Malware DataLab: uma Plataforma de Ensino-Aprendizagem e Experimentação para Geração de Dados Sintéticos na Detecção de Malware Android

Diego Kreutz[1], Rodrigo Brandão Mansilha[1], Luiz Felipe Laviola[1]
Angelo Gaspar Diniz Nogueira[1], Kayuã Oleques Paim[1], Hendrio Bragança[1]

[1]AI Horizon Labs, PPGES, Universidade Federal do Pampa (UNIPAMPA)
Alegrete, RS, Brasil

{diegokreutz,rodrigomansilha}@unipampa.edu.br

{luizlaviola,angelodiniz}.aluno@unipampa.edu.br

kayua.paim@inf.ufrgs.br

hendrio.luis@icomp.ufam.edu.br

Abstract. *Android malware detection based on machine learning is limited by the scarcity, bias, and short validity of representative datasets, while the underlying techniques (such as generative adversarial networks) remain hard to teach and to reproduce. This paper presents Malware DataLab, an integrated web platform that combines a teaching-and-learning laboratory with an experimentation laboratory for synthetic tabular data generation. The platform couples MalSynGen, a neural-network-based synthetic data generator, with Cloud AutoDroid, a scalable distributed backend that orchestrates containerized experiments on demand. We describe the tool's functionalities, its architecture and implementation, and a user evaluation with 76 participants (including residents of the Hackers do Bem program), which reported an 84.6% task completion rate and a Net Promoter Score of approximately 81. The platform, its documentation, and the associated artifacts are openly available.*

Resumo. *A detecção de malware Android baseada em aprendizado de máquina é limitada pela escassez, pelo viés e pela curta validade de conjuntos de dados representativos, enquanto as técnicas envolvidas (como redes adversárias generativas) permanecem difíceis de ensinar e de reproduzir. Este artigo apresenta o Malware DataLab, uma plataforma web integrada que combina um laboratório de ensino-aprendizagem com um laboratório de experimentação para geração de dados tabulares sintéticos. A plataforma integra o MalSynGen, um gerador de dados sintéticos baseado em redes neurais, ao Cloud AutoDroid, um backend distribuído e escalável que orquestra experimentos conteinerizados sob demanda. Descrevemos as funcionalidades da ferramenta, sua arquitetura e implementação, e uma avaliação com 76 usuários (incluindo residentes do programa Hackers do Bem), que apontou 84,6% de conclusão de tarefas e um Net Promoter Score de aproximadamente 81. A plataforma, sua documentação e os artefatos associados estão abertamente disponíveis.*

1. Introdução

O ecossistema Android conecta bilhões de dispositivos em todo o mundo e concentra uma parcela expressiva das ameaças móveis atuais. O crescimento contínuo de malwares, muitos deles produzidos ou modificados com o apoio de técnicas de Inteligência Artificial (IA) generativa, impõe prejuízos financeiros que podem atingir dezenas de milhares de euros por incidente e pressiona a evolução dos mecanismos de detecção [Meijin et al. 2022]. Essa sofisticação crescente, em que a própria IA é usada para gerar variantes capazes de escapar dos classificadores, encurta a validade das assinaturas e acelera a corrida entre atacantes e defensores. Nesse cenário, técnicas de aprendizado de máquina tornaram-se centrais para a detecção de malware, mas seu desempenho depende diretamente da disponibilidade de dados representativos, atualizados e não enviesados.

Justamente aí reside um gargalo persistente. Conjuntos de dados de malware Android são escassos, frequentemente desatualizados e sujeitos a vieses que comprometem a validade dos resultados obtidos [Miranda et al. 2022, Soares et al. 2021b, Soares et al. 2021a]. Levantamentos empíricos sobre as fontes de dados do domínio evidenciaram problemas recorrentes de disponibilidade, atualização e reprodutibilidade [Soares et al. 2021b, Soares et al. 2021a], e o viés desses conjuntos pode inflar artificialmente as métricas dos detectores, mascarando seu desempenho real [Miranda et al. 2022]. Essa fragilidade não é um detalhe operacional: uma fração substancial dos projetos de IA fracassa por problemas relacionados à qualidade e à quantidade de dados [AI & Data Today 2023]. A geração de dados sintéticos surge como uma alternativa promissora para mitigar essa limitação, permitindo ampliar e balancear conjuntos de treinamento sem os riscos associados à manipulação de malware real e, em estudos de caso do próprio grupo, elevando o poder de classificação dos detectores em até 15% [Nogueira et al. 2024b, Nogueira et al. 2024a].

Contudo, transformar essa alternativa em prática enfrenta duas barreiras adicionais. Primeiro, as técnicas envolvidas, como redes adversárias generativas (*GANs*) e validação cruzada, possuem curva de aprendizado elevada, o que dificulta sua adoção por estudantes e profissionais em formação [Paim et al. 2024]. Segundo, a execução de experimentos de IA generativa demanda alto poder computacional e infraestrutura distribuída, raramente acessível de forma simplificada em contextos educacionais [Laviola et al. 2024, Laviola et al. 2025].

