



UNIVERSIDADE FEDERAL DO PARÁ
CAMPUS UNIVERSITÁRIO DE CASTANHAL
FACULDADE DE ENGENHARIA DA COMPUTAÇÃO

LUCAS DEJARD MOREIRA MENDONÇA

EMULADORES PARA TESTES EM AMBIENTES WEB2:
uma revisão da arte

Castanhal – Pará
2025

LUCAS DEJARD MOREIRA MENDONÇA

EMULADORES PARA TESTES EM AMBIENTES WEB2:

uma revisão da arte

Trabalho de Conclusão de Curso apresentado à Faculdade de Engenharia da Computação, do Campus Universitário de Castanhal, da Universidade Federal do Pará, como requisito parcial para obtenção do título de Bacharelado(a) em Engenharia da Computação. Orientadora: Dra. Hieda Adriana Nascimento Silva

Castanhal – Pará
2025

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD
Sistema de Bibliotecas da Universidade Federal do Pará
Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)

- M539e Mendonça, Lucas Dejard Moreira.
 Emuladores para testes em ambientes web2 : Uma revisão da
 arte / Lucas Dejard Moreira Mendonça. — 2025.
 27 f. : il. color.
- Orientador(a): Prof^ª. Dra. Hieda Adriana Nascimento Silva
 Trabalho de Conclusão (Graduação) - Universidade Federal do
 Pará, Campus Universitário de Castanhal, Faculdade de Engenharia
 da Computação, Castanhal, 2025.
1. Emuladores. 2. Ambiente Web2. 3. Teste de software. I.
 Título.

CDD 003

LUCAS DEJARD MOREIRA MENDONÇA

EMULADORES PARA TESTES EM AMBIENTES WEB2:

uma revisão da arte

Trabalho de Conclusão de Curso apresentado à Faculdade de Engenharia da Computação, do Campus Universitário de Castanhal, da Universidade Federal do Pará, como requisito parcial para obtenção do título de Bacharelado(a) em Engenharia da Computação.

Data da aprovação: ____/____/____

Conceito: _____

BANCA EXAMINADORA

Profa. P.h.D. Hieda A. Nascimento Silva
Orientadora - Universidade Federal do Pará

Prof. Dr. Thiago A. Sidônio Coqueiro
Membro interno - Universidade Federal do Pará

Profa. Dra. Yomara Pinheiro Pires
Membro interno - Universidade Federal do Pará

A todos que me auxiliaram na busca pelo
conhecimento e em especial **minha mãe**, que
nunca me deu uma aula, mas sempre foi minha
melhor Professora.

AGRADECIMENTOS

Este artigo só foi possível de ser realizado com o apoio de inúmeras pessoas durante toda minha trajetória acadêmica e pessoal, entre tantas creio que algumas não poderiam deixar de ser citadas, a meus professores que foram notáveis conselheiros durante minha vida e pessoas que me inspiram e me deram muitas oportunidades, especialmente *Teacher* Cleber que me deu oportunidade de concluir meu curso de inglês e descobrir minha vocação e paixão pela docência e minha orientadora Hieda Burnage a qual eu possuo profunda admiração, preciso agradecer pelo apoio de todos meus amigos em especial meus melhores amigos Jhon Victor e Laura Mayumi, por último preciso agradecer a minha família que me apoiou de maneira integral durante toda minha vida, a minhas tias que me incentivaram ao hábito da leitura e da busca pelo conhecimento desde muito pequeno, ao meu pai que me apresentou a tecnologia, e a minha mãe, sem ela nada disso seria possível, toda minha trajetória acadêmica, todos os ensinamentos de vida e todo o cuidado e amor que recebi foram cruciais para que hoje eu fosse capaz de realizar esse trabalho, a pessoa mais importante da minha vida e a quem eu tenho uma gratidão eterna, esse trabalho é além de um artigo uma extensão da experiência coletiva que pôde ter vida através das minhas mãos, sem vocês e tantos outros nada disso seria possível.

A minha avó Maria Julia da Silva, o ser mais bondoso com o qual já tive contato, a pessoa mais cuidadosa, atenciosa e carinhosa que um dia eu conheci, nunca vou te esquecer pois **hoje e para sempre carrego um pouco da sua história comigo.**

“In the end, glorification of splendid underdogs is nothing other than
the glorification of the splendid system that makes them so.”
(W. Adorno, 1951, *Minima Moralia*).

“Eles acham que venceram, provemos a eles o contrário”
(Swain, 2010, *League of legends*).

“Palavra não é privilégio de algumas pessoas, mas o direito de todos.”
(Freire, 1987, Pedagogia do oprimido).

RESUMO

Este trabalho aborda a temática dos emuladores para testes em ambientes *Web 2.0*, destacando a importância e os desafios associados a essa abordagem. Diante das características interativas e colaborativas dessa nova fase da *web*, surge a necessidade de métodos eficazes para avaliar sua robustez e desempenho. A relevância deste contexto reside na influência direta sobre o desenvolvimento de aplicações web modernas, nas quais a experiência do usuário é central, e a atualização contínua do produto é fundamental para o sucesso. Nesse cenário, aplicações interativas e colaborativas demandam métodos eficientes para avaliar robustez e desempenho, exigindo ferramentas que auxiliem em sua criação e manutenção. Os emuladores destacam-se nesse processo, permitindo a simulação de ambientes próximos aos do usuário final, o que os torna instrumentos valiosos para testes de *software*. Devido à sua importância, este trabalho investiga emuladores adequados para testes na *Web 2.0*, visando ampliar o conhecimento sobre métodos de avaliação nesse contexto.

