

# **Comparação do desempenho e usabilidade do Android Studio segundo o sistema operacional usado**

*Android Studio performance and usability comparison by operating system used*

Renato José da Silva Camões,

Fábio Lúcio Lopes Mendonça,

Rafael Melo Peres

## **Resumo**

*Este trabalho tem como objetivo realizar comparações técnicas de usabilidade entre dois dos principais sistemas operacionais, Windows e Linux (Ubuntu), utilizando como base a aplicação Android Studio por meio de virtualização dos sistemas. A principal função de um sistema operacional é controlar o funcionamento de um computador assim como o gerenciamento dos recursos Hardware/Software disponíveis no terminal. Atualmente existem três sistemas operacionais conhecidos e usados no mundo, como é o caso de Windows, Linux e iOS, no entanto o presente trabalho se concentra na análise comparativa de usabilidade entre Windows e Linux devido à quantidade de usuários que trabalham sobre esses sistemas operacionais. A análise comparativa proposta permite diferenciar a natureza da arquitetura de cada um dos sistemas operacionais escolhidas, por uma parte a arquitetura licenciada no caso de Windows e uma arquitetura open-source, como é o caso do Linux na sua distribuição Ubuntu. Com a ferramenta de desenvolvimento Android Studio foi possível comparar a usabilidade nos dois sistemas operacionais, detalhando as vantagens e desvantagens de cada um através da metodologia proposta nesse documento.*

**Palavras – chave:** Sistema Operacional; Android Studio; Windows; Linux; Ubuntu;

## **Abstract**

*Operating system is a program or set of programs that communicates between hardware and other software, however complex it may seem, an operating system becomes a set of routines executed by the processor, like any other program. Its main function would be to control the operation of the computer as well as manage various resources available in the system. One of the main operating systems most used by users would be Windows, as it would be the easiest computer system to organize and users can carry out their respective tasks without many complications and which has greater compatibility with the drivers of most hardware. Ubuntu continues as one of the safest and fastest operating systems on the market without the need for more powerful hardware, it has an extremely positive point for Linux users (Ubuntu) because its system is based on Free Software (Open Source), where experienced users or laymen can customize whatever they want in the system, even create programs based on other programs and distribute it. This work aims to perform technical comparisons of usability between two of the main operating systems, Windows and Linux (Ubuntu), using the Android Studio application as a basis through system virtualization.*

**Keywords:** operating system, comparatives, Windows, Linux, Ubuntu

## **INTRODUÇÃO**

O Sistema Operacionais (SO) é parte fundamental na operação e desempenho de um computador, garantindo confiabilidade, eficiência e segurança por parte do SO aos usuários e sistemas dependentes. O sistema operacional poderia se considerar como um gestor do computador, organizando a execução dos aplicativos, alocando espaço em memória para a execução dos programas, gerenciando a disputa de recursos, tratando o envio e recebimento de dados dos usuários com a máquina dentre outras funções importantes. (SOUSA, 2014)

Nos inícios da ciência da computação, os computadores não possuíam sistema operacional, apenas existindo o hardware do computador. Sendo assim, os sistemas operacionais foram criados com a finalidade de tornar mais fácil e mais eficiente a utilização do computador. Um dos primeiros sistemas operacionais que tiveram destaque, o CTSS, desenvolvido pelo Centro Computacional Instituto de Tecnologia Massachusetts (MIT). (UAB.IFSUL, 2017)

O computador não é útil sem um SO instalado, já que, é responsável pelo funcionamento do computador que de fato se tornou muito popular após a criação do sistema operacional Windows, permitindo o acesso de milhões de pessoas aos computadores pessoais. Uma das principais características apresentadas no Windows, é sua interface gráfica e usabilidade baseada em itens visuais, como ícones, janelas e menus. Tudo isso se deve a principal linguagem de programação usada para escrever o código-fonte das várias versões do Windows o C/C++ e Assembly, tornando mais simples o funcionamento, onde o usuário não teria a necessidade de utilizar comandos para navegar no sistema. (KEIZER, 2019)

Linguagens de programação possuem a capacidade de executar qualquer tarefa, suas implementações podem limitar um pouco o que elas podem acessar no hardware onde vão rodar. A programação sem um sistema operacional seria em linguagem de máquina, que traria dificuldades, onde o programador teria que programar todos os componentes do hardware, um a um e cada aplicação teria que ser detalhada em um nível muito baixo para o que o hardware funcione, um exemplo: O sistema operacional Windows é essencialmente escrito em linguagem de programação C.

