

16º Congresso de Inovação, Ciência e Tecnologia do IFSP - 2025

DETECÇÃO E CLASSIFICAÇÃO DE AEDES AEGYPTI UTILIZANDO VISÃO COMPUTACIONAL

GUILHERME N. MARQUES¹, EDUARDO A. MOSSIN²

¹ Graduando em Engenharia Elétrica, Bolsista FAPESP, IFSP, Campus Sertãozinho, nunes.marques@aluno.ifsp.edu.br

² Docente e orientador do Projeto FAPESP, IFSP, Campus Sertãozinho, emossin@ifsp.edu.br

Área de conhecimento (Tabela CNPq): 1.03.03.04-9 Sistemas de Informação

RESUMO: O *Aedes aegypti* representa uma ameaça relevante à saúde pública no Brasil, sendo vetor de doenças como dengue, zika e chikungunya. O controle eficaz desse mosquito é essencial para prevenir surtos e proteger a população. Este projeto propõe o desenvolvimento de uma solução baseada em inteligência artificial, com foco na detecção e classificação de mosquitos, especialmente do gênero *Aedes* e *Culex*. Foi utilizado o algoritmo YOLOv11 em cinco versões (*nano*, *small*, *medium*, *large* e *extra-large*), treinado com um *dataset* contendo 4.998 imagens de mosquitos dos gêneros *Aedes* e *Culex*, anotadas com *bounding boxes* e rotuladas conforme o gênero. Os resultados, avaliados por métricas de precisão e confiabilidade, indicaram desempenho satisfatório. Contendo, imagens com ângulos ou cenários distintos dos presentes no *dataset* apresentaram queda de desempenho. A versão *extra-large* se destacou como a mais robusta entre as testadas. Conclui-se que a aplicação de algoritmos de visão computacional, como o YOLOv11, é promissora para auxiliar no monitoramento e controle do mosquito vetor, sendo recomendada a ampliação do treinamento com imagens mais variadas para melhoria dos resultados.

PALAVRAS-CHAVE: YOLO; detecção de mosquitos; *Culex*; *Aedes Aegypti*.

DETECTION AND CLASSIFICATION OF AEDES AEGYPTI USING COMPUTER VISION

ABSTRACT: *Aedes aegypti* poses a significant threat to public health in Brazil, serving as a vector for diseases such as dengue, Zika, and chikungunya. Effective control of this mosquito is essential to prevent outbreaks and protect the population. This project proposes the development of a solution based on artificial intelligence, focusing on the detection and classification of mosquitoes, particularly those of the *Aedes* and *Culex* genera. The YOLOv11 algorithm was employed in five versions (*nano*, *small*, *medium*, *large*, and *extra-large*), trained on a dataset containing 4,998 images of mosquitoes from the *Aedes* and *Culex* genera, annotated with bounding boxes and labeled according to genus. The results, evaluated using precision and reliability metrics, indicated satisfactory performance. However, images with angles or environments different from those in the dataset showed reduced accuracy. The extra-large version stood out as the most robust among those tested. It is concluded that the application of computer vision algorithms such as YOLOv11 is promising for supporting the monitoring and control of mosquito vectors, with further training using more diverse datasets recommended to improve results.

KEYWORDS: YOLO; mosquito detection; *Culex*; *Aedes aegypti*.

INTRODUÇÃO

O controle de mosquitos vetores constitui um dos principais desafios contemporâneos para a saúde pública no Brasil e em países tropicais. De acordo com o Ministério da Saúde (BRASIL, 2025), o *Aedes aegypti* é o principal mosquito transmissor dos vírus da dengue, *chikungunya* e *Zika*. O mesmo ministério informa que em 2025 o Brasil teve 1.008 mortes confirmadas e outras centenas sob investigação (CNN BRASIL, 2025). Neste contexto, este projeto propõe a aplicação de inteligência artificial para a identificação de mosquitos dos gêneros *Aedes* e *Culex* por meio de técnicas de visão computacional. De forma mais específica, o este projeto aplica e avalia diferentes versões de modelos do *YOLOv11* em imagens reais, contribuindo com estratégias de monitoramento e controle do vetor.

