veterinaria**, v. 38, n. 3, p. 111-115, 2022.

MACHADO, Henrique. Aplicações Móveis: Nativas ou Web?. **DevMedia**. 2014. Disponível em: <https://www.devmedia.com.br/aplicacoes-moveis-nativas-ou-web/30392>. Acesso em: 3 jul. 2022.

MACHIAVELI, Emanuel A.; MARCIANO, Juan Morysson Viana; DOS SANTOS, Felipe Gonçalves. Snake Classifier: Aplicativo mobile para classificação de serpentes peçonhentas. In: **Anais do XII Workshop de Computação Aplicada à Gestão do Meio Ambiente e Recursos Naturais**. SBC, 2021. p. 147-156.

Ministério da Saúde (MS). **Boletim Epidemiológico de Monkeypox nº 16 (COE)**. Disponível em: <https://www.gov.br/saude/pt-br/centrais-de-conteudo/publicacoes/boletins/epidemiologicos/variola-dos-macacos/boletim-epidemiologico-de-monkeypox-no-16-coe/view>. Acesso em: 1 dez. 2022.

Ministério da Saúde (MS). **Card Situação Epidemiológica de Monkeypox no Brasil nº120**. Disponível em: <https://www.gov.br/saude/pt-br/composicao/svs/resposta-a-emergencias/coes/monkeypox/atualizacao-dos-casos/card-situacao-epidemiologica-de-monkeypox-no-brasil-no-120/view>. Acesso em: 5 dez. 2022.

OLIVEIRA, Fabiano Pereira de et al. TMIC-Uma extensão do App Inventor para a implantação de modelos de ML voltados a classificação de imagens treinados no Teachable Machine. 2022.

Organização Pan-Americana da Saúde (OPAS). **Diretor-geral da OMS declara que surto de monkeypox constitui uma emergência de saúde pública de importância internacional**. Disponível em: <https://www.paho.org/pt/noticias/23-7-2022-diretor-geral-da-oms-declara-que-surto-monkeypox-constitui-uma-emergencia-saude>. Acesso em: 19 nov. 2022.

Organização Pan-Americana da Saúde (OPAS). **Casos de varíola dos macacos - Região das Américas**. Disponível em: <https://shiny.pahobra.org/monkeypox/>. Acesso em: 5 dez. 2022.

SACHINDANA, Sudara. Introduction to Flutter. **Dev**. 22 de jun. de 2021. Disponível em: <https://dev.to/sudarasach/intro-to-flutter-2odk>. Acesso em: 13 jun. 2022.

STATISTA. **Number of smartphone subscriptions worldwide from 2016 to 2027**. Disponível em: <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>. Acesso em: 5 jul. 2022.

STATISTA. **Share of population in Brazil that use a smartphone from 2015 to 2026**. Disponível em: <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>. Acesso em: 5 jul. 2022.

TENSORFLOW. **For Mobile & Edge**. Disponível em: <https://www.tensorflow.org/lite>. Acesso em: 20 nov. 2022.

World Health Organization (OMS). **2022 Monkeypox Outbreak: Global Trends**. Disponível em: https://worldhealthorg.shinyapps.io/mpx_global/. Acesso em: 5 dez. 2022.

World Health Organization (OMS). **2022 Monkeypox**. Disponível em: https://www.who.int/health-topics/monkeypox#tab=tab_1. Acesso em: 19 nov. 2022.

ZACCONE, Giancarlo. **Getting started with TensorFlow**. Birmingham: Packt Publishing, 2016.

GLOSSÁRIO

API: conjunto de padrões que permitem que dois componentes de software se comuniquem. **Backend:** é a estrutura que possibilita a operação do sistema.

Framework: conjuntos de bibliotecas ou componentes que permitem o desenvolvimento de sistemas e aplicações.

Frontend: é responsável pela parte visual, como apresentação, design, linguagens, cores, entre outros.

Open Source: código-fonte disponibilizado gratuitamente para modificação e redistribuição.