Palavras-chave: Teste de *Software*. Emuladores. *Web 2.0*.

ABSTRACT

This paper addresses the issue of emulators for testing in Web 2.0 environments, highlighting the importance and challenges associated with this approach. Given the interactive and collaborative characteristics of this new phase of the web, there is a need for effective methods to evaluate its robustness and performance. The relevance of this context lies in its direct influence on the development of modern web applications, in which the user experience is central, and continuous product updating is essential for success. In this scenario, interactive and collaborative applications demand efficient methods to evaluate robustness and performance, requiring tools that assist in their creation and maintenance. Emulators stand out in this process, allowing the simulation of environments close to those of the end user, which makes them valuable instruments for software testing. Due to their importance, this paper investigates emulators suitable for testing in Web 2.0, aiming to expand knowledge about evaluation methods in this context.

Keywords: Software Testing. Emulators. Web 2.0.

LISTA DE SIGLAS E ABREVIATURAS

CPU	Central Processing Unit
UX	User Experience
AVD	Android Virtual Device
API	Application Programming Interface
PC	Personal Computer
IOT	Internet Of Things
BIOS	Basic Input/Output System
ROM	Read Only Memory
QA	Quality Assurance

SUMÁRIO

1 INTRODUÇÃO.....	12
2 INTRODUÇÃO A EMULADORES PARA <i>WEB 2.0</i>.....	12
2.1 Uso de emuladores e descobertas sobre efetividade.....	12
3 CONCEITOS IMPORTANTES DA <i>WEB 2.0</i>.....	13
4 TESTE DE <i>SOFTWARE</i>, IMPORTÂNCIA E LIMITAÇÕES.....	15
5 METODOLOGIAS.....	16
6 DESENVOLVIMENTO.....	17
6.1 Emuladores de dispositivos móveis.....	17
6.1.1 Sistema <i>Android</i>	18
6.1.1.1 <i>Android Virtual Device</i>	18
6.1.1.2 <i>Bluestacks</i>	19
6.1.2 <i>Ios</i>	20
6.1.2.1 <i>Máquina Virtual Hackintosh e Xcode</i>	21
6.2 Computadores <i>desktop</i>.....	21
6.2.1 <i>VirtualBox</i>	21
6.3 Emuladores de sistemas <i>IOT</i>.....	22
6.4 Emuladores de <i>Game Consoles</i>	23
6.5 Evoluções e novas tendências.....	24
7 CONCLUSÃO.....	26
7.1 Conclusão.....	26
7.2 Trabalhos futuros.....	26
8 REFERÊNCIAS.....	27

1 INTRODUÇÃO

A evolução incessante da internet trouxe consigo a transição para uma era marcada pela interatividade, colaboração e participação ativa do usuário. A *Web 2.0*, como popularmente é conhecida, não apenas redefiniu a maneira como as informações são apresentadas, mas também transformou profundamente a dinâmica digital, impulsionando a criação coletiva de conteúdo e a formação de comunidades online.

A *Web 2.0* representa um marco na história da internet, deslocando o paradigma de uma web estática para um ambiente altamente interativo e colaborativo. Caracterizada pela emergência de redes sociais, *wikis*, *blogs* e outras ferramentas participativas, a *Web 2.0* não apenas conecta pessoas ao redor do mundo, mas também fornece um terreno fértil para a criação conjunta de conteúdo, promovendo a troca de ideias e experiências.

É nesse espaço que a necessidade de assegurar a qualidade do produto é essencial para grandes empresas, e os emuladores se apresentam como possibilidade de baratear custos, aumentar a velocidade de testes e diversificar a equipe responsável.

2 INTRODUÇÃO A EMULADORES PARA WEB 2.0

Um emulador é um programa de *software* que replica o comportamento de uma arquitetura de computador ou sistema de *hardware* específico (Phillips, 2010; Page, 2009). Ele permite que os usuários executem programas projetados para uma plataforma em outra, simulando a funcionalidade do hardware original (Simonák; Jakubco, 2008). Os emuladores normalmente interpretam as instruções da CPU e fornecem dispositivos simulados em nível de *hardware* para entrada e saída, traduzindo entre o host e os sistemas emulados (Phillips, 2010). Eles são úteis no desenvolvimento, teste e suporte a software incompatível com máquinas (Simonák; Jakubco, 2008). Embora os emuladores pretendam aparecer como o processador real para a execução do software, eles podem tomar medidas simplificadoras para fornecer essa ilusão e seu uso para auxiliar no processo de produção de *software* precisa ser avaliado para melhor desempenho e funcionalidade a qual está sendo empregado.

2.1 Uso de emuladores e descobertas sobre efetividade

Os emuladores são amplamente utilizados em pesquisas na *web* para testes *mobile* e de *software*, mas sua validade em comparação com dispositivos nativos é debatida. Zhang et al.

