THE MARKER: FERRAMENTA DE MARCAÇÃO DE IMAGEM ASSISTIDA POR INTELIGÊNCIA ARTIFICIAL

THE MARKER: ARTIFICIAL INTELLIGENCE-ASSISTED IMAGE MARKING TOOL

EL MARCADOR: HERRAMIENTA DE MARCADO DE IMÁGENES ASISTIDA POR INTELIGENCIA ARTIFICIAL

10.56238/MultiCientifica-032

Arthur Siqueira da Cunha
Graduando em Engenharia de Computação
Instituição: Faculdade Engenheiro Salvador Arena
E-mail: arthursiqcunha@gmail.com

Danilo Rodrigues Dantas
Graduando em Engenharia de Computação
Instituição: Faculdade Engenheiro Salvador Arena
E-mail: danilo_rodrigues90@hotmail.com

Maik Soares Luiz
Graduando em Engenharia de Computação
Instituição: Faculdade Engenheiro Salvador Arena
E-mail: maik.masl@gmail.com

Victor Inácio de Oliveira
Doutor em Engenharia de Controle e Automação
Instituição: Escola Politécnica da Universidade de São Paulo (USP)
E-mail: pro14724@cefsa.edu.br

RESUMO
O presente trabalho de conclusão de curso tem como principal objetivo desenvolver o The Marker, uma ferramenta de marcação de imagens assistida por inteligência artificial voltada à criação de conjuntos de dados personalizados para aplicações de visão computacional. O estudo fundamenta-se em conceitos de inteligência artificial, aprendizado de máquina, redes neurais profundas e ergonomia, destacando a importância da anotação de imagens na construção de modelos computacionais eficazes e os impactos físicos associados a atividades repetitivas, como LER, DORT e Síndrome de Visão computacional. A metodologia aplicada envolveu o desenvolvimento de uma aplicação modular composta por interface gráfica em React, processamento em Rust, execução do modelo Segment Anything Model por meio de scripts em Python e armazenamento seguro com criptografia AES-GCM. Foram realizados testes experimentais para avaliar precisão, tempo de inferência, quantidade de interações manuais necessárias e desempenho do sistema em diferentes resoluções de imagem. Os resultados indicam que a ferramenta reduz significativamente o esforço manual ao sugerir pontos de segmentação automaticamente, funcionando em ambiente offline e em máquinas com menor poder de

processamento, oferece experiência ergonômica aprimorada e demonstra potencial para acelerar a criação de bases de dados visuais de forma colaborativa.

Palavras-chave: Aprendizado de Máquina. Visão Computacional. Marcação de Imagem.

ABSTRACT
This undergraduate thesis aims to develop The Marker, an AI-assisted image-annotation tool designed to create customized datasets for computer-vision applications, grounding the study in concepts of artificial intelligence, machine learning, deep neural networks, and ergonomics while emphasizing the importance of image annotation in building effective computational models and the physical impacts associated with repetitive tasks such as RSI, WMSDs, and Computer Vision Syndrome. The applied methodology involved developing a modular application composed of a React graphical interface, processing modules in Rust, execution of the Segment Anything Model (SAM) via Python scripts, and secure storage with AES-GCM encryption; experimental tests were conducted to evaluate accuracy, interference time, the number of manual interactions required, and system performance across different image resolutions. The results indicate that the tool significantly reduces manual effort by automatically suggesting segmentation points, operates offline and on lower-powered machines, provides an improved ergonomic experience, and shows strong potential to accelerate the collaborative creation of visual datasets.

Keywords: Machine Learning. Computer Vision. Labeling.

RESUMEN
Este proyecto final de carrera tiene como objetivo desarrollar The Marker, una herramienta de marcado de imágenes asistida por IA para la creación de conjuntos de datos personalizados para aplicaciones de visión artificial. El estudio se basa en conceptos de inteligencia artificial, aprendizaje automático, redes neuronales profundas y ergonomía, destacando la importancia de la anotación de imágenes en la construcción de modelos computacionales efectivos y los impactos físicos asociados con actividades repetitivas, como RSI (lesión por esfuerzo repetitivo), WRULD (trastorno laboral de las extremidades superiores) y síndrome de visión artificial. La metodología implicó el desarrollo de una aplicación modular compuesta por una interfaz gráfica en React, procesamiento en Rust, ejecución del modelo Segment Anything mediante scripts de Python y almacenamiento seguro con cifrado AES-GCM. Se realizaron pruebas experimentales para evaluar la precisión, el tiempo de inferencia, el número de interacciones manuales requeridas y el rendimiento del sistema con diferentes resoluciones de imagen. Los resultados indican que la herramienta reduce significativamente el esfuerzo manual al sugerir automáticamente puntos de segmentación, funciona sin conexión y en máquinas con menor potencia de procesamiento, ofrece una experiencia ergonómica mejorada y demuestra potencial para acelerar la creación colaborativa de bases de datos visuales.

Palabras clave: Aprendizaje Automático. Visión Artificial. Etiquetado de Imágenes.

1 INTRODUÇÃO

A inteligência artificial (IA) surgiu como uma das tecnologias mais inovadoras e transformadoras do século XXI, e é definida como a capacidade de uma máquina em replicar o comportamento gerado pela inteligência humana (Wang *et al.*, 2024). Dentre os campos da inteligência artificial se destaca a visão computacional (*Computer Vision*, CV): o uso da inteligência artificial para processar e extrair as informações de uma imagem como o caso de LeNet que foi pioneira no reconhecimento de texto escrito à mão (Wang *et al*., 2024). As aplicações da visão computacional por inteligência artificial, como a classificação de imagens, são diversas (Khalil *et al.*, 2023): Diagnósticos médicos como a assistência na detecção de pneumonia (Kundu *et al.*, 2021), sistemas de navegação de veículos autônomos (Bojarski *et al.,* 2016), análise automática de serviços de segurança (Chen *et al*., 2025), pesquisa de produtos em comércios eletrônicos (Shin *et al.,* 2022) e até detecção de doenças em plantas na agricultura (Gohill *et al.,* 2024). A diversidade de aplicações apenas destaca o potencial da visão computacional e sua importância em área de estudo (Wang *et al.,* 2024). Existe uma variedade de modelos baseados em Redes Neurais Convolucionais (*Convolutional Neural Networks*, CNNs) e *Transformers* feitos para lidar com os diversos problemas encontrados na visão computacional (Wang *et al.,* 2024).

Um ingrediente fundamental para desenvolver os modelos de visão computacional é o conjunto de dados (*datasets*) utilizado em seu treinamento (Schuhmann *et al.,* 2022). Um exemplo é o *Microsoft Common Objects in Context* (MS COCO) que possui 328 mil imagens totalizando 2 milhões e 500 mil objetos marcados (Lin *et al.*, 2014). Esse processo ocorre com o modelo recebendo uma imagem como *input* oriunda do conjunto de dados, e resultando em um *output* com as informações extraídas da imagem de acordo com o modelo e no conhecimento adquirido de seu conjunto de dados de treinamento (Khalil *et al.*, 2023). Uma parte fundamental na criação desses conjuntos de dados é a marcação manual dessas imagens e objetos (Papadopoulos *et al*., 2017). Estudos indicam que tarefas repetitivas e prolongadas, como a marcação manual de imagens, estão associadas a um aumento significativo no risco de desenvolvimento de Lesões por Esforço Repetitivo (LER) (Kovashka *et al*., 2016). O uso de ferramentas de inteligência artificial na anotação de imagens pode melhorar a precisão e eficiência do processo de marcação, especialmente quando combinadas com a supervisão humana (Dutta e Zisserman, 2019). A implementação dessas tecnologias não apenas reduz o custo e o tempo associados à anotação de imagens, mas também minimiza a fadiga dos trabalhadores, promovendo um ambiente de trabalho mais saudável e produtivo (Kovashka *et al*., 2016). Além disso, interfaces colaborativas, como o *Fluid Annotation*, permitem uma interação mais intuitiva entre humanos e máquinas, onde o modelo de IA fornece sugestões que podem ser rapidamente ajustadas pelo anotador humano, resultando em um processo de anotação mais rápido e menos propenso a erros (Andriluka *et al*., 2018).

Esse cenário reflete uma movimentação crescente no mercado mundial, no qual empresas e instituições investem em soluções voltadas para marcação automatizada de imagens para acelerar o desenvolvimento de modelos de visão computacional. Como foi o caso do investimento da *National Geospatial-Intelligence Agency* (NGA) de US$ 700 milhões para o avanço da marcação de imagem automatizada (Defense Scoop, 2024). Tal como a Meta que anunciou iniciativas para expandir a coleta e o processamento de dados visuais em grande escala (Time, 2024). Existem ferramentas já desenvolvidas para esse processo, como LabelImg, Label Studio, CVAT, VIA etc. Entretanto, mapeamos as funcionalidades delas e analisamos que nenhuma contempla simultaneamente todas as qualidades desejadas entre suas funcionalidades.