Este artigo apresenta o **Malware DataLab**, uma plataforma web integrada que unifica, em um único ambiente, o ensino-aprendizagem e a experimentação prática com geração de dados sintéticos para detecção de malware Android. A plataforma articula dois laboratórios complementares: um Laboratório de Ensino-Aprendizagem, com módulos teóricos e práticos, e um Laboratório de Experimentação, que dá acesso à ferramenta MalSynGen sobre uma infraestrutura distribuída e escalável (Cloud AutoDroid). O objetivo é democratizar o acesso a tecnologias avançadas de IA aplicadas à cibersegurança, reduzindo a curva de aprendizado e a barreira computacional.

As principais contribuições deste trabalho são: (i) a descrição de uma plataforma educacional integrada que conecta conteúdo pedagógico e experimentação real de geração de dados sintéticos; (ii) uma arquitetura de backend distribuída e escalável para execução sob demanda de tarefas intensivas de IA generativa; e (iii) uma avaliação com 76 usuários de perfis diversos, incluindo residentes do programa *Hackers do Bem*, evidenciando alta

taxa de conclusão de tarefas e elevado índice de satisfação.

2. Trajetória

O grupo de pesquisa que sustenta o Malware DataLab acumula uma trajetória de ferramentas voltadas à detecção de malware Android e à qualidade de dados. Ferramentas de construção e avaliação de conjuntos de dados, como o ADBuilder [Vilanova et al. 2022], e estudos sobre a disponibilidade, atualização e reprodutibilidade das fontes de dados [Soares et al. 2021b, Soares et al. 2021a] evidenciaram, de forma recorrente, a escassez e o viés dos datasets disponíveis. Em paralelo, ferramentas de *AutoML* para o domínio de malware Android, como o DroidAutoML [Assolin et al. 2022] e avaliações comparativas de *AutoML* [Siqueira et al. 2022], exploraram a automação do treinamento de classificadores.

A vertente de geração de dados sintéticos teve origem no DroidAugmentor [Casola et al. 2023], uma ferramenta de treinamento e avaliação de *cGANs* (*conditional GANs*) para dados tabulares. Sua evolução resultou no MalSynGen [Nogueira et al. 2024b], que emprega redes neurais artificiais e validação cruzada e cuja reprodutibilidade foi reconhecida com o selo de Artefato Destaque (Disponível, Funcional, Reprodutível e Sustentável) no Salão de Ferramentas do SBSeg 2024. Em estudos de caso controlados, os detectores treinados com os dados sintéticos gerados alcançaram ganhos de até 15% no poder de classificação, avaliados por métricas como acurácia, revocação e F1-*score*, resultado que rendeu menção honrosa na trilha principal do SBSeg 2024 [Nogueira et al. 2024a]. Trabalhos subsequentes trataram da otimização de hiperparâmetros da rede geradora, premiado como melhor artigo no WRSeg 2024 [Nogueira et al. 2024c], e da modularização do gerador no MalDataGen [Nogueira et al. 2025], que abstrai o pipeline de geração para acomodar múltiplas famílias generativas. Do ponto de vista de infraestrutura, o Cloud AutoDroid [Laviola et al. 2024, Laviola et al. 2025] introduziu um backend distribuído para executar serviços de IA generativa na nuvem, com decisões de projeto amparadas por avaliações do custo de desempenho da conteinerização aninhada [Fava et al. 2024, Schepke et al. 2024].

Diferentemente dessas contribuições isoladas, que atacam etapas específicas (geração, otimização ou execução), o Malware DataLab integra essas peças em uma plataforma única de ensino e experimentação. Ele adiciona uma camada pedagógica estruturada e uma interface web acessível sobre a infraestrutura distribuída, aproximando a fronteira técnica da formação prática de novos especialistas. Ao longo do projeto, essa trajetória consolidou-se em mais de dez trabalhos científicos publicados e três premiações, evidenciando a maturação técnica que sustenta a plataforma aqui descrita.

3. Visão Geral e Funcionalidades da Ferramenta

O Malware DataLab é uma plataforma web integrada de ensino-aprendizagem e experimentação, acessível por um ponto único de entrada. Seu diferencial está em combinar, no mesmo ambiente, a formação teórica e a experimentação real de geração de dados sintéticos para o fortalecimento de sistemas de detecção de malware Android. A plataforma foi projetada para atender a diferentes perfis e níveis de maturidade técnica, do ensino médio técnico à pós-graduação. A Figura 1 apresenta uma visão geral da ar-

quitetura, relacionando o usuário, a aplicação web, a API e os serviços de geração e armazenamento de dados.

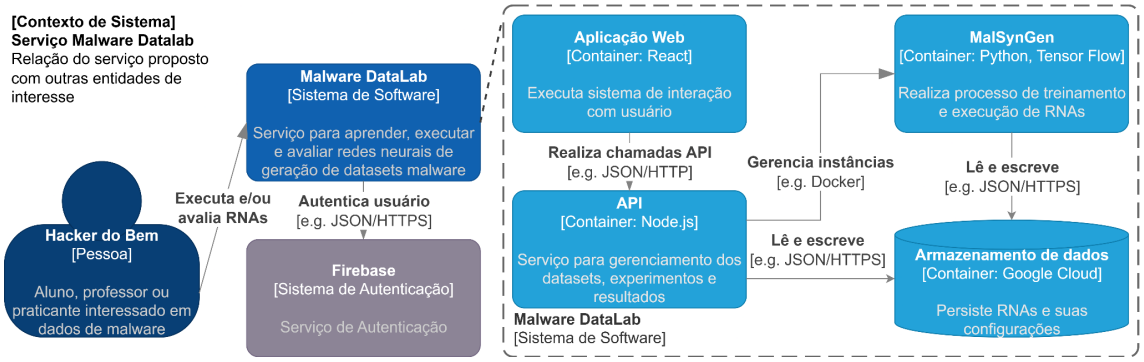

Figura 1. Arquitetura geral do Malware DataLab: o usuário interage com a aplicação web (frontend), que se comunica com a API e, por meio dela, com o gerador MalSynGen e o armazenamento de dados, com autenticação via Firebase.