(2014) descobriram que os emuladores podem não replicar com precisão características importantes como desempenho e a percepção do usuário em dispositivos nativos, sugerindo cautela no seu uso. No entanto, Levulis e Harris (2015) relataram apenas pequenas diferenças nos tempos de conclusão de tarefas e fixações entre testes de emuladores e dispositivos. Com o aumento da utilização e adoção das redes móveis ao passar dos anos para as mais diversas atividades, Satoh (2004) propôs uma abordagem de “emulador voador” para testar software móvel em diferentes redes, enfrentando os desafios dos ambientes sem fio 4G. Para emuladores de CPU, Martignoni et al. (2013) desenvolveram uma metodologia de teste baseada em *fuzzing* para identificar defeitos comparando estados emulados e físicos da CPU. Embora os emuladores ofereçam vantagens no teste rápido de vários dispositivos, os pesquisadores devem estar cientes das possíveis limitações e considerar o contexto específico do teste ao decidir entre as abordagens de teste com emulador ou com dispositivos nativos.

3 CONCEITOS IMPORTANTES DA WEB 2.0

A história da *web* começou em 1989, quando Tim Berners-Lee inventou a *World Wide Web*, permitindo a interconexão de documentos via *hyperlinks*. Inicialmente, a web era estática (*Web 1.0*), focada em disponibilizar informações sem muita interação com sites como *GeoCities* e *Yahoo Directory*.

Figura 1 - *Site Geocities Home Page*, 1996



Fonte: Arquivo.pt (2025)

Figura 2 - Site Yahoo Directory Home Page, que funcionava como um indexador de *links*.



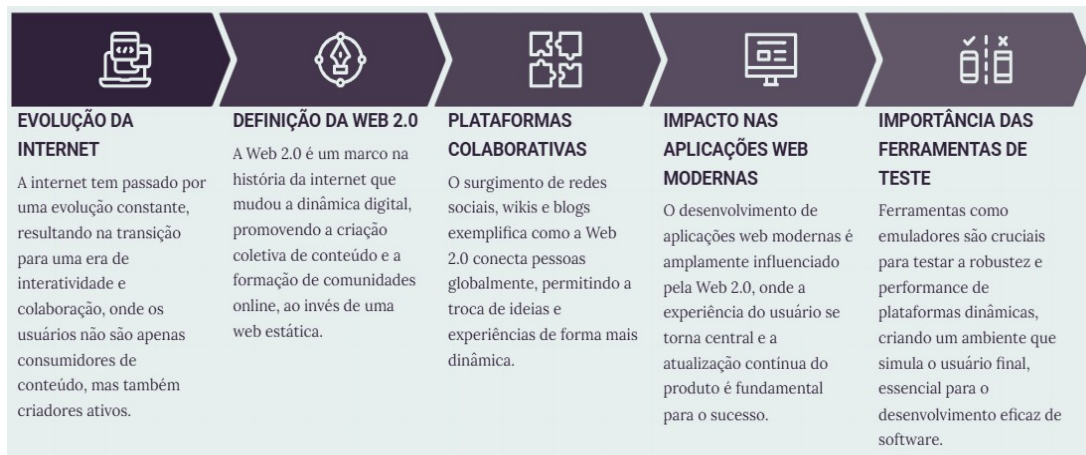
Fonte: The verge (2025)

Com o tempo, evoluiu para a *Web 2.0*, marcada por interatividade, colaboração e conteúdo gerado por usuários em plataformas como blogs e redes sociais, e atualmente, estamos caminhando para a *Web 3.0*, que promete maior descentralização e uso de tecnologias como *blockchain* e inteligência artificial.

A *Web 2.0* se destaca por suas características fundamentais, que incluem a interatividade, o conteúdo gerado pelos usuários e o compartilhamento em redes sociais. Plataformas como *blogs*, *wikis*, redes sociais e serviços de compartilhamento de mídia (como YouTube e Netflix) exemplificam essa nova abordagem da web. A *Web 2.0* também trouxe a adoção em massa de tecnologias como a programação assíncrona, que permitem a atualização de conteúdo na página sem a necessidade de recarregar o site, proporcionando uma experiência mais fluida e responsiva aos usuários.

Além disso, com a utilização dos smartphones por grande parte da população, a implementação da web mobile foram criadas técnicas na produção de *software* como o design responsivo, que ajusta o *layout* das páginas conforme o tamanho da tela, e o desenvolvimento de *Progressive Web Apps* (PWAs), que combinam as vantagens de sites e aplicativos nativos. A utilização desses dispositivos mudou a forma como consumimos conteúdo e serviços *online*, trazendo desafios e inovações.

Figura 3 - Infográfico evolutivo da *Web* e suas características.



Fonte: Reprodução do autor (2025)

4 TESTE DE SOFTWARE, IMPORTÂNCIA E LIMITAÇÕES

O teste de software é uma fase crucial no ciclo de vida de desenvolvimento de *software*, visando melhorar a confiabilidade, o desempenho e outros fatores-chave definidos nas especificações de requisitos de software (Mithelesh Parihar; Anu Bharti, 2019). Recomenda-se iniciar os testes no início do processo de desenvolvimento para evitar dificuldades na correção de bugs em fases posteriores (Devi, 2012).

Os testes servem múltiplos objetivos, incluindo detecção de defeitos e estimativa de confiabilidade, embora não possam garantir a ausência de erros (Ahamed, 2010). O processo envolve a comparação dos resultados reais com os resultados esperados usando vários dados de teste (Saini; Kestina, 2013). Mesmo com possíveis limitações, os testes continuam a ser essenciais para medir a qualidade do *software* e garantir a satisfação do cliente, uma vez que os usuários procuram sempre uma melhor UX na utilização desses produtos (Schrepp *et al.*, 2023).