Lesões por Esforço Repetitivo (LER) e Distúrbios Osteomusculares Relacionados ao Trabalho (DORT) são condições que afetam o sistema musculoesquelético, resultantes de tarefas laborais que exigem movimentos repetitivos, posturas inadequadas mantidas por longos períodos ou sobrecarga física (Sanar, 2024; Hagberg *et al*., 1995). Tais lesões são frequentemente diagnosticadas em profissionais que exercem funções como digitação, montagem em linhas de produção ou uso contínuo de dispositivos manuais (Armstrong *et al*., 1993; Hagberg *et al*., 1995). Os sintomas mais comuns incluem dor localizada, formigamento, redução de força muscular e limitação funcional, especialmente em membros superiores e coluna cervical (Sanar, 2024; Armstrong *et al*., 1993).

Outro efeito colateral pode ser a fadiga visual, também conhecida como Síndrome da Visão do Computador (SVC) ou fadiga ocular digital que é definida como um conjunto de sintomas visuais e oculares que surgem após longos períodos de foco em telas digitais, como computadores, *tablets* e *smartphones* (Sheppard e Wolffsohn, 2018). Os sintomas mais comuns incluem visão embaçada, olhos secos, dores de cabeça, cansaço ocular e desconforto geral ao redor da região ocular (Rosenfield, 2016). A condição é reconhecida como um problema crescente na era digital, especialmente entre trabalhadores que permanecem longas horas diante de telas (Gowrisankaran e Sheedy, 2015). As atividades com foco visual constante, como a rotulagem de dados, aumentam o esforço ocular e podem comprometer a precisão e o conforto do usuário (Coles-Brennan *et al.,* 2019). Estudos demonstram que a prevalência da fadiga ocular digital ultrapassa 75% em grupos específicos, como profissionais de tecnologia, o que reforça a importância dessas medidas preventivas para que o conforto visual e a precisão das tarefas sejam mantidos (Sheppard, 2018).

A Norma Regulamentadora nº 17 (NR-17) que trata da ergonomia no ambiente de trabalho estabelece parâmetros mínimos para a adaptação das condições de trabalho às características dos trabalhadores, especialmente em atividades que envolvem esforço repetitivo, como a digitação, a fim de preservar a saúde do trabalhador Entre suas recomendações, limita-se a 8.000 toques reais por hora de trabalho, visando prevenir a LER e DORT (Brasil, 2023). Nesse mesmo sentido, a Organização Mundial da Saúde (OMS) destaca que o uso prolongado de telas sem pausas pode causar fadiga ocular

e problemas musculoesqueléticos. Por isso, recomenda-se pausas regulares de 10 a 15 minutos a cada hora de uso contínuo de dispositivos digitais, como forma de prevenir esses efeitos (Organização Mundial da Saúde, 2020). Considerando assim que um usuário leva aproximadamente 4.000 horas para marcar 1.000.000 de objetos.

Com base nessas condições é proposto o The Marker, uma ferramenta de marcação de imagem assistida por inteligência artificial, com o objetivo de auxiliar a criação de conjuntos de banco de dados personalizados e criptografados. A proposta busca reduzir o número de toques reais por marcação de imagem, impactando diretamente no tempo total necessário para concluir a atividade, outro efeito desejado é o incentivo para equipes menores, como pesquisadores independentes e usuários com menor familiaridade tecnológica, consigam desenvolver seus próprios conjuntos de dados por meio de uma instalação simples e acessível com uma interface ergonômica, intuitiva e colaborativa.

2 REFERENCIAL TEÓRICO

2.1 INTELIGÊNCIA ARTIFICIAL E APRENDIZADO DE MÁQUINA

A inteligência artificial (IA) e o aprendizado de máquina (*Machine Learning*) constituem campos fundamentais da ciência da computação, voltados ao desenvolvimento de sistemas capazes de aprender, generalizar e tomar decisões com base em dados, essas tecnologias têm sido amplamente exploradas e se consolidado como pilares da transformação digital em diversos setores, impulsionando avanços em áreas como saúde, segurança, educação, transporte, automação industrial (Morais, 2020).

2.1.1 Conceitos fundamentais

A inteligência artificial (IA) pode ser definida como o campo da ciência da computação que busca simular processos cognitivos humanos por meio de sistemas computacionais. Essa área se divide em duas vertentes: a chamada IA forte, que procura reproduzir integralmente a cognição, e a IA fraca, voltada para tarefas específicas (Russell; Norvig, 2010). Embora a IA forte ainda esteja restrita a debates filosóficos, é a IA fraca que encontramos em aplicações práticas, como assistentes virtuais e sistemas de recomendação (Goodfellow; Bengio; Courville, 2016).

2.1.2 Aprendizado de máquina e aprendizado profundo

O aprendizado de máquina (*machine learning*) é uma das áreas centrais da IA, responsável por criar algoritmos que aprendem a partir de dados. Mitchell (1997) o define como a capacidade de um programa de computador melhorar seu desempenho em determinada tarefa conforme adquire experiência. Esse conceito abriu caminho para avanços em áreas como saúde e comércio eletrônico (Lecun; Bengio; Hinton, 2015).

Nos últimos anos, o aprendizado profundo (*deep learning*) se consolidou como a principal técnica, graças ao uso de redes neurais profundas. Goodfellow, Bengio e Courville (2016) explicam que esse modelo de aprendizado elimina a necessidade de criar manualmente atributos para os dados, permitindo resultados superiores em visão computacional, tradução automática e reconhecimento de fala. Esse tipo de abordagem é fundamental para o The Marker, pois possibilita implementar segmentação automática e sugerir anotações ao usuário (Krizhevsky; Sutskever; Hinton, 2012).

2.2 VISÃO COMPUTACIONAL

2.2.1 Métodos clássicos

A visão computacional é a área da IA que procura permitir que sistemas extraiam informações de imagens e vídeos. Durante muito tempo, essa tarefa foi realizada por métodos clássicos, como o SIFT e o SURF, projetados para identificar pontos de interesse em imagens com diferentes escalas e rotações (Szeliski, 2010). Apesar de eficazes em situações controladas, esses métodos exigiam parametrização manual e frequentemente falhavam em ambientes ruidosos (Everingham *et al*., 2010). Mesmo com limitações, esses algoritmos criaram a base para a adoção de abordagens mais modernas, principalmente as redes neurais convolucionais (Lecun; Bengio; Hinton, 2015).

2.2.2 Avanços com redes neurais

O grande salto ocorreu em 2012, quando a rede AlexNet superou métodos anteriores no desafio ImageNet, tornando-se um divisor de águas para a área (Krizhevsky; Sutskever; Hinton, 2012). Desde então, arquiteturas mais avançadas, como o Faster R-CNN (Ren *et al*., 2015) e o YOLO (*You Only Look Once*) (Redmon *et al*., 2016), passaram a dominar o cenário, permitindo reconhecimento em tempo real. Esses avanços impulsionaram aplicações em segurança, veículos autônomos e diagnóstico médico (Szeliski, 2010). Essas redes representam a base tecnológica de algoritmos de anotação assistida, o que viabiliza sugestões automáticas de marcação.

2.2.3 Intersection-over-Union

A métrica de *Intersection-over-Union* (IoU) avalia a sobreposição de duas áreas, sendo primeira área referente a inferência do modelo e a segunda área sendo a área de referência. Sobreposição 100% significa que as áreas são idênticas e sobreposição 0% indica que elas são totalmente diferentes.

2.2.4 Segment Anything Model (SAM)

A Meta AI apresentou o *Segment Anything Model 2.1* (SAM), que introduziu a segmentação generalista: um único modelo capaz de lidar com objetos variados sem a necessidade de treinamento específico (Kirillov *et al*., 2023). Essa característica reduz custos de preparação de dados e amplia as

possibilidades de uso em diferentes cenários. O SAM pode ser explorado para sugerir automaticamente regiões de interesse em imagens, reduzindo o esforço manual e aumentando a qualidade das anotações (Paszke *et al*., 2019).

2.3 ANOTAÇÃO DE IMAGENS EM INTELIGÊNCIA ARTIFICIAL

2.3.1 Importância da anotação

O processo de anotação de imagens é uma das etapas mais delicadas da visão computacional, pois define a qualidade dos dados usados no treinamento de modelos. Dados inconsistentes resultam em algoritmos pouco confiáveis, conforme a lógica "*garbage in, garbage out*" (Zhou, 2018). Sager, Janiesch e Zschech (2021) destacam que em muitos projetos de IA a anotação consome mais tempo que o desenvolvimento dos modelos. Isso é ainda mais evidente em áreas como a medicina, em que apenas especialistas podem realizar a rotulagem (Rajpurkar *et al*., 2017).

2.3.2 Ferramentas de anotação

Diversas ferramentas foram criadas para facilitar a anotação. O LabelImg é amplamente usado para marcação de *bounding boxes* (Tzelepis *et al*., 2021). O LabelMe, desenvolvido no MIT, oferece recursos mais flexíveis, permitindo a criação de polígonos para cenários que exigem precisão (Russell *et al*., 2008). Já o VGG Image Annotator (VIA) é leve e roda diretamente no navegador, sem necessidade de instalação (Dutta; Zisserman, 2019). Outras soluções buscam juntar a inteligência artificial ao processo, como o Fluid Annotation (Andriluka; Uijlings; Ferrari, 2018) e o Extreme Clicking (Papadopoulos *et al*., 2017). Essas iniciativas apontam para uma tendência de colaboração entre humanos e máquinas na anotação de dados.