O objetivo deste artigo é descrever a preparação do ambiente de programação ou ambiente de desenvolvimento para o Android Studio que considera as questões dos SO Linux e Windows, onde em ambos é enfatizado suas semelhanças, diferenças e características, destacando a execução da aplicação em cada sistema operacional e verificando o funcionamento da mesma em testes técnicos com as mesmas características de hardware e software, podendo atribuir de forma prática em qual sistema operacional o Android Studio tem melhor desempenho e assim extrair os resultados de usabilidade, desempenho da máquina.

## **AMBIENTE DE PROGRAMAÇÃO**

O ambiente de programação também conhecido como ambiente de desenvolvimento é o local onde o trabalho de construção e codificação de uma solução Software será realizada. Esse ambiente é integrado pelos elementos necessários que irão permitir aos especialistas desenvolverem um determinado

sistema. O ambiente de programação é constituído por elementos como *templates*, ferramentas e diretrizes.

Essas ferramentas são essenciais para a implementação de um determinado projeto envolvendo programação, por isso, aprimorar/melhorar/ajustar esse ambiente é fundamental para organizar melhor o trabalho. É importante destacar que o ambiente de programação é diferente da linguagem de programação, o primeiro consiste na interface de trabalho, enquanto o segundo corresponde a um conjunto de códigos semânticos.

Dentro do ambiente, além da interface e os códigos semânticos, existe o papel importante dos programadores que tem conseguido desenvolvimentos cada vez mais complexos, diminuindo a quantidade de pessoas e o tempo requerido para a criação de um software, devido principalmente às ferramentas disponíveis. Conhecer um conjunto de ferramentas de programação e frameworks torna a atividade do desenvolvedor mais fácil e aumenta a produtividade do profissional, tendo a disposição as ferramentas necessárias para programação segundo as necessidades da solução.

A ferramenta de programação é um programa ou aplicativo por um desenvolvedor para criar, depurar, manter, ou realizar algum outro tipo de apoio para a criação de outros programas e aplicativos. Algumas das ferramentas mais utilizadas que podem ser encontradas em um ambiente de programação segundo PROGRAMAÇÃO PROGRESSIVA (2017) são:

- **Compilador:** De acordo com o que o próprio nome indica, ele tem o intuito de compilar o código-fonte do programa para, dessa maneira, transformá-lo em uma linguagem mais adequada para o trabalho;
- **Editor:** Essa ferramenta exerce a função de editar o código-fonte do programa que está no ambiente de programação;
- **Modelagem:** Responsável pela elaboração e desenvolvimento de modelos de classes, dos objetos e das interfaces que estão envolvidas no desenvolvimento do software. Para tornar o trabalho de programação mais eficiente, organizado e rápido é importante adotar maneiras que melhoram esse ambiente de programação. Sendo assim, um deles consiste na melhor exibição e disposição dos elementos. Inclusive, hoje em dia é possível que os desenvolvedores possam utilizar um ambiente de desenvolvimento baseado em *cloud computing*. Esse ambiente é o Cloud9, que disponibiliza depurador para inspeção de código em tempo real. Existem vários ambientes de programação, mas os dois mais utilizados e conhecidos pelos programadores são:
  1. **NetBeans:** Ele é um ambiente de desenvolvimento do tipo open *source*, ou seja, ele é gratuito. Além dessa vantagem, ele conta também com um suporte técnico em idiomas variados, incluindo o português, o japonês e o russo e chinês. Esse ambiente além de suportar vários frameworks de PHP, também disponibiliza para você opções de personalização que facilitam e agilizam os projetos de programação.
  2. **Sublime:** Esse é um ambiente bastante requisitado para arquivos individuais. Quando o assunto é personalização, ele oferece uma

série de plugins interessantes, que tornam o desenvolvimento de programas com mais opções e recursos.