Apesar de todos os benefícios apresentados, atualmente o tempo necessário para realizar o processo de teste é apresentado como principal barreira para sua implementação (Rodrigues, Andreia, 2010), problema que somado ao custo inicial alto para implementação de casos de teste, compra de ferramentas de automação de teste, e capacitação de equipe responsável tornam a implementação do teste de *software* mais complexa e menos vantajosa. Além disso, a automação de testes apresenta algumas barreiras adicionais como a falta de tempo na criação das automações (Wilklund, Kristian, 2017) além da manutenção de testes automatizados ser entendida como um problema pois a mudança na tecnologia e evolução dos

produtos de *software* dificultam a manutenção de testes automatizados (Quadri; Farooq, 2010).

5 METODOLOGIA

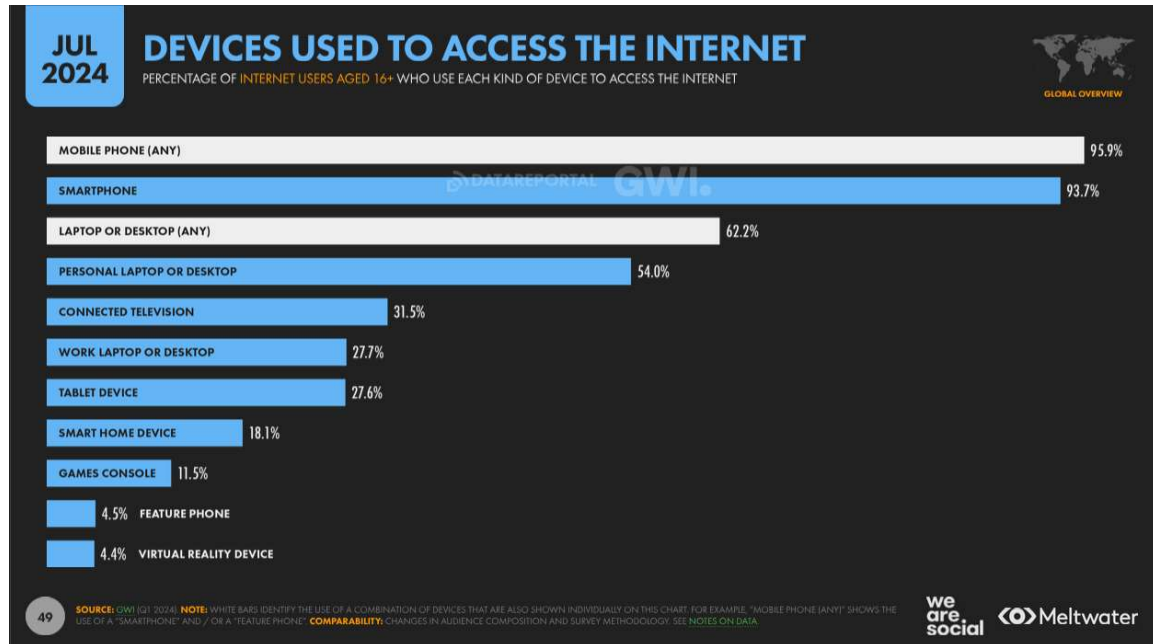
Diante da pesquisa (Datareportal, 2024) é possível entender que a *web* atual é acessada majoritariamente pelos dispositivos *mobile* (*smartphones e tablets*), e computadores, além de televisores, *Smart home devices* e *game consoles* com um número considerável de acessos o que permite traçar os dispositivos mais importantes para se emular no intuito de melhorar o teste de *software*. Cada um desses dispositivos possui suas características que foram levadas em consideração na hora de se emular, com destaque ao sistema operacional usado, que dependendo do dispositivo podem ser diversos e com diferentes versões.

Para uma melhor criação de produtos digitais sempre deve ser usada a versão mais recente recomendada pelos fabricantes e desenvolvedores dos sistemas operacionais, portanto para os estudos realizados foram utilizadas sempre versões mais recentes recomendadas.

Os emuladores emergem como ferramentas promissoras nesse cenário, proporcionando um ambiente controlado para a realização de testes abrangentes em um contexto que reflete fielmente a dinâmica da *Web 2.0*. Esta escolha visa contribuir para o avanço da área de testes de *software* em um ambiente digital em constante evolução.

Para entender e aplicar os emuladores no processo de teste de *software* este trabalho apresenta dentro dos dispositivos mais influentes na *web 2.0* os principais ambientes, ferramentas de emulação disponíveis para eles e suas aplicações, além de prós e contras, por fim é exposto neste trabalho um grande apanhado de evoluções e novas tendências na área da emulação. Podendo dessa forma ser uma pedra angular no processo de produção de *software* com auxílio de emuladores para mitigação de falhas.

Figura 4 - Pesquisa *Data Reportal* de dispositivos usados para acessar a internet no ano de 2024, destacando a predominância de dispositivos mobile, seguidos de *desktop*, IOT e *game*



consoles.

Fonte: Digital 2024 July Global Statshot Report | Datareportal. GWI. (2025)

Dessa forma, para entender e aplicar os emuladores no processo de teste de software, este trabalho apresenta os principais ambientes e ferramentas de emulação disponíveis para os dispositivos mais influentes na *Web 2.0*, além de suas aplicações, e suas características mais relevantes, as recomendações de uso dos sistemas, os emuladores mais populares e suas vantagens e desvantagens, além dos critérios legais que envolvem o uso desses *softwares*.