2.3.3 Formatos de datasets

Os formatos de anotação também têm papel crucial. O Pascal VOC organiza informações em XML (Everingham *et al*., 2010), enquanto o COCO utiliza JSON para permitir descrições mais ricas, incluindo segmentações e pontos-chave (Lin *et al.*, 2014). Por outro lado, o YOLO adota TXT com coordenadas normalizadas (Redmon *et al*., 2016). Ao oferecer suporte a múltiplos formatos, o The Marker possibilita maior flexibilidade e compatibilidade com diferentes fluxos de trabalho em IA (Abadi *et al.*, 2016).

2.4 ERGONOMIA E SAÚDE OCUPACIONAL

Atividades repetitivas como a anotação manual podem causar efeitos negativos à saúde, incluindo Lesões por Esforço Repetitivo (LER) e Distúrbios Osteomusculares Relacionados ao Trabalho (DORT). Hagberg, Silverstein e Wells (1995) já destacavam esses riscos em atividades de

digitação e uso contínuo de dispositivos. No Brasil, a Norma Regulamentadora nº 17 estabelece parâmetros de ergonomia para prevenir tais condições (Brasil, 2023). A Organização Mundial da Saúde (OMS, 2003) recomenda pausas regulares a cada hora de uso de dispositivos digitais

Outro problema recorrente é a fadiga ocular digital, também chamada de *Computer Vision Syndrome*. Segundo estudos, apontam que mais de 70% dos trabalhadores de TI relatam sintomas como dores de cabeça e visão embaçada (Rosenfield, 2016; Sheppard; Wolffsohn, 2018). A proposta do projeto procura minimizar esses efeitos ao reduzir o número de cliques e interações manuais, contribuindo para um ambiente de trabalho mais saudável (Merrill; Alleman, 2012).

Com base nos conceitos apresentados sobre inteligência artificial, visão computacional, anotação de imagens e ergonomia, definiu assim a arquitetura proposta para o The Marker. A integração desses fundamentos permitiu projetar uma ferramenta capaz de unir eficiência computacional e o bem-estar do usuário, conciliando aspectos técnicos de processamento e segmentação de imagens com princípios ergonômicos e de acessibilidade.

3 METODOLOGIA

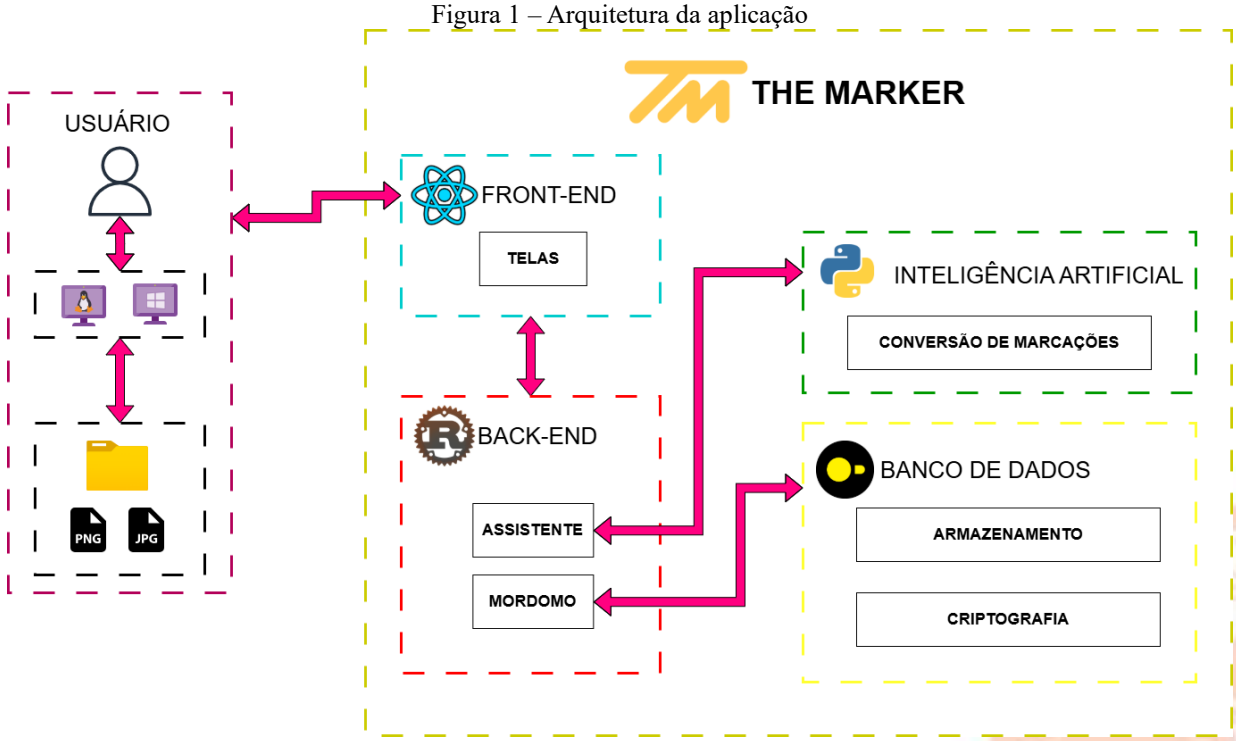
A arquitetura do projeto conforme apresentada na Figura 1 foi definida para funcionar de maneira modular, com módulos de *front-end*, *back-end*, inteligência artificial e banco de dados. Fazendo com que eles se conectem de forma agnóstica ao não estarem conectados como um pedaço monolítico ou homogêneo de código, mas criando um cenário que eles se comuniquem por meio de APIs para que cada módulo possa utilizar as linguagens de programação e *frameworks* adequados para suas funções específicas.

O *front-end* foi desenvolvido com React por suas funcionalidades de criação de interfaces modernas e dinâmicas, também por sua presença no desenvolvimento Web que oferece muito material de estudos e suporte de sua utilização. Além da presença da comunidade que disponibiliza bibliotecas funcionais e gráficas que facilitam o desenvolvimento. Utilizada para criar elementos como a tela de deslize infinito com a biblioteca Konva. A interface se comunica diretamente com a camada de processamento por meio do Tauri, que atua interligando o *front-end* com a camada *back-end*, desenvolvido em Rust. Na camada *back-end*, se encontram as operações lógicas do sistema em questão da comunicação entre os módulos. Gerenciando e permitindo que eles se comuniquem.

A camada do banco de dados é estruturada utilizando o DuckDB, onde os dados são armazenados nas tabelas estruturadas e consultados através de comandos SQL. Assim cada projeto gera um arquivo único que pode ser armazenado com ou sem criptografia, promovendo alto desempenho e portabilidade. Assim, futuras expansões podem ser implementadas isoladamente, sem comprometer a estrutura de outros projetos já existentes quando estes forem criados, atualizados ou excluídos.

A inteligência artificial existe como uma camada de *scripts* em linguagem Python que foram compilados em binários e são consultados pelo processo principal através de subprocessos do sistema operacional. Foi escolhido o modelo SAM por sua capacidade de propor máscaras de área preenchida por objetos sem um treinamento prévio. Capacidade conhecida como *zero-shot*.

Figura 1 – Arquitetura da aplicação



Fonte: Autoria própria (2025)

Esses componentes e módulos são englobados pelo *framework* Tauri, que foi escolhido por apresentar menor tamanho no arquivo compilado e melhor desempenho quando comparado a alternativas como Flutter e Electron, conforme demonstrado no Quadro 1. Os critérios analisados incluíram: ano de lançamento, popularidade entre desenvolvedores, tamanho dos arquivos gerados e o número de dependências necessárias. Durante a configuração do Tauri, definiu-se a utilização do React com JavaScript.

Quadro 1 – Comparação de *frameworks* ordenado por ano de publicação

Framework	Ano de publicação	Estrelas no GitHub	Downloads semanais	Arquivos gerados	Tamanho do arquivo
Tauri	2022	92 mil	130 mil	1	2 MB
Flutter	2017	170 mil	?	100	50 MB
Electron	2013	117 mil	831 mil	20	47 MB

Fonte: Autoria própria (2025)

Para o gerenciamento local de dados, optou-se pelo uso do DuckDB por ser um projeto *open source*, diferentemente de sistemas tradicionais como MySQL ou PostgreSQL, pode ser executado localmente e sem necessidade de servidores externos, essa decisão contribuiu para portabilidade e

independência do sistema, permitindo que o usuário manipule os dados de forma individual, sem instalações adicionais. O projeto foi compilado como um aplicativo *desktop* multiplataformas compatível com Windows e Linux-Debian, garantindo ampla acessibilidade e praticidade de uso pelos usuários finais. Todas as definições das ferramentas utilizadas em cada módulo estão apresentadas no Quadro 2.