A relevância dessa tarefa é reconhecida na literatura com diferentes abordagens já exploradas. Kittichai *et al.* (2021) e Butera *et al.* (2022) demonstraram o potencial do *YOLO* na classificação de mosquitos em imagens, evidenciando sua eficácia como tecnologia promissora, aspecto diretamente associado à solução proposta neste estudo. De forma complementar, Laranjeira *et al.* (2023) utilizaram IA para mapear focos de dengue a partir de imagens aéreas de drones, enquanto Gresse von Wangenheim *et al.* (2023) desenvolveram o aplicativo *XôDengue*, que emprega *deep learning* para classificar larvas de mosquitos. Embora tais iniciativas compartilhem o objetivo de aprimorar o monitoramento desses vetores de doenças, este projeto diferencia-se ao adotar como estratégia central a detecção e classificação de mosquitos adultos em imagens utilizando o *YOLOv11* em suas variações *nano*, *small*, *medium*, *large* e *extra-large*. Após a execução da metodologia proposta aqui, o estudo comprova a eficácia do *YOLO* na identificação de mosquitos. Destaca-se ainda a necessidade de ampliar o conjunto de dados para fortalecer a generalização e viabilizar aplicações em saúde pública.

MATERIAL E MÉTODOS

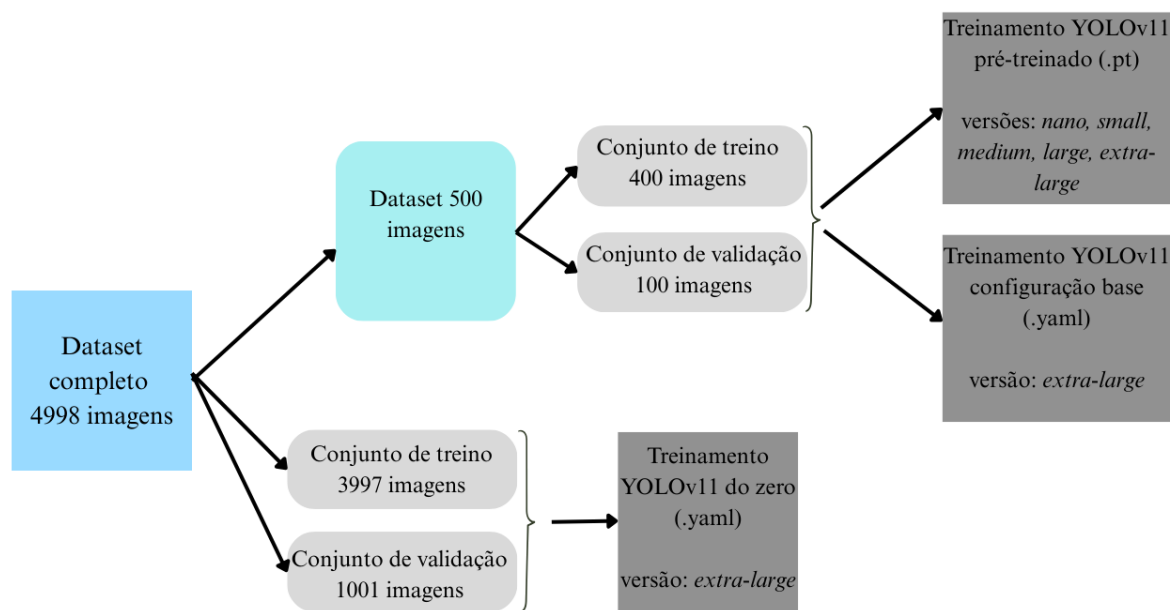
Antes de detalhar os materiais e métodos adotados neste projeto, apresenta-se a Figura 1, que ilustra de forma sintética as etapas de preparação e treinamento dos modelos, considerando o conjunto completo de 4.998 imagens. Cabe informar que estas imagens foram obtidas a partir do *dataset “6 Species Mosquito in Thailand for Classification”* (Pora, 2023).

Para os testes realizados, foram utilizadas as variações *nano*, *small*, *medium*, *large* e *extra-large* do *YOLO*. Essas correspondem a diferentes configurações arquiteturais do modelo, ajustadas principalmente em termos de profundidade da rede (número de camadas), largura (número de filtros por camada) e parâmetros treináveis totais. Em linhas gerais, a versão *nano* é a mais leve, projetada para dispositivos com recursos computacionais limitados, priorizando velocidade em detrimento da acurácia. As versões *small* e *medium* representam um equilíbrio entre desempenho e custo computacional, sendo adequadas a cenários de aplicação em tempo real com restrições moderadas. Já as versões *large* e *extra-large* possuem maior número de parâmetros, aumentando a capacidade de aprendizado e a robustez na detecção de objetos complexos ou em cenários mais desafiadores, ainda que com maior demanda de memória e processamento. Essa escalabilidade permite ao pesquisador selecionar o modelo mais adequado conforme o compromisso entre acurácia, velocidade e recursos disponíveis (Jocher *et al.*, 2023).