Por fim, é exposto neste trabalho um panorama das evoluções e novas tendências na área da emulação. Dessa forma, o artigo pode ser uma base fundamental no processo de produção de software com o auxílio de emuladores para a mitigação de falhas.

6 DESENVOLVIMENTO

6.1 Emuladores de dispositivos móveis

Os dispositivos móveis predominantes da internet atual são os *smartphones* e *tablets* e neles se concentram o foco da pesquisa de emuladores para dispositivos móveis, quando se procura emular estes ambientes entende-se que existem poucas diferenças entre eles, sendo assim, a característica mais relevante que precisamos levar em consideração é o seu sistema operacional, atualmente os sistemas mais utilizados são *Android* e *IOS* e quanto a sua emulação podemos destacar pontos importantes na hora de se criar uma emulação.

6.1.1 Sistema *Android*

O sistema operacional Android é desenvolvido pelo Google, baseado no núcleo Linux. Foi projetado principalmente para dispositivos móveis com telas sensíveis ao toque, como *smartphones* e *tablets*. Lançado pela primeira vez em 2008, o Android rapidamente se tornou o sistema operacional móvel mais popular do mundo, devido à sua flexibilidade e à vasta gama de dispositivos que o utilizam. Ele suporta uma grande variedade de aplicativos, que podem ser baixados da *Google Play Store*, e é conhecido por sua personalização, permitindo que os usuários adaptem a interface e as funcionalidades de acordo com suas preferências. Além disso, o Android é de código aberto, o que significa que desenvolvedores e fabricantes de dispositivos podem modificá-lo e adaptá-lo às suas necessidades específicas, contribuindo para sua ampla adoção global.

Os emuladores de *Android*, como o *Android Studio* ou *BlueStacks*, simulam o comportamento de um dispositivo móvel, permitindo que os usuários experimentem a interface, realizem depurações e vejam como os aplicativos se comportam em diferentes versões do sistema e em várias configurações de *hardware*. Além disso, essa emulação oferece uma maneira prática de explorar o Android sem comprometer a integridade de um dispositivo real, sendo uma ferramenta valiosa para o teste de *software*.

6.1.1.1 *Android Virtual Device*

O AVD é uma ferramenta poderosa que permite criar e configurar dispositivos virtuais com diferentes versões do *Android*, tamanhos de tela, resoluções e outras especificações de *hardware*. Esses dispositivos virtuais funcionam como um emulador, simulando o ambiente de um dispositivo Android real no seu computador. Isso facilita o teste e a depuração de aplicativos em diversas configurações sem a necessidade de ter vários dispositivos físicos. Além disso, o AVD permite que os desenvolvedores testem funcionalidades específicas, como diferentes APIs e versões do sistema operacional, garantindo que o aplicativo funcione corretamente em uma ampla gama de dispositivos *Android*.

Figura 5 - Emulador AVD em execução simulando um *smartphone* Pixel 3A API level 34 (referente ao *Android* 14) executado em dispositivo *Linux*

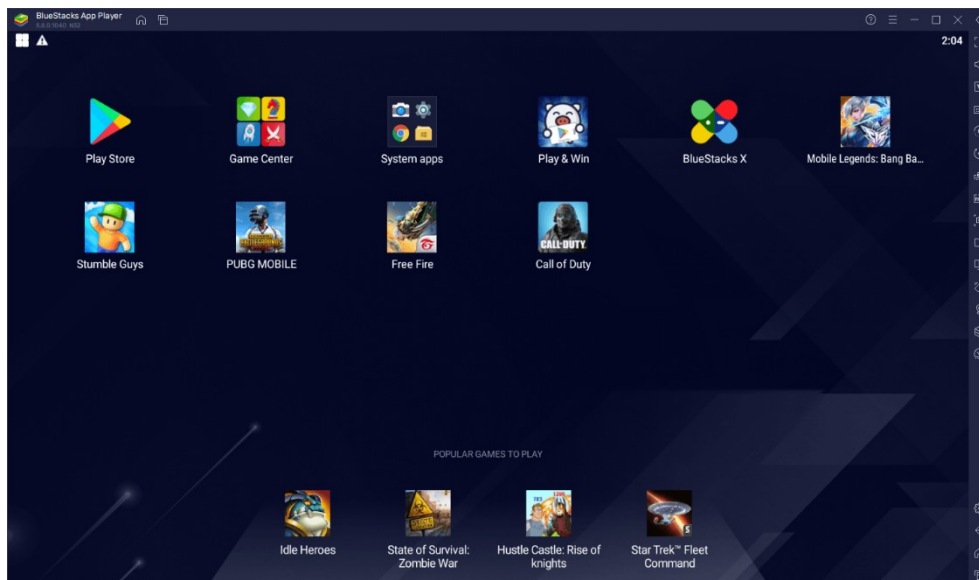


Fonte: Reprodução própria (2025)

6.1.1.2 BlueStacks

O *BlueStacks* é um popular emulador que permite rodar o sistema operacional *Android* em um computador com *Windows* ou *macOS*. Ele cria um ambiente virtual que simula o *Android*, permitindo que os usuários acessem e utilizem aplicativos móveis diretamente no PC, como jogos, redes sociais e outros aplicativos disponíveis na *Google Play Store*. A emulação com o *BlueStacks* é especialmente útil para quem deseja testar aplicativos, jogar jogos mobile com teclado e *mouse*, ou simplesmente aproveitar a experiência *Android* em uma tela maior. Além disso, o *BlueStacks* oferece diversas funcionalidades, como a capacidade de mapear teclas, gravar a tela e executar múltiplas instâncias do *Android*, tornando-o uma ferramenta versátil tanto para desenvolvedores quanto para usuários comuns.