3.1. Laboratório de Ensino-Aprendizagem

O Laboratório de Ensino-Aprendizagem organiza o conteúdo em módulos fundamentais, que abrangem desde o nivelamento inicial até conceitos avançados de redes neurais e *cGANs*. Os materiais são apresentados em formatos diversos (textos, slides e vídeos) e acompanhados de um sistema de avaliação em três níveis, cada um com função pedagógica distinta: a avaliação diagnóstica sonda o conhecimento prévio e posiciona o usuário na trilha adequada; a formativa acompanha a evolução ao longo dos módulos e fornece retorno contínuo; e a somativa consolida a aprendizagem ao final de cada percurso. Essa estrutura permite acompanhar a progressão do usuário e adaptar dinamicamente o percurso pedagógico ao seu nível de maturidade técnica.

Os módulos foram concebidos para atender a públicos distintos, articulando teoria e prática em três faixas de complexidade. No nível básico, voltado ao ensino médio técnico, estudantes geram amostras sintéticas simples para compreender como alterações em permissões ou comunicações externas afetam a classificação em modelos elementares. No nível intermediário, voltado à graduação, os usuários atuam como analistas de segurança, explorando *logs* de execução e propondo estratégias de mitigação a partir de padrões maliciosos extraídos dos dados. No nível avançado, voltado à pós-graduação e à pesquisa, empregam-se GANs para produzir amostras capazes de desafiar classificadores robustos, em abordagens de treinamento adversarial. A conteinerização das técnicas de fundo permite que essa progressão ocorra sem que o iniciante precise lidar com a complexidade de configuração de ambientes de aprendizado profundo. Os conteúdos foram revisados iterativamente a partir do retorno dos usuários, com regravação dos vídeos de tutorial e reorganização do fluxo didático para tornar mais clara a interpretação das métricas geradas na plataforma.

3.2. Laboratório de Experimentação

O Laboratório de Experimentação permite que o usuário aplique os conhecimentos adquiridos utilizando a ferramenta MalSynGen para gerar conjuntos de dados sintéticos de

alta qualidade. O fluxo de trabalho segue quatro etapas: o usuário seleciona um *dataset* de referência, parametriza a execução, dispara a geração e, ao final, recebe os *datasets* sintéticos acompanhados de um relatório de métricas. As execuções são processadas de forma assíncrona pela infraestrutura distribuída, o que libera o usuário da espera síncrona e permite enfileirar múltiplos experimentos simultaneamente.

O relatório de avaliação reporta a fidelidade e a utilidade dos dados gerados por meio de métricas como acurácia, revocação e F1-*score*, obtidas ao treinar classificadores sobre os dados sintéticos e validá-los contra dados reais. Isso viabiliza análises comparativas, por exemplo, contrastar o desempenho de diferentes algoritmos de detecção (redes neurais convolucionais, máquinas de vetores de suporte e árvores de decisão) sobre um mesmo conjunto, ou medir o quanto o aumento sintético eleva o poder de classificação, que em estudos de caso da própria ferramenta chegou a 15% [Nogueira et al. 2024a]. Esse laboratório concretiza a proposta de valor da plataforma: possibilitar experimentação em larga escala com IA generativa sem exigir que o usuário gerencie a complexidade computacional subjacente nem manipule amostras de malware real.

3.3. Interface Web e Integrações

A interface web foi desenvolvida com foco em acessibilidade, usabilidade e design responsivo, consolidando melhorias identificadas em ciclos anteriores de avaliação e evoluindo a partir de protótipos de GUI voltados à aprendizagem sobre malwares sintéticos [Sonco et al. 2024]. A Figura 2 ilustra a interface do Laboratório de Experimentação, com a barra de navegação que unifica o acesso aos dois laboratórios e a listagem de *datasets* disponíveis para geração de dados sintéticos.

A autenticação é realizada via Google Firebase, com suporte a contas Google, e integra-se à Comunidade Acadêmica Federada (CAFe) da Rede Nacional de Ensino e Pesquisa (RNP), permitindo que usuários das instituições federadas acessem a plataforma. A ferramenta também expõe APIs REST e GraphQL, com um SDK disponibilizado via NPM, e integra-se a ambientes externos como Google Docs e Google Colab para atividades práticas. Um canal de sugestões embutido na própria plataforma alimenta o processo contínuo de melhoria.

4. Arquitetura e Implementação

A arquitetura do Malware DataLab foi projetada para assegurar escalabilidade, modularidade e alta capacidade de processamento, requisitos essenciais para aplicações de IA generativa. Ela se organiza em duas camadas principais: um *frontend* web, que hospeda os módulos de ensino-aprendizagem e o laboratório de experimentação, e um *backend* distribuído, denominado Cloud AutoDroid, responsável por orquestrar as operações intensivas de geração de dados. A Figura 3 detalha a organização desse backend.