Inicialmente, foram extraídas 500 imagens desse *dataset* para compor um subconjunto destinado a testes preliminares. Esse subconjunto foi dividido em 400 imagens para treino e 100 para validação. Com ele, foram realizados dois tipos de treinamento com o modelo *YOLOv11*: (i) utilizando pesos pré-treinados com o *dataset COCO*, avaliando diferentes versões do modelo (*nano*, *small*, *medium*, *large* e *extra-large*); (ii) com o treinamento do modelo a partir de sua configuração base, ou seja, sem o uso de pesos pré-treinados e com inicialização com parâmetros aleatórios. Nessa segunda etapa, optou-se exclusivamente pela versão *extra-large* do *YOLOv11*, uma vez que os testes preliminares com os modelos pré-treinados indicaram que essa versão apresentou o melhor desempenho em termos de acurácia. Ressalta-se que, no contexto deste projeto, priorizou-se a maximização da taxa de acertos em detrimento da otimização do tempo de processamento, dada a natureza da aplicação.

Na etapa seguinte, foi utilizado o conjunto completo de dados (4998 imagens), dividido em 3997 imagens para treino e 1001 para validação. Esse conjunto foi usado para o treinamento final do modelo *YOLOv11* a partir do modelo base e na versão *extra-large*. Cada uma dessas fases será detalhada nos parágrafos seguintes.

FIGURA 1. Fluxograma do processo de treinamento do modelo.



FONTE: autoria própria

Para desenvolver um sistema de visão computacional, o primeiro passo é construir um banco de dados com imagens. Esse conjunto de dados será fundamental para que o sistema aprenda a identificar os padrões necessários ao reconhecimento dos mosquitos. Para que essa identificação seja precisa, é essencial contar com uma grande quantidade de imagens, possibilitando que o algoritmo compreenda todas as variações relevantes.

Para melhor performance nos treinamentos, foi decidido utilizar um computador do Instituto Federal de São Paulo - *Campus Sertãozinho*, localizado no Centro de Pesquisa e Inovação em Computação Aplicada à Indústria, equipado com uma GPU Nvidia RTX 3060. Foi necessária a configuração do ambiente de desenvolvimento para garantir o uso eficiente dos recursos de *hardware*. Foram instalados e configurados os seguintes *softwares*, bibliotecas *Python* e gerenciadores de pacotes: *Frameworks* como *PyTorch* e *Ultralytics*; *Drivers CUDA Toolkit* e *cuDNN* compatíveis com a GPU e ferramentas auxiliares como *Jupyter Notebook* e *PyCharm*. Além disso, foram realizados testes para verificar a compatibilidade do ambiente e garantir que a GPU seja devidamente utilizada nos treinamentos.

Como mostra a Figura 1, inicialmente, foi selecionada uma amostra composta por 500 imagens, distribuídas igualmente entre quatro espécies: 125 de *Aedes albopictus*, 125 de *Aedes aegypti*, 125 de *Culex quinquefasciatus* e 125 de *Culex vishnui*. Cada tipo de imagem foi armazenado em uma pasta separada. Em seguida, as imagens foram anotadas por meio da criação de *bounding boxes*. Após essa etapa, os treinamentos foram iniciados com as imagens organizadas em quatro pastas, classificadas por gênero e espécie, mas rotuladas apenas pelo gênero (*Aedes* ou *Culex*). A rotulação foi realizada com o auxílio da plataforma *Roboflow*. Essas imagens foram organizadas entre os conjuntos de treinamento (*train*) e validação (*val*).

Assim, foram realizados testes preliminares para comparar o desempenho das diferentes versões do modelo *YOLOv11* (*n*, *s*, *m*, *l* e *x*). Após essa etapa, já com a conclusão de que o modelo *extra-large* alcançou a maior taxa de acertos, foi realizado um novo treinamento, porém, utilizando um modelo a partir de sua configuração base, também na configuração *extra-large*.