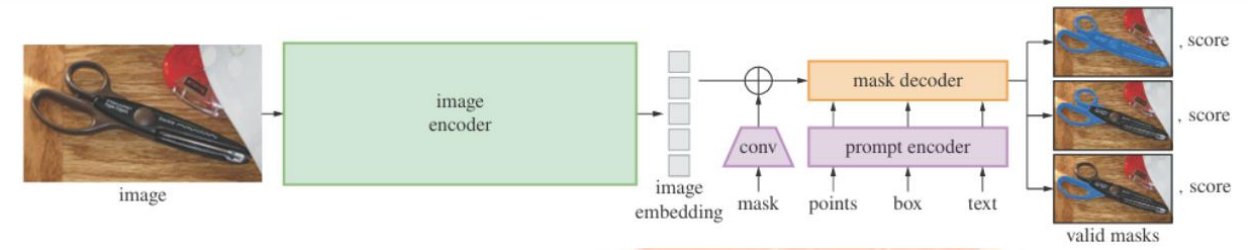
Quadro 2 – Lista de escolhas por função

Função	**Ferramenta optada**
Front-End	React
Back-End	Rust
Banco de dados	DuckDB
Sistema operacional	Windows e Linux-debian

Fonte: Autoria própria (2025)

O modelo SAM foi desenvolvido pela equipe da Meta AI (anteriormente Facebook AI Research) e disponibilizado em 2023. O modelo foi treinado com mais de 1 bilhão de máscaras em 11 milhões de imagens que utiliza uma arquitetura baseada em Pytorch incorporando um *image encoder* (baseado no Vision Transformer – ViT) e um *prompt encoder* conforme Figura 2 que permite ao modelo receber instruções via pontos, caixas delimitadoras ou máscaras anteriores. Essa versatilidade permite que o SAM realize segmentações de forma interativa e em tempo real, mesmo sem treinamento para tarefas específicas, diferentemente de modelos convencionais que apresentavam limitações em cenários não previamente treinados. O SAM mostrou-se versátil e robusto, permitindo maior autonomia na marcação das imagens. Para manipular as imagens e alimentar o modelo a fim de inferência foi utilizado o suporte oficial ao modelo através da biblioteca Transformers.

Figura 2 – Arquitetura do SAM

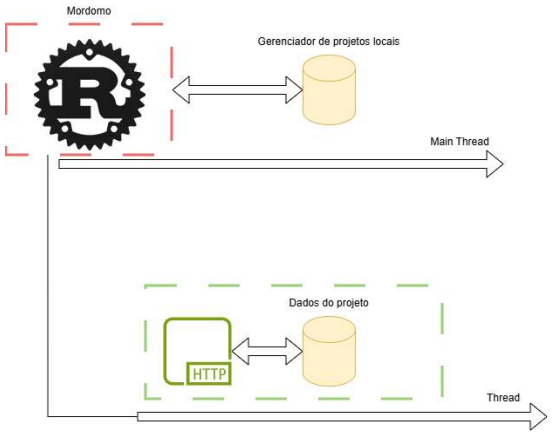


Fonte: Kirillov (2023)

Durante o desenvolvimento foram identificadas algumas limitações e dificuldades que impactaram na escolha do processo metodológico, o *framework* Electron não foi utilizado devido ao elevado consumo de memória e ao tamanho final dos pacotes gerados e o Flutter, no entanto apresentou uma menor maturidade em aplicações em *desktop*, além de exigir pacotes adicionais que aumentaram a complexidade da instalação.

Para que exista a opção de sincronização dos dados em rede local, é utilizado o sistema mordomo representado na Figura 3. Esse sistema existe como um *thread* assíncrono do sistema operacional que cria um servidor em uma porta de rede efêmera, permitindo que o *back-end* da máquina responsável e de outras máquinas na rede local se comuniquem através de requisições HTTP.

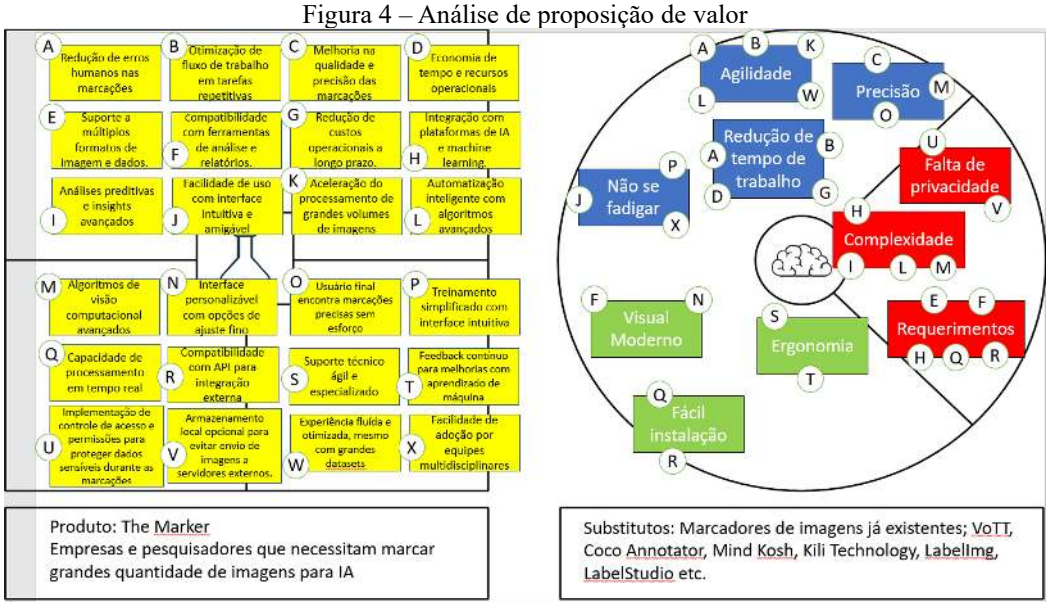
Figura 3 – Sistema mordomo



Fonte: Autoria própria (2025)

O projeto iniciou com a identificação da oportunidade de mercado voltada à criação de ferramentas de anotações mais intuitivas e eficientes, a metodologia foi estruturada inicialmente pela definição da arquitetura geral do sistema, estabelecendo os módulos essenciais. A partir dessa base da arquitetura, elaborou-se a proposição de valor do The Marker, apresentada na figura 4, que destaca a importância de reduzir o esforço operacional por meio de uma interface moderna, acessível e visualmente organizada, aliada a fluxos de interações simplificados, definiu-se como diretriz central o desenvolvimento de uma experiência que priorizasse a diminuição de movimentos repetitivos, a clareza visual e a facilidade de uso, permitindo que pesquisadores, estudantes e pequenas equipes criassem bases de dados personalizadas com menor carga física e cognitiva. Essa orientação metodológica guiou as etapas de design, usabilidade e implementação garantindo que o The Marker fosse concebido desde o início com foco em eficiência, conforto e acessibilidade.

Figura 4 – Análise de proposição de valor



Fonte: Autoria própria (2025)

Trazendo com isso a análise SWOT (ou FOFA) para analisar as forças, oportunidades, fraquezas e ameaças que a ferramenta poderia enfrentar uma vez disponibilizada. Conforme apresentado na Figura 5 é possível denotar que as forças do The Marker contêm a alta precisão na marcação e uma oportunidade surge do aumento de demanda da criação de conjuntos de dados visuais para processos de reconhecimento de imagens.

Figura 5 – Análise SWOT

FORÇAS	FRAQUEZAS
• Alta precisão na marcação automática. • Redução de tempo e custo operacional. • Uso de tecnologias modernas (Machine Learning, Visão Computacional). • fácil integração com bases de dados.	• Dependência de datasets bem rotulados. • Necessidade de hardware potente. • Alto custo de aquisição e atualização da IA. • Dificuldade de interpretar erros de marcação.
OPORTUNIDADES	**AMEAÇAS**
• Demanda crescente da IA em agricultura, indústria e segurança. • Parcerias com empresas que usam grandes volumes de imagens. • Expansão internacional em mercados emergentes. • Crescimento do mercado de Inteligência Artificial Generativa.	• Aparecimento de concorrentes com soluções mais baratas. • Rápida evolução tecnológica. • Questões regulatórias (proteção de dados, LGPD/GDPR). • Barreiras culturais.

Fonte: Autoria própria (2025)

Considerando esses aspectos, criou-se uma tabela de comparação das propostas do The Marker com elas já implementadas por produtos implementados no mercado. No Quadro 3 podemos analisar

a relação dos projetos externos em relação à sua instalação (o nível de complexidade para instalar ou utilizar a ferramenta em uma máquina), o tipo de licença (seja aberta para uso comercial ou não), se funciona *Offline* (ou requer conexão *internet*), se possui assistência por inteligência artificial (ou todas as marcações são manuais) e se ele possui sincronização entre máquinas diferentes (ou se os projetos podem ser acessados apenas nas máquinas do usuário principal).

Quadro 3 - Relação do mercado com as necessidades

Ferramenta	Instalação	Licença	Offline	Assistência	Sincronização
LabelImg	Simples	Aberta	Sim	Não	Não
Label Studio	Simples	Fechada	Sim	Não	Não
CVAT	Complexa	Aberta	Sim	Não	Não
SuperAnnotate	Simples	Fechada	Não	Sim	Sim
Makesense.ai	Simples	Fechada	Não	Sim	Sim
VGG VIA	Complexa	Aberta	Sim	Não	Não

Fonte: Autoria própria (2025)

Os resultados coletados foram feitos em uma máquina notebook com processados Intel I5 10400K, 12GB de memória RAM, 250GB de memória de armazenamento e rodando o sistema operacional Windows 11. Os testes foram feitos sem conexão à *internet* disponível.

4 RESULTADOS E DISCUSSÃO

Os primeiros resultados do The Marker começam com a implementação das ferramentas de marcação para detecção e segmentação de objetos. Ambas as opções ficam disponíveis na tela no formato de três botões: “Seleção”, “Quadrado” e “Polígono”, sendo respectivas às opções de: “não fazer marcações”, “fazer marcações de detecção de objetos” e “fazer marcações de segmentação de objetos”. As marcações são feitas com pontos de cor azul e verde e linhas de cor vermelhas para melhor contraste contra as outras opções da ferramenta. Como evidenciado na Figura 6 o processo de detecção de objetos, no qual o sistema identifica regiões a partir das marcações realizadas pelo usuário.