4.1. Orquestração e Execução de Contêineres

O Cloud AutoDroid adota uma abordagem de orquestração de contêineres distribuídos, gerenciando dinamicamente máquinas denominadas *workers*, responsáveis por executar os contêineres com a ferramenta MalSynGen e por distribuir as cargas de trabalho de forma eficiente. O backend foi desenvolvido em TypeScript, executado sobre o runtime Node.js e utilizando o framework Express.js para o tratamento de requisições HTTP. A

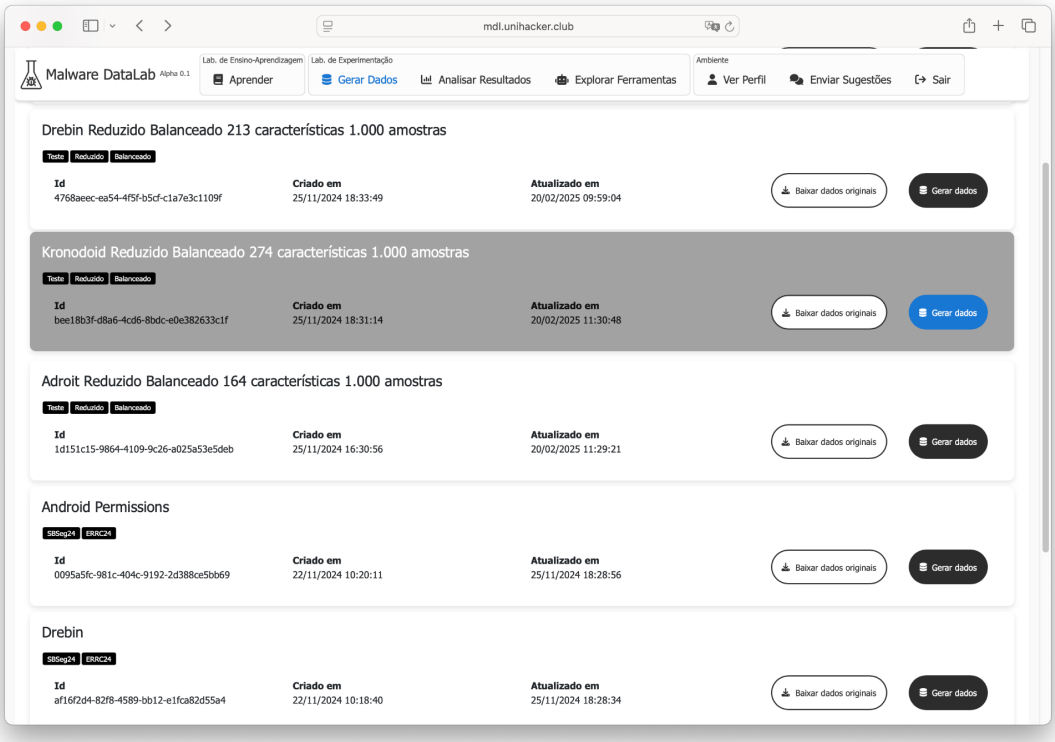

Figura 2. Interface web do Malware DataLab, destacando o Laboratório de Experimentação (*Gerar Dados*) e a navegação unificada entre os laboratórios de ensino-aprendizagem e de experimentação.

comunicação bidirecional em tempo real entre o backend e os *workers* é realizada com Socket.io, permitindo o controle das execuções distribuídas. A biblioteca Dockerode, por meio do arquivo `docker.sock`, viabiliza o controle direto dos contêineres Docker nos *workers*. A escolha por contêineres Docker apoia-se em avaliações prévias do impacto de desempenho da conteinerização, inclusive em cenários aninhados [Fava et al. 2024, Schepke et al. 2024].

A autenticação dos usuários é feita via Firebase. Para os *workers*, emprega-se uma autenticação inicial baseada em *token* administrativo (configurável para uso único ou múltiplo), posteriormente trocado por um par de *tokens* JWT (*access token* e *refresh token*), seguindo padrões de autenticação segura.

4.2. Armazenamento e Processamento Assíncrono

A camada de dados combina tecnologias distintas conforme a natureza da informação: o PostgreSQL é utilizado como SGBD relacional para dados estruturados e mestres (usuários, *tokens*, *workers*); o MongoDB persiste arquivos, *logs* e representações de processos; o Redis atua como banco chave-valor para o gerenciamento de tarefas assíncronas; e a biblioteca Bull.js, baseada no Redis, controla as filas de execução, garantindo confiabilidade e controle transacional das tarefas distribuídas. A aplicação suporta o *upload* de arquivos via URL assinada e expõe duas interfaces de programação: uma API RESTful, implementada com Express.js, e uma API GraphQL, por meio do Apollo Server.