Considerando que o modelo base na configuração *extra-large* alcançou melhores resultados, decidiu-se então utilizá-lo para a realização de testes com um *dataset* com maior número de imagens. Dessa maneira, além das 500 imagens utilizadas no teste anterior, foram selecionadas mais 4.500 imagens do mesmo conjunto de dados. Todas as imagens foram rotuladas utilizando a plataforma *Roboflow*. Houve alguns erros na rotulação de determinadas imagens, nos quais o *YOLO* não reconhecia os mosquitos presentes, tratando essas imagens como *background*. Após a remoção de estas imagens

do *dataset*, a base finalizou com um total de 4.998 imagens. Com essa base ampliada e organizada, foi realizado um novo ciclo de treinamento.

É importante destacar o uso de inteligência artificial generativa na elaboração deste texto, em conformidade com a Portaria Normativa nº 136/2025 do IFSP. Especificamente, utilizou-se o ChatGPT como ferramenta auxiliar para a revisão e o aprimoramento linguístico de trechos pontuais, sem prejuízo à autoria intelectual nem aos princípios de integridade acadêmica que orientam este trabalho.

RESULTADOS E DISCUSSÃO

A partir das análises realizadas sobre os diferentes modelos treinados nas distintas versões do conjunto de dados, foram feitas comparações com o objetivo de identificar o modelo mais adequado para o projeto. Observou-se que as variantes do modelo *YOLOv11*, pré-treinadas com o *dataset* COCO e treinado com o *dataset* composto por 500 imagens, apresentaram desempenhos variados, com resultados condizentes com seus respectivos níveis de complexidade e robustez, como pode ser visto na Tabela 1.

TABELA 1. Comparação das métricas dos modelos pré-treinados com o *dataset* COCO e treinado com o *dataset* composto por 500 imagens.

Modelo	Precision	recall	mAP@0.5	mAP@0.5:0.95	Val Class Loss
YOLOv11 Nano	0.99720	1.0	0.99500	0.98391	0.15856
YOLOv11 Small	0.99774	1.0	0.99500	0.98274	0.15897
YOLOv11 Medium	0.99944	1.0	0.99500	0.98549	0.14301
YOLOv11 Large	0.97997	1.0	0.99463	0.98554	0.12425
YOLOv11 Extra-Large	0.99948	1.0	0.99500	0.98454	0.14008

FONTE: autoria própria

As métricas analisadas na Tabela 1 incluem a *precision*, que mede a proporção de detecções corretas em relação ao total de detecções realizadas; e o *recall*, que indica a capacidade do modelo em detectar todos os objetos relevantes da imagem. A métrica mAP@0.5 representa a média da precisão em um limiar de IoU (*Intersection over Union*) de 0.5, enquanto a mAP@0.5:0.95 oferece uma avaliação mais rigorosa, considerando uma faixa de IoUs entre 0.5 e 0.95. Por fim, a *Val Class Loss* reflete o erro de classificação no conjunto de validação. Nesta métrica, quanto melhor seu valor, melhor. Diante dos resultados, optou-se pela utilização do modelo *YOLOv11 extra-large*, que apresentou excelente desempenho, com alta precisão (0.99948), *recall* perfeito (1.0) e uma das menores perdas de classificação (0.14008), além de ótimo desempenho em mAPs. Vale destacar que os testes foram realizados com um conjunto de apenas 500 imagens, o que pode beneficiar modelos menores com menor capacidade de generalização. No entanto, a escolha por um modelo mais robusto, como o *extra-large*, se justifica pela expectativa de que, com uma base de dados mais ampla, ele possa superar os demais de forma mais consistente, devido à sua maior capacidade de aprendizado e generalização.

A partir deste ponto, os experimentos concentraram-se em verificar o impacto do tipo de inicialização do modelo, comparando o desempenho obtido com pesos pré-treinados com treinamento do modelo a partir de sua configuração base. Foi utilizado o mesmo conjunto de imagens e a mesma configuração de treino do treinamento anterior. A Tabela 2 apresenta os resultados obtidos nessas duas abordagens.

TABELA 2. Comparação das métricas dos modelos treinados com ou sem pesos.