Figura 6 - Interface de menu do emulador *Bluestacks* com múltiplos jogos instalados para execução.



Fonte: *Bluestacks* (2025)

6.1.2 Sistema IOS

O sistema iOS, desenvolvido pela Apple Inc., é o sistema operacional móvel utilizado nos dispositivos *iPhone* e *iPad*. Lançado em 2007, ele é conhecido por sua interface intuitiva e desempenho otimizado, além de sua integração com o ecossistema *Apple*, algo que ajudou a torná-lo um dos principais sistemas operacionais móveis do mundo. A arquitetura do iOS é baseada no núcleo *Unix*, com camadas de *software* que garantem segurança, desempenho e suporte a aplicativos de terceiros. Ao contrário do *Android*, o *IOS* se destaca pelo seu ambiente de desenvolvimento fechado e controlado, garantindo maior proteção contra

malware, ao custo de restrições para desenvolvedores, que precisam seguir as diretrizes da *Apple* para distribuir aplicativos na *App Store* e a emulação desse sistema só é possível através de um *software apple*, o *MacOS*, por esses motivos a emulação do *IOS* encontra fortes barreiras, já que segundo a empresa em sua licença de uso do *software IOS/IPADOS 18* (Apple Inc, 2024): “...Você não deve (e concorda que não o fará nem possibilitará que outros o façam) copiar (exceto se expressamente permitido por esta Licença), descompilar, fazer engenharia reversa, desmontar, tentar derivar o código fonte, descriptografar, modificar ou criar obras derivadas do *Software Apple*”. O que torna a emulação do *IOS* algo não recomendado e cria consequências para os usuários que tentarem realizar essas emulações. Entretanto, ainda assim existem possibilidades de se arriscar nesta empreitada.

6.1.2.1 Máquina Virtual Hackintosh e Xcode

Hackintosh refere-se ao processo de instalar sistemas operacionais nativos de computadores *Apple* em outros sistemas operacionais. É comum essa instalação ser realizada em uma máquina virtual no *Windows*. A partir daí, pode-se usar o *Xcode*, ele contém o simulador de *iOS* embutido, onde você pode executar aplicativos *iOS*.

É importante ressaltar que esses emuladores *iOS* no *Windows* geralmente não têm o mesmo desempenho e compatibilidade que um simulador nativo no *macOS*. Para desenvolvimento sério, é altamente recomendável usar um dispositivo *Apple*. Ademais, essa solução pode violar os termos de serviço da *Apple*, então deve ser usada com cautela.

6.2 Emulação de máquinas *desktop*

6.2.1 *VirtualBox*

O *VirtualBox* é uma plataforma de virtualização de código aberto desenvolvida pela *Oracle* que permite a criação de máquinas virtuais podendo rodar sistemas operacionais de forma isolada em um único dispositivo físico. Seu uso é amplamente adotado tanto em ambientes acadêmicos quanto em corporativos, devido à sua versatilidade e compatibilidade com diversas plataformas, incluindo *Windows*, *macOS* e *Linux*. A ferramenta possibilita que usuários executem múltiplos sistemas operacionais simultaneamente, sendo especialmente útil para o desenvolvimento, teste e ensino de diferentes ambientes de software sem a necessidade de múltiplos computadores físicos. Além disso, o *VirtualBox* suporta snapshots, que permitem

restaurar uma máquina virtual a um estado anterior, facilitando o processo de experimentação e resolução de problemas durante o uso. Essas características fazem do *VirtualBox* uma solução popular para estudantes e profissionais que precisam simular cenários de rede e segurança, ou testar *software* em diferentes plataformas de maneira eficiente e econômica.

Figura 7 - Emulador *VirtualBox* em execução instalado em uma máquina *Windows* com sistema *linux* sendo emulado



Fonte: Youtube (2025)

6.3 Emuladores de sistemas IOT

O desenvolvimento e o teste de aplicações IoT enfrentam desafios devido à complexidade e ao custo da construção de redes físicas de IoT. Para resolver isso, várias estruturas de emulação foram propostas; NIOTE que gera dados de sensores e atuadores para fácil configuração e implantação sobre protocolos TCP/IP e MQTT (Martin, et al,2020). A E-IoT virtualiza as operações dos dispositivos IoT, fornecendo ambientes operacionais reais e simulando capacidades de *hardware* (KHALED, 2022). IoTNetEMU combina componentes de código aberto para oferecer uma ferramenta de validação para desenvolvedores e pesquisadores, minimizando a lacuna entre simulações e experiências ao vivo (Oliveira,2023). Para emulação em grande escala, o MAMMoTH foi proposto para abordar as limitações das soluções existentes no teste de redes com milhões de nós simultâneos (LOOGA,2012). Esses emuladores visam facilitar o desenvolvimento de aplicativos IoT, otimizar o tempo de desenvolvimento e melhorar a qualidade do produto, fornecendo ambientes de teste realistas