Figura 6 – Marcação para detecção de objetos

Fonte: Autoria própria (2025)

A Figura 7 apresenta o processo de segmentação de objetos, que tem como objetivo destacar com maior precisão as áreas pertencentes a cada elemento da imagem. Nessa etapa é essencial para

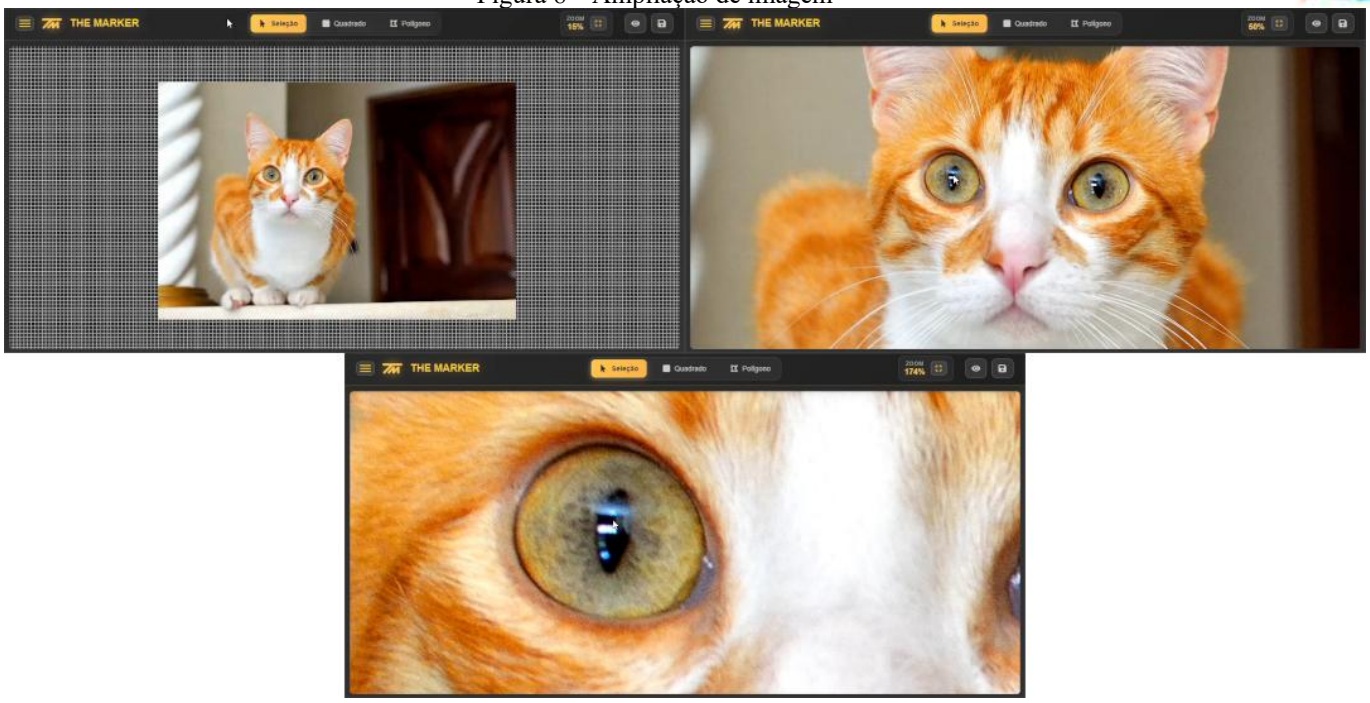
gerar máscaras mais detalhadas e consistentes, fundamentais na criação de conjuntos de dados, também o sistema exibe as delimitações com cores contrastantes para facilitar a distinção entre diferentes regiões segmentadas, dessa forma proporcionando uma experiência visual mais intuitiva e organizada.

Figura 7 – Marcação para segmentação de objetos

Fonte: Autoria própria (2025)

Além disso, as imagens são mostradas sobre uma tela infinita conforme Figura 8 que propõe liberdade vertical e horizontal para o usuário ser capaz de ajustar a posição da imagem na tela como for de sua preferência sem perder as informações de coordenadas da marcação. Em conjunto, foi implementada uma funcionalidade de ampliar e reduzir a imagem (popularmente conhecido como "*zoom-in*" e "*zoom-out*"), isso permite ao usuário focar em detalhes menores da imagem para marcar com melhor refino.

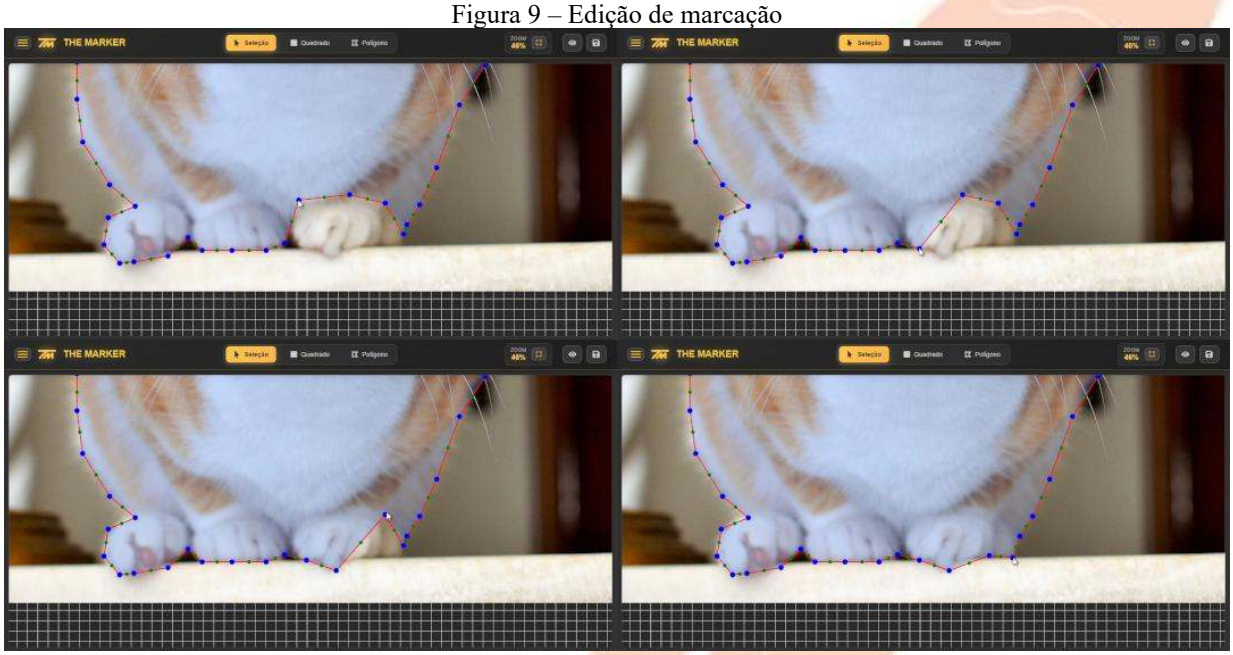
Figura 8 – Ampliação de imagem



Fonte: Autoria própria (2025)

Em adição à funcionalidade de ampliação e redução da imagem, foi implementada a opção de mover pontos de marcação que já foram feitos, permitindo que mesmo que erros tenham sido feitos durante o processo de marcação, os pontos possam ser movimentados para as posições corretas, conforme Figura 9 onde a pata do gato foi propositalmente esquecida e depois ajustada posteriormente.

Figura 9 – Edição de marcação

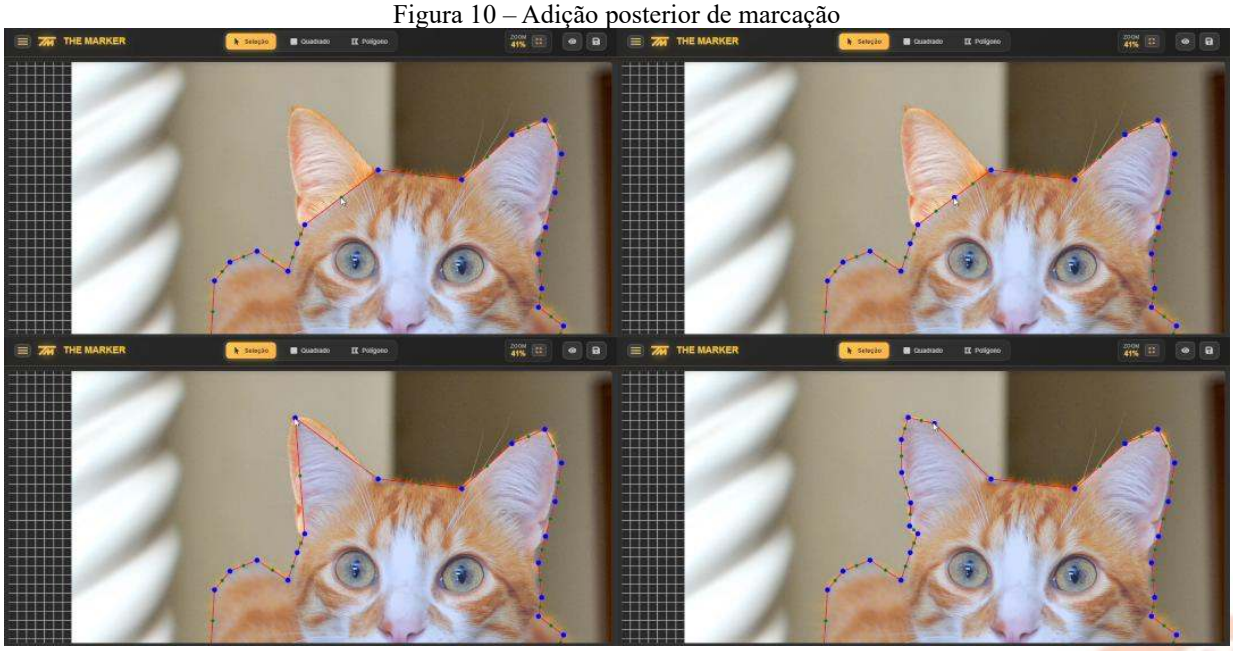


Fonte: Autoria própria (2025)