Segundo Erick (2013) o mercado de trabalho é dividido em duas vertentes: Trabalhos *open-source* e trabalhos corporativos. Os trabalhos corporativos geralmente trabalham com sistemas e tecnologias privadas, aqueles sistemas que são confidenciais por questões de segurança e existem diversas pessoas das quais fazem a manutenção. Enquanto aos profissionais da vertente *open-source* focam em divulgar seus trabalhos, projetos e códigos fonte.

Em ambos os casos existem vantagens e desvantagens, mas a característica mais relevante no ambiente de desenvolvimento é compartilhar conhecimento. Existem discussões e debates sobre a escolha dos usuários ou desenvolvedores sobre o melhor sistema operacional para desenvolvimento, o Linux (Ubuntu) ou Windows, talvez seja uma discussão que pode continuar por vários anos.

## DIFERENÇAS ENTRE OS SISTEMAS OPERACIONAIS WINDOWS E LINUX

Linux e Windows são sistemas operacionais que podem ser usados em computadores domésticos, empresas ou servidores. Além das diferenças técnicas tem também a relação do uso de licença. As principais diferenças fazem referência à flexibilidade que o Linux oferece, uma vez que existem várias distribuições com foco diferente como por exemplo uma versão mais leve para computadores antigos, uma que já vem preparado para quem trabalha com edição de vídeos e imagem. Existem distribuições focadas no usuário doméstico, ou seja, é possível encontrar uma distribuição Linux apropriada para as necessidades do usuário.

Por outro lado, o Windows não oferece essa flexibilidade, tendo no mercado versões para usuários particulares e outra para servidores. O Suporte oficial do Windows é através de *help-desk* pago, onde o usuário pode encontrar as respostas de perguntas mais frequentes, tendo diversos fóruns nos quais usuários ativos resolvem algumas das principais dúvidas (TODADIFERENCA, 2018).

As comparações entre os sistemas operacionais e suas funcionalidades se apresentam na Tabela 1:

Principais Diferenças	Windows	Linux
<b>Linha de Comando:</b>	O Windows possui uma preferência por alterações através de interface gráfica, assim, o acesso a linha de comando não é intuitivo.	O Linux por ser um sistema <i>Open Source</i> , prioriza as modificações do usuário o que torna natural que a linha de comando fique evidente e de fácil acesso.
<b>Suporte aos Usuários:</b>	O Windows possui um suporte feito através do Help Desk pago, que facilita sanar as dúvidas dos usuários de imediato.	O Linux possui uma ampla rede de fóruns de suporte gratuito, porém se o problema for urgente, depende dos usuários disponíveis.
<b>Acesso ao Código Fonte:</b>	O Windows possui seu código-fonte restrito, ou seja, o usuário não tem o direito de fazer qualquer alteração no código do kernel do sistema.	O Linux foi desenvolvido com base no sistema GNU (Licença Pública) o que significa que qualquer usuário pode ter acesso ao código-fonte do sistema.

<b>Usabilidade:</b>	O Windows sempre tenta inovar com sua interface gráfica porem mantendo um padrão de seus antecessores, onde melhora o acesso e visualização das informações em que mantem os comandos de usabilidade das versões anteriores.	O Linux deixou de ser um sistema para programadores baseado em linhas de comando. Há várias versões do sistema com interfaces intuitivas que facilitam a usabilidade tanto para usuários experientes quanto para novos usuários.
<b>Liberdade ao uso de Licença:</b>	O Windows caso o usuário comprou somente uma licença de uso, não poderá utilizar a mesma em outra máquina e não pode desenvolver aplicativo e comercializá-lo no sistema.	O Linux tem uma maior autonomia para que você possa fazer cópias de distribuição sem problemas e que tem liberdade para criar um aplicativo e comercializá-lo sem autorização da empresa.

Tabela 1: principais diferenças entre o Windows e Linux -Fonte: (IMPACTA, 2016)

Visto que o sistema operacional é uma variável chave para questões de desenvolvimento, onde determinado programa necessita de um sistema para o desenvolvimento de uma aplicação, nesse trabalho se propõe a ferramenta Android Studio que será utilizada para a comparação de usabilidade, assim como testes monitorados em cada um dos SO, servindo como base para justificar de forma técnica em qual sistema o usuário tem maiores facilidades e menores dificuldades para usar a ferramenta proposta.