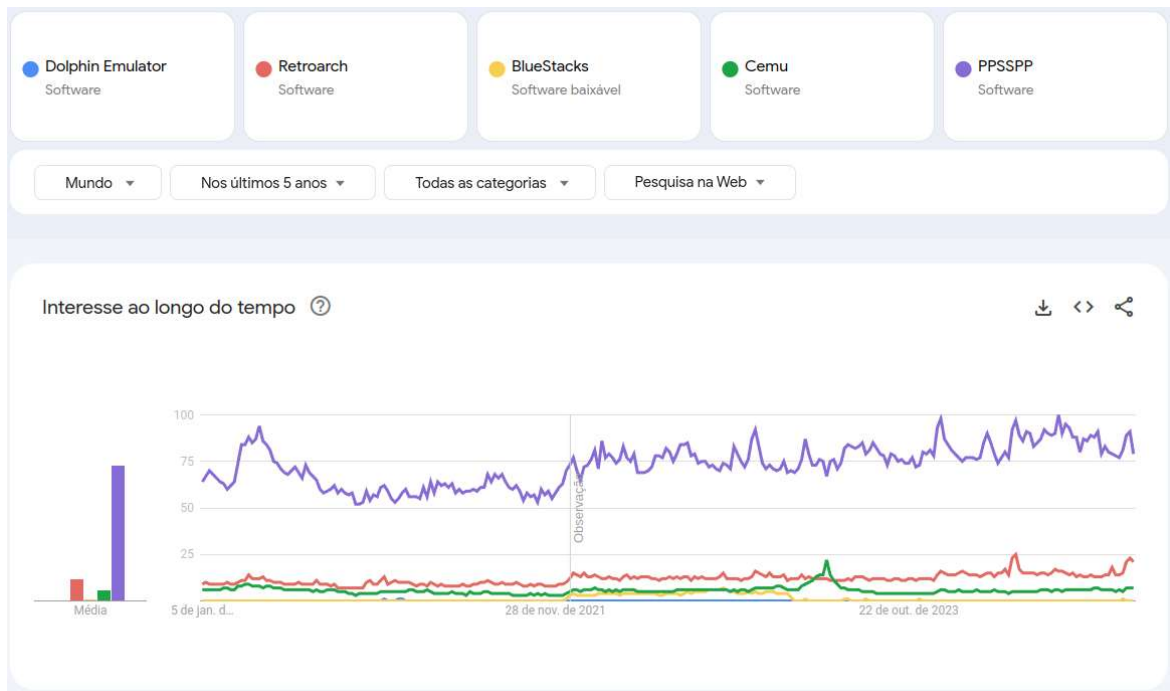
sem a necessidade de redes físicas de IoT.

6.4 Emuladores de game consoles

O uso de emuladores para consoles de videogame enfrenta um grande debate no que diz respeito a sua legalidade e por tanto para realizar a emulação desse software dificilmente se encontra um padrão e um software recomendado pelas grandes empresas de *games*, além do mais a criação desses *softwares* enfrenta ainda problemas jurídicos e legais, proibindo a criação e distribuição desses emuladores de maneira livre. Entretanto, ainda é possível encontrar na internet esses programas que utilizam técnicas avançadas de engenharia reversa para reproduzir a arquitetura do console original, traduzindo as instruções binárias dos jogos para um formato compatível com o sistema hospedeiro que geralmente é um *Desktop*. Embora sejam amplamente utilizados para preservação histórica de jogos e facilitação de pesquisas acadêmicas sobre tecnologias passadas, os emuladores frequentemente enfrentam debates jurídicos e éticos relacionados à propriedade intelectual, especialmente no que diz respeito ao uso de BIOS e ROMs que possam infringir direitos autorais. Dessa forma, eles ilustram um campo interdisciplinar que combina tecnologia, preservação cultural e debates legais sobre os limites do uso de *software* original.

Na busca de emuladores mais usados foi utilizada uma pesquisa de interesse do *google* com os emuladores mais famosos de suas respectivas plataformas e podemos perceber uma predominância do PPSSPP um emulador do console PSP da empresa de *videogames sony interactive entertainment*, esses emuladores em grande maioria podem ser utilizados em um computador e alguns possuem também suporte para dispositivos *android*. Entretanto, a predominância de uso dessas ferramentas não se dá para criação, manutenção ou teste de jogos e afins e sim para utilização de *software* pirata.

Figura 8 - Pesquisa com o gráfico representativo aos emuladores para *game consoles* mais pesquisados no *Google*.



Fonte: <https://trends.google.com.br/trends/> (2025)

6.5 Evoluções e novas tendências

O desenvolvimento e o teste de aplicações com o passar dos anos apresenta sempre novas ferramentas, e com os emuladores não foi diferente, atualmente os emuladores que antes rodavam em máquinas locais, hoje, com o advento da *Cloud Computing* podem ser utilizados remotamente por múltiplos usuários, grandes empresas de computação em nuvem hoje fornecem ferramentas de testes manuais e automatizados em nuvem; É o caso da gigante *Amazon* que possui o serviço de *device farms*, que permite o usuário criar múltiplos dispositivos para executar o *software* em nuvem, além de permitir execução concorrente de múltiplas instâncias.

Para testes mobile a empresa dominante no mercado é a *Browserstack* que oferece os mesmos serviços da *Amazon*, porém com um foco maior nos dispositivos móveis, com além de emulações dispositivos reais controlados remotamente.