Além de editar a posição de marcações já feitas, existe a opção para o usuário criar novos pontos intermediários (representados pelos pontos verdes) conforme Figura 10. Assim é possível

aumentar a resolução do polígono e podendo movimentar suas posições para preencher áreas que havia sido deixada de fora e que apenas mover os pontos existentes não preencheria.

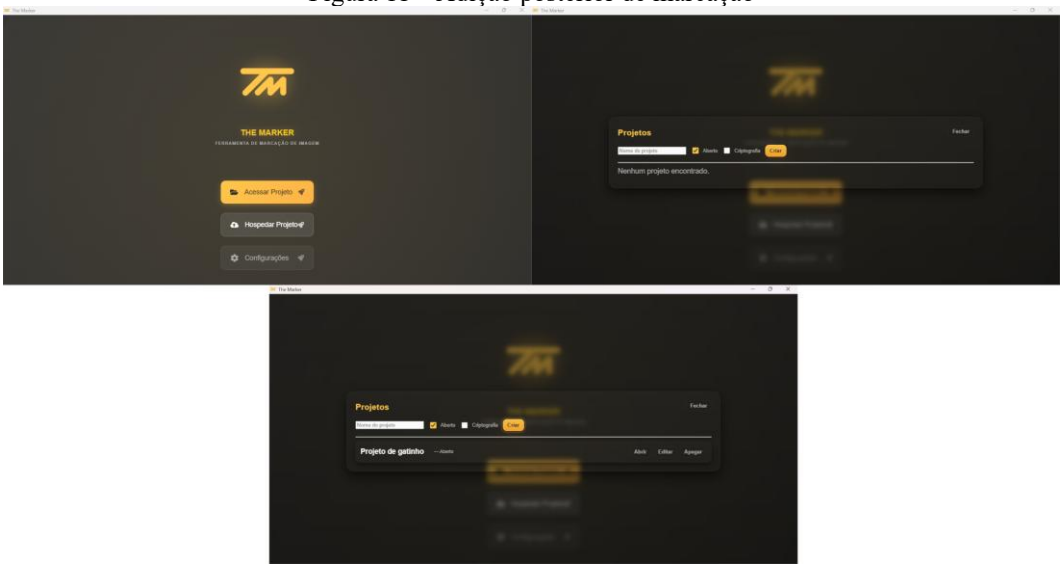
Figura 10 – Adição posterior de marcação



Fonte: Autoria própria (2025)

O usuário também possui a possibilidade de organizar conjuntos de imagens de trabalho, denominados “Projetos”, conforme apresentado na Figura 11, essa funcionalidade foi desenvolvida para facilitar o gerenciamento de diferentes bases de dados e contextos de marcações dentro do The Marker. Na tela principal, encontramos as opções “Hospedar Projeto” que permite ao usuário criar um novo grupo de trabalho e disponibilizá-los para acessar a rede, hospedamos diretamente em sua própria máquina, “Acessar Projeto” que possibilita a conexão a um grupo de trabalho existente hospedado em outros dispositivos, promovendo a colaboração entre diferentes usuários e distribuindo o fluxo de marcações de forma integrada e “configurações”.

Figura 11 – Adição posterior de marcação



Fonte: Autoria própria (2025)

Para o projeto foi criado um banco de dados e com opção para criptografar o armazenamento dos dados com a criptografia AES-GCM utilizando chaves de 256 bits. Dessa maneira, cada projeto possui uma chave diferente ao ser armazenado e permitindo que mesmo que o equipamento seja compartilhado entre diversas pessoas, apenas aquelas que tenham as credenciais corretas possam acessar.

No quadro 4 podemos ver a comparação de uma execução de teste que compara um processo repetido em cem vezes de: criar um banco de dados, criar uma tabela e preencher esta tabela com cinco itens. Pode-se ver que mesmo o tempo de execução em modo sem criptografia ser mais rápido, o modo com criptografia apresenta um aumento de 4 milissegundos.

Quadro 4 – Comparação entre bancos de dados com e sem criptografia

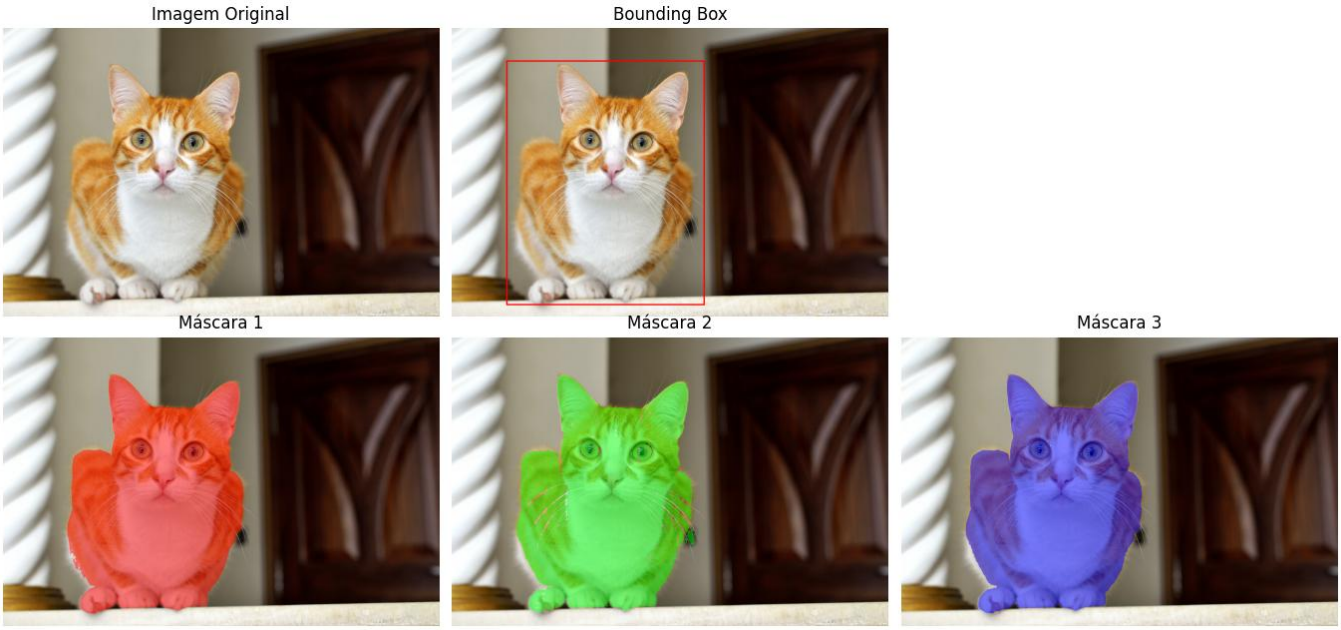
Modo	**Duração do teste (ms)**
Sem criptografia	6
Com criptografia	10

Fonte: Autoria própria (2025)

Esse aumento de tempo pode ser considerado desprezível para o cenário do The Marker porque o tempo de análise entre inserções no banco de dados não é esperado que dure menos que um segundo e considerando o nível de segurança que a criptografia AES oferece com uma chave de 256 bits.

Com a implementação do SAM 2.1, o modelo de inteligência artificial é capaz de analisar a região de um *bounding box* marcado manualmente e sugerir três máscaras com a maior probabilidade de serem o objeto desejado. Como mostrado na Figura 12, onde o *bounding box* demarca a região que está o gato e as máscaras são propostas com base nele.

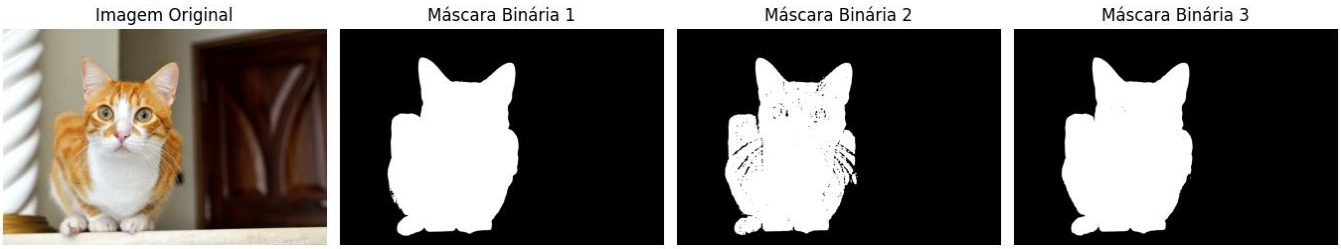
Figura 12 – Proposição de máscaras



Fonte: Autoria própria (2025)

As máscaras são informações de imagem com a mesma dimensão da imagem original, porém elas marcam em cor preto (valor 0) onde a máscara não existe e cor branca (valor 255) onde a máscara existe. Ela sozinha não permite o projeto identificar a segmentação da imagem para o usuário. Essa segregação da máscara e do fundo da imagem pode ser visto na Figura 13.