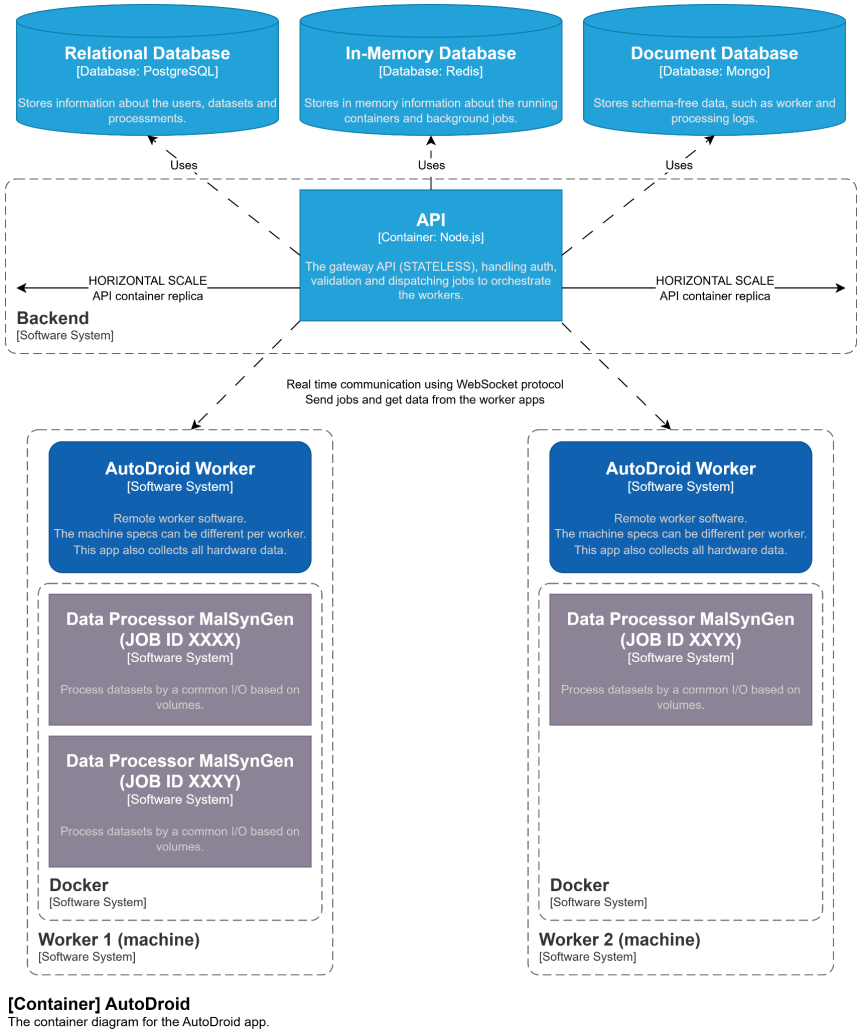

Figura 3. Arquitetura do backend Cloud AutoDroid: a API sem estado (*stateless*) orquestra os *workers* via WebSocket e utiliza PostgreSQL, Redis e MongoDB, enquanto cada *worker* executa contêineres Docker com o MalSynGen.

4.3. Infraestrutura

O Cloud AutoDroid foi implantado sobre uma infraestrutura escalável em nuvem do tipo *Infrastructure as a Service* (IaaS), com servidores virtuais de configuração típica de 4 vCPUs, 14 GB de memória RAM, 50 GB de disco local e 100 GB de armazenamento adicional em bloco. O ambiente opera com autoescalabilidade, atendendo a múltiplos usuários simultâneos de forma eficiente e sob demanda. Durante a fase de testes com usuários, foram utilizados quatro nós de computação no datacenter da RNP em Brasília e sete servidores adicionais nas dependências da UNIPAMPA, que também forneceu suporte com laboratórios e infraestrutura técnica. A Figura 4 ilustra essa distribuição geográfica dos servidores.

4.4. MalSynGen e o Núcleo de Geração de Dados

No núcleo da experimentação está o MalSynGen, gerador de dados tabulares sintéticos baseado em redes neurais artificiais e validação cruzada, sucessor do DroidAugmen-