Todas essas ferramentas se tornam aliadas de desenvolvedores, QA's e demais profissionais da área na busca de melhorar a qualidade do *software* desenvolvido, e se apresentam como um futuro promissor para uma área em constante expansão e que demanda que profissionais de todas as partes do mundo se conectem na realização do trabalho.

Figura 9 - *Pixel device farm* privado da *BigTech Uber*, todos esses dispositivos podem ser acessados remotamente para testes e demais etapas de desenvolvimento pela equipe.



Fonte: *Twitter* (2025)

7 CONCLUSÃO E TRABALHOS FUTUROS

7.1 Conclusão

Com essa pesquisa realizada acerca dos emuladores mais usados e atualizados disponíveis, foi possível compreender de que maneira se estrutura o papel dessas ferramentas no processo de desenvolvimento, manutenção e teste dos *softwares* na *web* 2.0, podendo dessa forma ser uma pedra angular para o futuro dessa área, no que tange emulações para testes. Pode-se então concluir que ferramentas de emulação podem e devem ser usadas no processo de testagem desses *softwares* e alcançam performance muito similar ao produto original, além de ter vantagens significativas quanto à manutenção e distribuição, permitindo o teste independente para múltiplos usuários de maneira remota, esse projeto apresenta ainda as dificuldades enfrentadas na emulação de *software* proprietário, e os problemas legais enfrentados por quem opta pela emulação desses sistemas.

Todas essas informações apresentadas de maneira coesa e imparcial corroboram a importância da utilização dos emuladores e contribuem para a avaliação da comunidade científica a respeito do tema.

7.2 Trabalhos futuros

A relevância deste artigo se demonstra nas múltiplas linhas de pesquisa disponíveis para contínua expansão da área pesquisada, hoje com a recente transição da *web* 2.0 para o início da 3.0 é importante entender quais os emuladores pioneiros nessa nova versão da *web*, também é possível compreender atualmente como atuam os emuladores em nuvem e buscar diferentes visões dessa área que já é hoje uma realidade das “*Big Techs*”, é possível buscar se aprofundar nos padrões de projetos para a utilização de emuladores para testes, dessa forma compreender as diferentes maneiras de implementar essas ferramentas em projetos diversos, em um campo com crescimento constante, os emuladores se apresentam como uma ferramenta muito útil e passível de pesquisa e aprendizado para a execução de testes.

REFERÊNCIAS

- APPLE INC. **iOS and iPadOS Software License Agreement**. Apple Inc., 2024. Disponível em: https://www.apple.com/legal/sla/docs/iOS18_iPadOS18.pdf. Acesso em: 7 jan. 2025.
- DATAREPORTAL. **Digital 2024 July Global Statshot Report**. Datareportal, 2024. Disponível em: <https://datareportal.com/reports/digital-2024-july-global-statshot>. Acesso em: 7 jan. 2025.
- KHALED, Ahmed E.; ASHFAQ, Mohammed; SURESH, Sebin Puthenthara. E-IoT: Emulator to develop and test IoT application. In: 2022 IEEE 8th World Forum on Internet of Things (WF-IoT). **IEEE**, 2022. p. 1-6.
- LEVULIS, S. J.; HARRIS, D. J. Are all tests equal? A comparison of emulator and device testing for mobile usability evaluation. In: **Proceedings of the Human Factors and Ergonomics Society Annual Meeting**. Vol. 59. [s.l.]: [s.n.], 2015. p. 976-980.
- LOOGA, Vilen et al. Mammoth: A massive-scale emulation platform for internet of things. In: 2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems. **IEEE**, 2012. p. 1235-1239.
- MARTIGNONI, L. et al. Path-exploration lifting: hi-fi tests for lo-fi emulators. In: **ASPLOS XVII**. [s.l.]: [s.n.], 2012.
- MARTIN, Andrés Armando Sánchez et al. Emulator for data analytics and IoT development projects. **Visión Electrónica**, v. 14, n. 1, p. 57-63, 2020.
- OLIVEIRA, João; SOUSA, Filipe; ALMEIDA, Luís. IoTNetEMU-A Framework to Emulate and Test IoT Applications. In: **Proceedings of the 19th ACM International Symposium on QoS and Security for Wireless and Mobile Networks**. 2023. p. 33-37.
- PHILLIPS, G. Simplicity betrayed. **Queue**, v. 8, p. 10-20, 2010.
- RODRIGUES, Andreia; BESSA, Adriano; PINHEIRO, Plácido Rogério. Barriers to implement test process in small-sized companies. In: World Summit on Knowledge Society. Berlin, **Heidelberg: Springer Berlin Heidelberg**, 2010. p. 233-242.
- SATOH, I. Software testing for wireless mobile computing. **IEEE Wireless Communications**, v. 11, p. 58-64, 2004.
- SCHREPP, Martin et al. **On the importance of UX quality aspects for different product categories**. [s.l.: s.n.], 2023.
- SIMONÁK, S.; JAKUBCO, P. **Software-based CPU emulation**. [s.l.: s.n.], 2008.
- WIKLUND, Kristian et al. Impediments for software test automation: A systematic literature review. Software Testing, **Verification and Reliability**, v. 27, n. 8, p. e1639, 2017.
- ZHANG, D.; LAI, J.; DASH, A. An empirical investigation of emulators in mobile web research. **Journal of Computer Information Systems**, v. 54, p. 67-76, 2014.