Figura 13 – Máscaras binárias



Fonte: Autoria própria (2025)

Para isso, é utilizado a biblioteca Python *scikit-image* para aplicar o tratamento de imagem na máscara proposta a fim de converter os contornos da máscara em pontos de marcação. Como mostrado na Figura 14, esse método é capaz de transformar a marcação de dois pontos (o *bounding box*) em contornos de segmentação com centenas de pontos que acompanham figuras orgânicas com precisão.

Figura 14 – Conversão de máscaras para segmentação

Fonte: Autoria própria (2025)

Com o processo definido, foi montado um estudo que demonstra o tempo de processamento de cada etapa e o consumo de memória de acesso aleatório (RAM). Como visto no Quadro 5, o maior tempo de processamento demorou 8,58 segundos em processamento de CPU e consumiu menos de 1 GB de memória rodando localmente. Todo o processo demorou 11,39 segundos dado a natureza sequencial do processo.

Quadro 5 – Consumo do modelo em imagem individual

Modo	Duração (s)	RAM utilizada (MB)
Carregamento do modelo	2,10	9,92
Carregamento da imagem	0,23	61,86
Pré-processamento da imagem	0,29	54,60
Inferência do modelo	8,58	849,75
Pós-processamento das máscaras	0,19	48,27

Fonte: Autoria própria (2025)

Usando o conjunto de dados LVIS para fazer uma análise massiva com uma amostra de mil imagens que majoritariamente seguem a qualidade *VGA* (*Video Graphics Array*, resolução 640 *pixels* na horizontal por 480 *pixels* na vertical), os resultados demonstram que imagens de o modelo *Segment Anything Model* se divide em cenários e alto desempenho e cenários de baixo desempenho. Como no relacionado no Quadro 6. As menores precisões foram identificadas em imagens de qualidade baixa, marcação de objetos pequenos em relação à resolução da imagem (32 *pixels* ou menos) como na Figura 15, onde se tentou marcar a tomada conectada na parede; ou ambiguidade no que pode ser o alvo da marcação como demonstrado na Figura 16, que é possível que seja apenas o homem com seu jornal ou sem seu jornal.

A precisão é calculada através do método IoU e o nível de variação da precisão do modelo neste teste se deu pela natureza do LVIS possuir imagens em uma gama de resoluções e marcações de objetos em tamanhos pequenos e grandes, até mesmo redundância para marcações (como marcar um objeto grande, depois marcar partes pequenas dele separadamente).

Quadro 6 – Resultados massivos em precisão

Categoria	Precisão máxima	Precisão mínima	Precisão média
Global	98%	22%	52%
10% Maiores precisões	98%	87%	93%
10% Menores precisões	47%	22%	37%

Fonte: Autoria própria (2025)

Figura 15 – Marcação de objetos pequenos

Fonte: Autoria própria (2025)

Figura 16 – Ambiguidade de marcação

Fonte: Autoria própria (2025)

Também existe a diferença que mudanças na resolução causam no tempo de processamento, onde a quantidade de processamento necessário em segundos aumenta em relação à resolução, onde imagens de menores resoluções tendem a consumir tempo de processamento menor em relação a imagens de maior qualidade como mostrado no Quadro 7.

Quadro 7 – Resultados massivos em tempo de inferência

Categoria	Duração média (s)
Global	8
10% Maiores resoluções	13
10% Menores resoluções	6

Fonte: Autoria própria (2025)

Em relação à quantidade de pontos para a quantidade de toques do usuário, era esperado uma redução da média de 35 toques por segmentação para apenas dois toques por segmentação. Como demonstrado no Quadro 8, a aplicação da inteligência artificial permitiu a criação de uma quantidade maior de pontos que o estimado, gerando a necessidade de um pós-tratamento para simplificar a visualização e experiência do usuário. Por conta da imprecisão do modelo, ainda é necessária uma curadoria manual do usuário que adiciona 5 toques em média.

Quadro 8 – Relação de pontos por toques

Categoria de marcação	Média de toques por objeto	Média de pontos por objeto
Detecção	2	2
Anotação manual	35	35
Segmentação estimada	2	35
Inferência	2	127
Inferência pós-tratamento	2	49
Inferência com ajuste manual	7	49

Fonte: Autoria própria (2025)

5 CONSIDERAÇÕES FINAIS

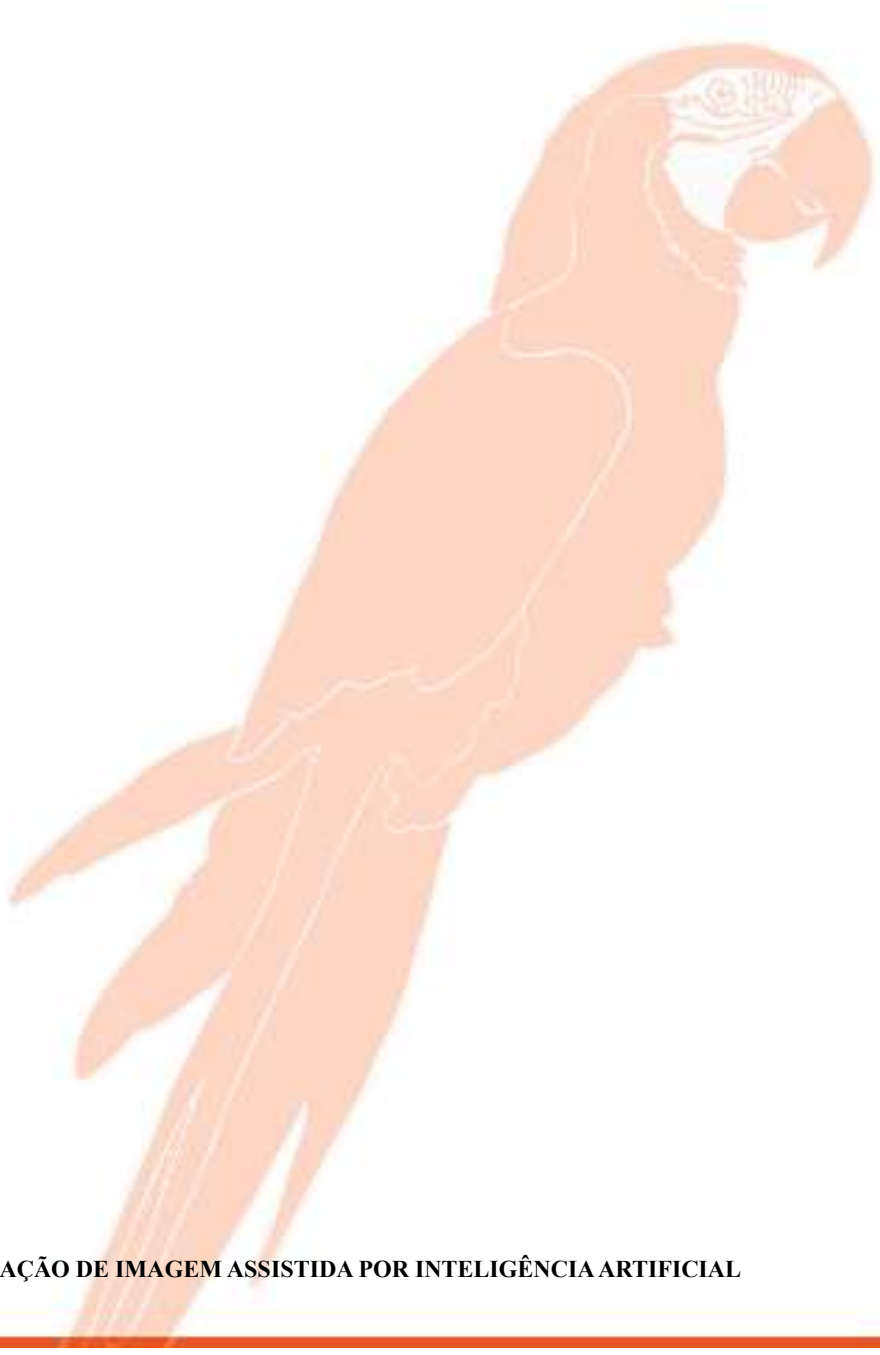
O projeto The Marker evidenciou a possibilidade de uma ferramenta capaz de integrar uma instalação simplificada em licença de código aberto que funcione *offline*, possua assistência por inteligência artificial e sincronização dos projetos entre múltiplos usuários simultaneamente em um cenário que as maiores ferramentas com funcionalidades parecidas são de código fechado e pagas. A natureza da utilização de ferramentas de desenvolvimento *Web* permite um visual moderno com suporte a longo prazo de suas tecnologias.