## ANDROID STUDIO

O Android Studio é o ambiente de desenvolvimento integrado (em inglês: *Integrated Development Environment* - IDE ) oficial para o desenvolvimento de apps para Android, baseado no IntelliJ IDEA. Foi anunciado em 16 de maio de 2013 na conferência Google I/O. Android Studio é disponibilizado gratuitamente sob a Licença Apache 2.0.

Além do editor de código e das ferramentas de desenvolvedor avançadas do IntelliJ, o Android Studio oferece recursos adicionais para aumentar a produtividade na compilação de apps Android, como:

- Um sistema de compilação flexível baseado em Gradle;
- Um emulador rápido com inúmeros recursos;
- Um ambiente unificado que possibilita o desenvolvimento para todos os dispositivos Android;
- A aplicação de alterações para enviar alterações de código e recursos ao aplicativo em execução sem reiniciar o aplicativo;
- Modelos de código e integração com GitHub para ajudar a criar recursos comuns de apps e importar exemplos de código;
- Frameworks e ferramentas de teste cheios de possibilidades;
- Ferramentas de *“lint”* para detectar problemas de desempenho, usabilidade, compatibilidade com versões, entre outros.

A interface do Android Studio é flexível para o usuário, tem um leque de *schema* considerável, podendo utilizar várias aparências diferentes conforme o usuário desejar. O usuário tem a liberdade da escolha de temas para a interface da aplicação, optando entre escuro e claro. Também é possível customizar os atalhos de teclado, de forma que sejam iguais a outras IDE's, como por exemplo o Eclipse

Porém, uma das desvantagens sobre o Android Studio está na instalação para determinado sistema operacional, onde pode se tornar complexo para alguns usuários, principalmente para os iniciantes, devido ao fato de ter que usar na instalação através de comandos pelo prompt do SO utilizado. (ANDROID STUDIO, 2020)

## METODOLOGIA

### INSTALAÇÃO E CONFIGURAÇÃO DO AMBIENTE COM ANDROID STUDIO

Os métodos de instalação do Android Studio consistem em seguir passos onde o usuário tem a liberdade de poder personalizar e configurar de forma simples em determinado SO antes de instalar a aplicação. O Android Studio possui compatibilidade com a maioria de sistemas operacionais Windows, Linux, Mac e Chrome OS.

Windows e no Linux são os SO que mais usam o Android Studio. O processo de instalação no Linux consiste em inserção de parâmetros disponibilizados pelos desenvolvedores no site, onde o usuário utiliza alguns códigos para usar via terminal de comando manualmente, permitindo realizar a instalação da aplicação.

Uma das principais funções para realizar a instalação da aplicação no Linux, se utiliza o comando *apt-get*, que ao ser inserido no terminal do Linux, tem como função, permitir a instalação, atualização e remoção de pacotes do sistema, onde devesse inserir um comando logo a frente da palavra: *Upgrade, Update, Install e Remove*.

Por outro lado, o processo de instalação do Android Studio no Windows, se resume em ir ao site da aplicação, baixar o arquivo instalador e executar o arquivo, tornando o processo de instalação simples para qualquer usuário, sem a necessidade de inserção de comandos ou códigos para realizar a instalação do Android Studio, como acontece com Linux.

A extensão “\*.exe” faz referência a aplicativos e scripts de softwares desenvolvidos para sistemas baseados no Microsoft Windows. Consistem em um arquivo usado para iniciar o processo de arquivos necessários para executar esse tipo de aplicações. O conteúdo de um arquivo “\*.exe” inclui código de referência e detalhes que podem ser usados para localizar e carregar arquivos associados ao aplicativo correspondente. (REVIVERSOFT, 2020)

Foram utilizados alguns recursos da máquina para poder preparar o ambiente de instalação do Android Studio, como:

- **Virtualização:** o programa de virtualização utilizado para realizar a instalação do Linux (Ubuntu) e Windows foi o Virtual Box que é um programa de sistemas operacionais no qual o usuário pode ter vários outros sistemas instalados e virtualizados em um só programa, o que não gera risco para a máquina e arquivos nela e também tem suas

principais características em que se pode compartilhar memória RAM, partição de disco e controlador gráfico.