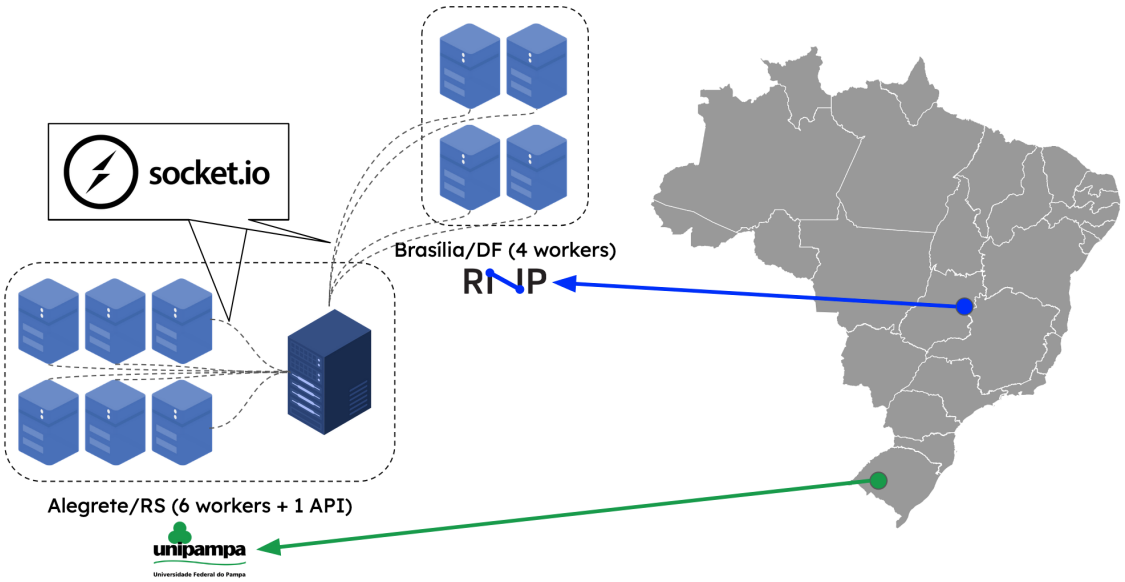

Figura 4. Infraestrutura distribuída que sustenta o Cloud AutoDroid, com nós de computação no datacenter da RNP em Brasília/DF e servidores na UNIPAMPA em Alegrete/RS, coordenados em tempo real via Socket.io.

tor [Casola et al. 2023, Nogueira et al. 2024b]. A abordagem parte de *cGANs*, em que uma rede geradora e uma rede discriminadora são treinadas de forma adversarial, condicionadas pela classe (benigno ou malicioso), de modo a produzir amostras que preservem a distribuição estatística das características reais (permissões, chamadas de API e demais atributos tabulares). A validação cruzada e a otimização de hiperparâmetros [Nogueira et al. 2024c] sustentam a qualidade e a estabilidade do treinamento, e a eficácia do resultado é aferida treinando detectores sobre os dados gerados: em estudos de caso controlados, esse aumento sintético elevou o poder de classificação em até 15% [Nogueira et al. 2024a].

O gerador é empacotado em contêineres Docker executados pelos *workers*, o que permite a execução paralela e isolada de múltiplos experimentos e garante a reprodutibilidade do ambiente. Essa organização modular favorece a extensibilidade: novos geradores ou variantes, como o MalDataGen [Nogueira et al. 2025], além de famílias generativas complementares às GANs, como autoencoders variacionais [Pinheiro Cinelli et al. 2021] e modelos de difusão [Ho et al. 2020], podem ser incorporados como novas imagens sem alterar a lógica de orquestração. O desenvolvimento adotou metodologias como Design Thinking, para o entendimento das necessidades dos usuários, e Scrum, para o desenvolvimento incremental em *sprints*, além de uma bateria de testes automatizados unitários, de integração e de ponta a ponta (E2E) que resguarda a estabilidade da plataforma a cada iteração.

5. Avaliação com Usuários

A avaliação do Malware DataLab combinou métricas técnicas de infraestrutura com *feedback* de usuários, ao longo de ciclos sucessivos de teste. Foram aplicados modelos e métodos consolidados: o TAM (*Technology Acceptance Model*), o UXUG-AP (*User Experience and Usability Guidelines - Agile Process*) e o ATAM (*Architecture Tradeoff*

Analysis Method), voltados, respectivamente, à aceitação tecnológica, à experiência de uso e à análise arquitetural. A coleta apoiou-se em questionários aplicados antes, durante e após o uso, complementados por ferramentas de telemetria e análise de comportamento, como Google Analytics, Microsoft Clarity, Amplitude e Sentry.

5.1. Metodologia

A validação seguiu ciclos incrementais, cada um retroalimentando o desenvolvimento. Um primeiro ciclo, em outubro e novembro de 2024, envolveu 115 usuários de diferentes regiões do Brasil e mapeou barreiras iniciais de usabilidade e de conteúdo. Um segundo ciclo, em janeiro de 2025, concentrou-se em usuários internos e em sete especialistas de UX/UI, refinando a interface a partir de uma avaliação especializada. O ciclo aqui reportado, realizado em abril de 2025, contou com 76 usuários, número acima da meta de 53, dos quais 26 residentes do programa *Hackers do Bem* e 50 estudantes e profissionais de perfis diversos.

A avaliação apoiou-se em três modelos consolidados, cada um cobrindo uma dimensão complementar: o TAM analisou a aceitação tecnológica (utilidade e facilidade de uso percebidas), o UXUG-AP orientou a análise de experiência e usabilidade em processo ágil, e o ATAM examinou os *trade-offs* da arquitetura frente a atributos de qualidade como escalabilidade e disponibilidade. Os instrumentos foram aplicados por meio de questionários antes, durante e após o uso, e triangulados com telemetria automatizada. Definiram-se como critérios de sucesso uma satisfação média mínima de 6 em uma escala de 1 a 10 e uma taxa mínima de 70% de conclusão das tarefas principais.

5.2. Resultados

Os resultados superaram as metas estabelecidas. A Tabela 1 sintetiza os principais indicadores quantitativos. Do lado da experiência do usuário, 95,3% avaliaram a interface como fácil de usar (48,8% com nota máxima), 84,6% completaram todas as tarefas propostas e a experiência geral recebeu nota 9 ou 10 em 76,9% das respostas. O *Net Promoter Score* aproximado de 81 indica alta disposição de recomendação da plataforma. No que diz respeito ao impacto formativo, 67,5% dos usuários relataram ter aumentado seu interesse pelos temas abordados e 90,4% avaliaram positivamente o potencial da plataforma para motivar o estudo da geração de dados sintéticos, sinal de que os objetivos pedagógicos foram alcançados além da mera usabilidade.