Modelo	Precision	recall	mAP@0.5	mAP@0.5:0.95	Val Class Loss
YOLOv11 Extra-Large (pré-treinado)	0.99948	1.0	0.99500	0.98454	0.14008
YOLOv11 Extra-Large (configuração base)	0.99742	1.0	0.99500	0.98183	0.13600

FONTE: autoria própria

Observa-se que ambas as abordagens atingiram resultados semelhantes, com mAP@0.5 idêntico (0.995) e *recall* máxima (1.0). No entanto, o modelo treinado a partir de sua configuração base apresentou uma *Val Class Loss* ligeiramente inferior (0.136 frente a 0.14008), além de manter uma mAP@0.5:0.95 competitiva (0.98183), o que indica um bom potencial de generalização, mesmo sem o uso de pesos previamente ajustados. Considerando esses resultados optou-se por seguir com os

experimentos subsequentes utilizando modelos treinados a partir de sua configuração base, permitindo maior controle sobre o processo de aprendizado e evitando possíveis vieses herdados de *datasets* genéricos usados no pré-treinamento.

Dando continuidade aos experimentos, foi realizado o treinamento final com o modelo *YOLOv11 extra-large* treinado a partir de sua configuração base, agora utilizando um conjunto ampliado de 4998 imagens. Essa etapa representa a fase conclusiva da avaliação prática do projeto, e seus resultados são apresentados a seguir, organizados na Tabela 3.

TABELA 3. Resultados do modelo treinado sem pesos.

Modelo	Precision	recall	mAP@0.5	mAP@0.5:0.95	Val Class Loss
YOLOv11 Extra-Large	0.99981	1.0	0.99500	0.98872	0.12625

FONTE: autoria própria

Os resultados obtidos no treinamento final demonstram um avanço em relação aos testes anteriores. Com o uso de um conjunto ampliado de 4998 imagens e mantendo o treinamento do zero com o modelo *YOLOv11 extra-large*, foi possível atingir um desempenho de alto nível, com mAP@0.5:0.95 de 0.98872, o mais elevado entre todos os experimentos realizados. A *precision* de 0.99981 e o *recall* de 1.0 reforçam a confiabilidade do modelo, enquanto a *Val Class Loss* de 0.12625 indica uma melhoria consistente na etapa de validação.

Além dos resultados obtidos no conjunto de validação, foram conduzidos testes com imagens inéditas ao modelo, com o objetivo de avaliar sua capacidade de generalização. A análise com essas imagens evidenciou algumas restrições na habilidade de generalização do modelo, conforme apresentado na Tabela 4.

TABELA 4. Métricas de generalização - 40 imagens externas.

Métricas	Valor	Porcentagem
Total de imagens	40	100%
Predições corretas	35	87,5%
Predições incorretas	5	12,5%
Acurácia Geral	35/40	87,5%

FONTE: autoria própria

A Tabela 5 apresenta o desempenho obtido em cada classe. O modelo atingiu 90,0% de acurácia na identificação do *Aedes* e 85,0% para *Culex*, com elevada confiança média nas predições (88,4% e 87,1%, respectivamente). Embora a taxa de erro tenha sido baixa (2/20 para *Aedes* e 3/20 para *Culex*), a diferença entre os resultados das classes e os erros observados apontam limitações quando aplicados em cenários não controlados.

TABELA 5. Desempenho por classe - Generalização.

Classe	Acurácia	Confiança Média*	Erros
Aedes	20	88,4%	2/20
Culex	20	87,1%	3/20

*A confiança média, no contexto desta tabela, é uma métrica que representa o nível médio de certeza que o modelo possui em suas predições ao longo de todo o conjunto de teste.

FONTE: autoria própria

CONCLUSÕES

Em síntese, este estudo não apenas valida a aplicação da visão computacional para identificar mosquitos, mas também estabelece uma estratégia de treinamento otimizada. A principal contribuição é demonstrar que o modelo *YOLOv11 extra-large*, treinado a partir de sua configuração base, alcança desempenho superior (maior mAP@0.5:0.95, *precision* e *recall*) em relação a modelos pré-treinados. Essa abordagem oferece uma solução mais precisa e menos suscetível a vieses, abrindo um caminho para o desenvolvimento de ferramentas de vigilância no combate a doenças como a dengue.

Para a implementação em larga escala, a próxima etapa é a expansão e diversificação do conjunto de dados. A incorporação de imagens com diferentes ângulos, iluminação e planos de fundo é

essencial para aprimorar a capacidade de generalização do modelo. Isso viabilizará o desenvolvimento de aplicações práticas, como sistemas de monitoramento em campo ou aplicativos para agentes de saúde, maximizando o impacto da pesquisa na saúde pública.