Mesmo que o projeto The Marker não obtenha resultados em categorias individuais como a instalação simplificada e leve do *LabelImg*, a comunidade em código aberto do Label Studio, a assistência por inteligência artificial com modelos com treinamento dedicado do Supervisely ou a sincronização do trabalho colaborativo do Diffgram; ele foi capaz de abranger e unir essas categorias em um único produto junto a um sistema de criptografia AES-GCM com 256 bits para armazenamento seguro e privado das informações.

Dentre as limitações do projeto ficam destacadas a ausência de testes com equipes que tenham capacidade de operar um *datasets* com dez mil ou mais imagens para ter métricas extensas de resultados em casos de uso reais. Também se destaca a limitação quanto ao uso de inteligência artificial que opera apenas com o modelo *Segment Anything Model 2* em sua capacidade mais leve (que requer menos poder de processamento).

Em possíveis desenvolvimentos futuros seria interessante a implementação de mais modelos de inteligência artificial e um teste com uma amostra maior, porém, supõem-se a possibilidade de adicionar a opção de *fine-tuning* de modelos dentro da ferramenta para melhorar a acurácia das marcações e reconhecimento automático de objetos. Adicionalmente: a elaboração de um modelo sem

front-end que exista apenas como gerenciador de acessos externos aos projetos de modo que eles possam ser hospedados em servidores com *IP* disponível na *internet* ao invés de serem limitados a rede local *LAN*.



REFERÊNCIAS

ABADI, M. et al. TensorFlow: a system for large-scale machine learning. 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI), 2016.

ANDRILUKA, M.; UIJLINGS, J. R.; FERRARI, V. Fluid annotation: a human–machine collaboration interface for full image annotation. arXiv preprint, arXiv:1806.07527, 2018. Disponível em: https://arxiv.org/abs/1806.07527. Acesso em: 28 set. 2025.

ARMSTRONG, T. J. et al. A conceptual model for work-related neck and upper-limb musculoskeletal disorders. Scandinavian Journal of Work, Environment & Health, v. 19, n. 2, p. 73–84, 1993.

BOJARSKI, M. et al. End to end learning for self-driving cars. arXiv preprint, arXiv:1604.07316, 2016. Disponível em: https://arxiv.org/abs/1604.07316. Acesso em: 28 set. 2025.

BRASIL. MINISTÉRIO DA SAÚDE. Saúde do trabalhador: notificações de LER/DORT no Brasil. Brasília: Ministério da Saúde, 2023. Disponível em: https://www.gov.br/saude/pt-br/assuntos/saude-do-trabalhador. Acesso em: 28 set. 2025.

BRASIL. MINISTÉRIO DO TRABALHO E EMPREGO. Norma Regulamentadora nº 17: Ergonomia. Brasília: MTE, 2023.

CHEN, X. et al. Brain tumor classification based on neural architecture search. Scientific Reports, v. 12, art. 19206, 2022. DOI: 10.1038/s41598-022-22172-6.

CHEN, X. et al. UCVL: a benchmark for crime surveillance video analysis with large models. Neurocomputing, v. 600, p. 128–142, 2025.

COLES-BRENNAN, C.; SULLEY, A.; YOUNG, G. Management of digital eye strain. Clinical and Experimental Optometry, v. 102, n. 1, p. 18–29, 2019.

DEFENSE SCOOP. NGA awards $700M data labeling contract to advance computer vision models. DefenseScoop, 3 set. 2024. Disponível em: https://defensescoop.com/2024/09/03/nga-700m-data-labeling-advance-computer-vision-models/. Acesso em: 3 nov. 2025.

DUTTA, A.; ZISSERMAN, A. The VIA annotation software for images, audio and video. arXiv preprint, arXiv:1904.10699, 2019. Disponível em: https://arxiv.org/abs/1904.10699. Acesso em: 28 set. 2025.

EVERINGHAM, M. et al. The Pascal Visual Object Classes (VOC) challenge. International Journal of Computer Vision, v. 88, n. 2, p. 303–338, 2010.

GOHILL, H. et al. A hybrid technique for plant disease identification and localisation in real-time. Computers and Electronics in Agriculture, v. 219, 108838, 2024.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. Deep learning. Cambridge: MIT Press, 2016.

GOWRISANKARAN, S.; SHEEDY, J. E. Computer vision syndrome: a review. Work, v. 52, n. 2, p. 303–314, 2015.

HAGBERG, M.; SILVERSTEIN, B.; WELLS, R. Work related musculoskeletal disorders: a reference book for prevention. London: Taylor & Francis, 1995.

KHALIL, K.; KIMIAFAR, K.; ZADEH, M. R.; et al.Artificial intelligence literacy among healthcare professionals and students: a systematic review. Health Informatics Journal, v. 29, n. 4, p. 1–15, 2023.

KIRILLOV, A. et al. Segment anything. arXiv preprint, arXiv:2304.02643, 2023. Disponível em: https://arxiv.org/abs/2304.02643. Acesso em: 28 set. 2025.

KOVASHKA, A. et al. Human-in-the-loop annotation. Foundations and Trends in Computer Graphics and Vision, 2016.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, v. 25, 2012.

KUNDU, R.; DAS, R.; GHOSH, S.; et al. Pneumonia detection in chest X-ray images using an ensemble of convolutional neural networks. PLOS ONE, v. 16, n. 9, e0256630, 2021.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. Nature, v. 521, p. 436–444, 2015.

LIN, T. Y. et al. Microsoft COCO: common objects in context. In: European Conference on Computer Vision (ECCV), 2014. Disponível em: https://arxiv.org/abs/1405.0312. Acesso em: 28 set. 2025.

MERRILL, R. M.; ALLEMAN, J. R. The relevance of ergonomic interventions for the prevention of musculoskeletal disorders. Journal of Occupational and Environmental Medicine, v. 54, n. 4, p. 427–433, 2012.

META. React – A JavaScript library for building user interfaces. 2013. Disponível em: https://react.dev/. Acesso em: 28 set. 2025.

MITCHELL, T. Machine learning. New York: McGraw-Hill, 1997.

MORAIS, D. M. G. et al. O conceito de inteligência artificial usado no mercado de softwares, da educação tecnológica e na literatura científica. Educação Profissional e Tecnológica em Revista, v. 4, n. 2, p. 98–109, 2020.

OMS – ORGANIZAÇÃO MUNDIAL DA SAÚDE. Ergonomics in the workplace. Geneva: WHO, 2003.

PAPADOPOULOS, D. P. et al. Extreme clicking for efficient object annotation. In: International Conference on Computer Vision (ICCV), 2017.

PASZKE, A. et al. PyTorch: an imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems, v. 32, 2019.

RAJPURKAR, P. et al. CheXNet: pneumonia detection. arXiv preprint, arXiv:1711.05225, 2017.

REDMON, J. et al. YOLO: real-time object detection. CVPR, 2016.

REN, S.; HE, K.; GIRSHICK, R.; SUN, J. Faster R-CNN: region proposal networks. NeurIPS, 2015.

ROSENFIELD, M. Computer vision syndrome (a.k.a. digital eye strain). Optometry in Practice, v. 17, n. 1, p. 1–10, 2016.

RUSSELL, B. et al. LabelMe: a database and web-based tool for image annotation. International Journal of Computer Vision, v. 77, p. 157–173, 2008.

RUSSELL, S.; NORVIG, P. Artificial intelligence: a modern approach. 3. ed. Upper Saddle River: Pearson, 2010.

SAGER, C.; JANIESCH, C.; ZSCHECH, P. A survey of image labelling for computer vision applications. arXiv preprint, arXiv:2104.08885, 2021. Disponível em: https://arxiv.org/abs/2104.08885. Acesso em: 28 set. 2025.

SANAR. Lesões por esforço repetitivo (LER) e distúrbios osteomusculares relacionados ao trabalho (DORT): conceitos e prevenção. Disponível em: https://www.sanar.com.br/. Acesso em: 17 nov. 2025.

SCHUHMANN, C.; BEAUMONT, R.; VENCU, R.; GORDON, C.; WIGHTMAN, R.; CHERTI, M.; et al. LAION-5B: An open large-scale dataset for training next generation image-text models. Advances in Neural Information Processing Systems, v. 35, p. 25278–25294, 2022.

SHEPPARD, A. L.; WOLFFSOHN, J. S. Digital eye strain: prevalence, measurement and amelioration. BMJ Open Ophthalmology, v. 3, n. 1, e000146, 2018.

SHIN, H. et al. Visual product search using deep learning. 2022.

SZELISKI, R. Computer Vision: Algorithms and Applications. Springer, 2010.

TZELEPIS, D. et al. Efficient bounding box annotation. Pattern Recognition Letters, 2021

TAURI. Tauri documentation. 2022. Disponível em: https://tauri.app/. Acesso em: 28 set. 2025.
TIME. Meta scales up the AI data industry. Time, 19 set. 2024. Disponível em: https://time.com/7294699/meta-scale-ai-data-industry/. Acesso em: 3 nov. 2025.

WANG, L.; ZHAO, X.; ZHANG, Y.; HAN, X.; DEVEÇI, M. A review of convolutional neural networks in computer vision. Artificial Intelligence Review, v. 57, n. 4, p. 1–27, 2024.

ZHOU, Z. H. A brief introduction to weakly supervised learning. Springer, 2018.