Do lado técnico, a telemetria de infraestrutura acompanhou em tempo real o consumo de CPU, o uso de memória, o tráfego de rede e o tempo de resposta das APIs, distribuídos entre os nós. O backend Cloud AutoDroid apresentou 100% de sessões sem falhas críticas, com apenas 8 erros registrados em mais de mil execuções conduzidas pelos *workers*, uma taxa de falha inferior a 1%, evidência de robustez sob carga distribuída. Considerados em conjunto, esses ciclos capacitaram mais de 195 usuários ativos ao longo do projeto.

Qualitativamente, a interface foi elogiada por ser intuitiva, organizada e responsiva, e o material didático diversificado foi valorizado. Dados de telemetria, como mapas de calor gerados pelo Microsoft Clarity, evidenciaram interação intensa nas seções “Explorar Ferramentas”, “Gerar Dados” e “Aprender”, com alta assertividade nos botões de ação.

Tabela 1. Principais indicadores da avaliação (ciclo de abril de 2025).

Indicador	Valor
Usuários participantes	76
dos quais residentes *Hackers do Bem*	26
Interface avaliada como fácil de usar	95,3%
Conclusão de todas as tarefas	84,6%
Experiência geral com nota 9 ou 10	76,9%
Net Promoter Score (aprox.)	81
Interesse em formação em IA para cibersegurança	96%
Aumento de interesse pelos temas após o uso	67,5%
Avaliação positiva do potencial motivacional	90,4%
Sessões sem falhas críticas no backend	100%
Erros em mais de mil execuções	8

5.3. Limitações

A avaliação também revelou limitações que orientam a evolução da ferramenta. Cerca de 15,4% dos usuários não completaram todas as atividades, indicando barreiras pontuais de usabilidade, clareza das instruções ou fatores técnicos. Do ponto de vista pedagógico, alguns usuários relataram dificuldade na interpretação das métricas apresentadas no laboratório de experimentação, e a conexão entre a geração de dados sintéticos e o combate a malwares Android permaneceu abstrata para parte do público iniciante. Foram ainda apontados problemas pontuais de *layout* em determinadas resoluções, falhas ocasionais de áudio em vídeos e atritos no fluxo de *login*. Esses achados alimentam diretamente o ciclo de melhoria contínua da plataforma.

6. Disponibilidade

O Malware DataLab está disponível publicamente, com documentação, materiais e artefatos acessíveis a partir do sítio do projeto[1]. A plataforma é integralmente baseada em software livre e de código aberto, e adota práticas de desenvolvimento modular que favorecem sua replicação por instituições de ensino e laboratórios de pesquisa. O acesso é realizado via interface web, com autenticação por conta Google (Firebase) ou por credencial da Comunidade Acadêmica Federada (CAFe) da RNP. As APIs REST e GraphQL, acompanhadas de um SDK publicado no NPM, permitem a integração com sistemas educacionais e projetos de pesquisa externos. Contatos e suporte podem ser obtidos pelo endereço `malwaredatalab@gmail.com`.

7. Conclusão e Trabalhos Futuros

Este artigo apresentou o Malware DataLab, uma plataforma web integrada que unifica ensino-aprendizagem e experimentação prática com geração de dados sintéticos para detecção de malware Android. Ao articular um laboratório pedagógico estruturado com um laboratório de experimentação apoiado pelo gerador MalSynGen e pelo backend distribuído Cloud AutoDroid, a ferramenta reduz simultaneamente a curva de aprendizado das técnicas de IA generativa e a barreira computacional de sua execução. A avaliação

[1] `https://malwaredatalab.github.io`

com 76 usuários, incluindo residentes do programa *Hackers do Bem*, indicou alta taxa de conclusão de tarefas (84,6%) e elevada satisfação (NPS aproximado de 81), confirmando a efetividade da plataforma tanto no ensino quanto na experimentação.

Como trabalhos futuros, planeja-se aprimorar a interpretabilidade das métricas do laboratório de experimentação, ampliar os conteúdos pedagógicos, introduzir mecanismos de gamificação e engajamento, e integrar o assistente conversacional Jarvas [Rodrigues et al. 2024] para facilitar o uso via aplicativos de mensagens. No plano técnico, prossegue a otimização do backend para sustentar alta demanda e a incorporação modular de novos geradores, como o MalDataGen [Nogueira et al. 2025]. Por fim, pretende-se explorar a aplicabilidade da plataforma em domínios correlatos que também dependem de dados de qualidade e segurança, como saúde e finanças.

Agradecimentos

O desenvolvimento do Malware DataLab é fruto de um esforço coletivo e interdisciplinar, com apoio do Programa P&D do *Hackers do Bem*. A Rede Nacional de Ensino e Pesquisa (RNP) é parceira estratégica da iniciativa, apoiando com infraestrutura e integração de serviços, como a autenticação CAFe. Agradecemos ainda aos bolsistas, voluntários e residentes que participaram das etapas de desenvolvimento, validação e testes da plataforma.

Referências

AI & Data Today (2023). Top 10 reasons why ai projects fail. `https://www.aidatatoday.com/top-10-reasons-why-ai-projects-fail`. Acesso em: 10 jul. 2026.