CONTRIBUIÇÕES DOS AUTORES

Eduardo A. Mossin contribuiu com a concepção e definição do escopo do estudo. Guilherme N. Marques realizou a curadoria e análise dos dados, bem como desenvolveu a metodologia e conduziu os experimentos.

Guilherme N. Marques e Eduardo A. Mossin atuaram na redação do trabalho, colaboraram na revisão crítica do conteúdo e aprovaram a versão submetida.

AGRADECIMENTOS

Agradeço à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) pelo financiamento e apoio à realização deste projeto. Estendo meus agradecimentos ao Instituto Federal de São Paulo – Campus Sertãozinho, pela infraestrutura fornecida e pelo incentivo contínuo à pesquisa.

Agradeço também à Prof.^a Fabiana Cristina de Freitas graduada em Licenciatura em Ciências Biológicas, mestre em Ciências e doutora em Agronomia, pelas orientações nos tipos de mosquitos presentes na região, contribuindo significativamente para o desenvolvimento deste trabalho.

REFERÊNCIAS

BRASIL. Ministério da Saúde. *Aedes aegypti*. Disponível em: <https://www.gov.br/saude/pt-br/assuntos/saude-de-a-a-z/a/aedes-aegypti>. Acesso em: 22 set. 2025.

BUTERA, L.; FERRANTE, A.; JERMINI, M.; PREVOSTINI, M.; ALIPPI, C. Precise agriculture: Effective deep learning strategies to detect pest insects. **IEEE/CAA Journal of Automatica Sinica**, v. 9, n. 2, p. 246-258, 2022. DOI: 10.1109/JAS.2021.1004317.

CNN BRASIL. **Brasil ultrapassa mil mortes por dengue em 2025**. [2025]. Disponível em: <https://www.cnnbrasil.com.br/nacional/brasil/brasil-ultrapassa-mil-mortes-por-dengue-em-2025/>. Acesso em: 21 set. 2025.

JOCHER, G. **YOLO by Ultralytics**. 2023. Repositório GitHub. Disponível em: <https://github.com/ultralytics/ultralytics>. Acesso em: 21 set. 2025.

KITTICHA, V.; PENGSAKUL, T.; CHUMCHUEN, K.; SAMUNG, Y.; SRIWICHA, P.; PHATTAMOLRAT, N.; TONGLOY, T.; JAKSUKAM, K.; CHUWONGIN, S.; BOONSANG, S. Deep learning approaches for challenging species and gender identification of mosquito vectors. **Scientific Reports**, v. 11, n. 4838, 2021. Disponível em: <https://doi.org/10.1038/s41598-021-84219-4>. Acesso em: 19 jul. 2025.

LARANJEIRA, C.; PEREIRA, M. B.; DE OLIVEIRA, R. V. F.; BARBOSA, G. L.; FERNANDES, C. M.; BERMUDI, P. M. M.; DE RESENDE, E. F. R.; FERNANDES, E. A. M.; NOGUEIRA, K.; ANDRADE, V. R.; QUINTANILHA, J. A.; DOS SANTOS, J. A.; CHIARAVALLOTI, F. **Automatic mapping of high-risk urban areas for *Aedes aegypti* infestation based on building facade image analysis**. 2023. *Preprint* (não paginado). Disponível em: <https://www.medrxiv.org/content/10.1101/2023.11.30.23298876v1>. Acesso em: 19 jul. 2025.

PORA, W.; KASAMSUMRAN, N.; THARAWATCHSARASART, K.; AMPOL, R.; SIRIYASATIEN, P.; JARIYAPAN, N. **6 Species Mosquito in Thailand for Classification**. 2023. *Dataset* (Conjunto de dados). Kaggle. Disponível em: <https://doi.org/10.34740/KAGGLE/DSV/5301446>. Acesso em: 19 jul. 2025.

WANGENHEIM, G. V. C.; MARTINS, R. M.; ARAUJO, P. P. **XÔDENGUE**. 2023. Patente: Programa de Computador. Número do registro: BR512023003755-9. Título: “XÔDENGUE”. Registro: INPI - Instituto Nacional da Propriedade Industrial, 12 dez. 2023.