Assolin, J. et al. (2022). DroidAutoML: uma ferramenta de AutoML para o domínio de detecção de malwares android. In *Anais Estendidos do XXII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg)*, pages 135–142, Porto Alegre, RS, Brasil. SBC.

Casola, K., Oleques, K., Mansilha, R. B., and Kreutz, D. (2023). DroidAugmentor: ferramenta de treinamento e avaliação de cGANs para geração de dados sintéticos. Technical report, Universidade Federal do Pampa (UNIPAMPA).

Fava, F. B. et al. (2024). Assessing the performance of docker in docker containers for microservice-based architectures. In *32nd Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, Dublin, Ireland. IEEE.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851.

Laviola, L. F., Mansilha, R. B., and Kreutz, D. (2024). Cloud AutoDroid: uma arquitetura de backend para executar serviços de ia generativa na nuvem. In *Anais Estendidos da Escola Regional de Engenharia de Software (ERES)*, Porto Alegre, RS, Brasil. SBC.

Laviola, L. F., Mansilha, R. B., and Kreutz, D. (2025). Cloud AutoDroid: Um sistema distribuído escalável para execução de ferramentas de ia generativa. In *Anais Estendidos do XLIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, Porto Alegre, RS, Brasil. SBC.

Meijin, L. et al. (2022). A systematic overview of android malware detection. *Applied Artificial Intelligence*, 36(1):2007327.

Miranda, T. d. C. et al. (2022). Debiasing android malware datasets: How can i trust your results if your dataset is biased? *IEEE Transactions on Information Forensics and Security*, 17:2182–2197.

Nogueira, A. G. D., Paim, K. O., Bragança, H., Mansilha, R. B., and Kreutz, D. (2024a). Geração de dados sintéticos tabulares para detecção de malware android: um estudo de caso. In *Anais do XXIV Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg)*, Porto Alegre, RS, Brasil. SBC.

Nogueira, A. G. D., Paim, K. O., Bragança, H., Mansilha, R. B., and Kreutz, D. (2024b). MalSynGen: redes neurais artificiais na geração de dados tabulares sintéticos para detecção de malware. In *Anais Estendidos do XXIV Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg)*, Porto Alegre, RS, Brasil. SBC.

Nogueira, A. G. D., Paim, K. O., Bragança, H., Mansilha, R. B., and Kreutz, D. (2025). MalDataGen: A modular framework for synthetic tabular data generation in malware detection. In *Anais Estendidos do Simpósio Brasileiro de Cibersegurança (SBSeg)*, Porto Alegre, RS, Brasil. SBC.

Nogueira, A. G. D., Paim, K. O., Mansilha, R. B., and Kreutz, D. (2024c). Otimização de hiperparâmetros da DroidAugmentor para geração de dados sintéticos de malware android. In *Anais da Escola Regional de Redes de Computadores (ERRC)*, Porto Alegre, RS, Brasil. SBC.

Paim, K. O., Nogueira, A. G. D., Mansilha, R. B., and Kreutz, D. (2024). Uma introdução sobre redes adversárias generativas (gans) e suas aplicações na cibersegurança. In *UniHacker: Tecnologias e Desafios em Cibersegurança I*. Disponível em: `https://unihacker-club.github.io/files/unihacker_ebook_2.pdf`.

Pinheiro Cinelli, L. et al. (2021). Variational autoencoder. pages 111–149. Springer International Publishing, Cham.

Rodrigues, R. N., Ferreira, L., Mansilha, R. B., and Kreutz, D. (2024). Jarvas: um chatbot assistente via aplicativos de mensagens instantâneas para aplicações científicas. In *Anais Estendidos da Escola Regional de Engenharia de Software (ERES)*, Porto Alegre, RS, Brasil. SBC.

Schepke, C. et al. (2024). DinD-Bench: impacto de contêineres docker em docker para a programação paralela. In *Anais da XXIV Escola Regional de Alto Desempenho da Região Sul (ERAD-RS)*, Porto Alegre, RS, Brasil. SBC.

Siqueira, G. et al. (2022). Avaliação de ferramentas de AutoML em datasets de detecção de malwares android. In *Anais do XXII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg)*, pages 302–315, Porto Alegre, RS, Brasil. SBC.

Soares, T. et al. (2021a). Detecção de malwares android: datasets e reprodutibilidade. In *Anais da XIX Escola Regional de Redes de Computadores (ERRC)*, pages 43–48, Porto Alegre, RS, Brasil. SBC.

Soares, T. et al. (2021b). Detecção de malwares android: levantamento empírico da disponibilidade e da atualização das fontes de dados. In *Anais da XIX Escola Regional de Redes de Computadores (ERRC)*, pages 49–54, Porto Alegre, RS, Brasil. SBC.

Sonco, L., Laviola, L. F., Mansilha, R. B., and Kreutz, D. (2024). Uma gui para hackers do bem aprenderem sobre malwares sintéticos. In *Anais Estendidos da Escola Regional de Engenharia de Software (ERES)*, Porto Alegre, RS, Brasil. SBC.

Vilanova, L. et al. (2022). ADBuilder: uma ferramenta de construção de datasets para detecção de malwares android. In *Anais Estendidos do XXII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg)*, pages 143–150, Porto Alegre, RS, Brasil. SBC.