- **Uso de Memória:** na criação dos dois ambientes Windows e Linux no programa de virtualização, foram compartilhados 4096MB (4GB) de memória RAM de um computador principal que possui 32GB de RAM, para que não haja perda ou desigualdades em testes em cada sistema operacional;
- **Partição de Disco:** cada sistema possui determinado GB para instalação na máquina virtual, então foi particionado de um HD de 1TB (terabyte), 150GB para cada sistema operacional, onde não haverá problemas de armazenamento cheio para instalar softwares e atualização dos sistemas;
- **Controlador Gráfico:** para os sistemas operacionais instalados onde eles geram pesos na máquina por possuírem interfaces gráficas dinâmicas, foi compartilhado 128MB de uma placa de vídeo GTX 1080TI para cada sistema operacional, em que o programa fara uso dos recursos 3D de hardware da máquina hospedeira;
- **Rede:** os recursos de rede são utilizados de forma automática pelo próprio programa de virtualização, em que o mesmo está utilizando 200MB de download e 15MB de upload via cabo interligado diretamente ao modem, sem a necessidade de utilizar a rede WIFI.

A aplicação Android Studio possui uma plataforma exclusiva com manuais e documentos, assim como tutoriais focados na instalação do aplicativo em cada sistema operacional compatível. onde o usuário pode acessar os passos de instalação pelo link: <https://developer.android.com/studio/install> disponibilizado pelo próprio fornecedor.

Na Tabela 2 se apresentam as facilidades e dificuldades identificadas na instalação do Android Studio para os SO escolhidos no presente trabalho.

Windows e Linux		Facilidade	Dificuldade
Instalação da aplicação:	<b>WINDOWS</b>	O Windows possui sua extensão .exe onde qualquer usuário consegue realizar a instalação sem nenhuma dificuldade somente clicando no arquivo instalável.	A instalação no Windows poderá apresentar algumas dificuldades com o executável do fornecedor, que pode exibir várias informações no instalador onde alguns contém malwares.
	<b>LINUX</b>	O Linux possui em sua interface gráfica (Ubuntu) que em alguns aplicativos não necessita a utilização de comandos via terminal para realizar a instalação, basta somente executar o instalador baixado pelo fornecedor da aplicação.	No Linux exige mais conhecimento e experiência do usuário para realizar instalações, pois em sua grande parte depende da utilização de comandos no terminal.

<b>Preparação do Ambiente Máquina:</b>	<b>WINDOWS</b>	A preparação do ambiente no Windows se torna um método mais simples e menos descomplicado, pois não necessita muito conhecimento e não ocupa muitos GB no armazenamento, sendo mais viável para qualquer usuário somente abrindo a extensão do instalador.	De certa forma pode-se dizer em que a única dificuldade que o usuário obtém na instalação pelo Windows, seria onde o fornecedor coloca anúncios em que atrapalha na instalação do Android Studio no executável do mesmo.
	<b>LINUX</b>	No Linux o ambiente se torna leve e de fácil manuseio por sua interface gráfica e velocidade, pois o sistema não exige muito do hardware o que torna mais fácil a instalação da aplicação mesmo sendo executada via terminal.	O Linux depende de uma instalação um pouco mais complexa e completa via terminal, que necessita um pouco mais de conhecimento e comandos para realizar a instalação.

Tabela 2: dificuldades e facilidades da instalação e preparação do ambiente nos sistemas

O ambiente onde foram feitos os testes técnicos e comparativos do Android Studio foi em uma máquina através de software de virtualização de SO, onde foram considerados recursos como a memória RAM, disco rígido, placa gráfica e rede de formas iguais, visando a integralidade da comparação proposta pelos autores desse trabalho.

Após a preparação do ambiente e instalação do software na máquina, foram feitos os testes de utilização da ferramenta Android Studio na máquina virtual, usando os dois SO escolhidos, obtendo os resultados da execução dos sistemas e identificando as características funcionais da execução do Android Studio, facilitando a comparação de desempenho segundo o sistema operacional usado.

## CARACTERÍSTICAS DOS PACOTES DO ANDROID STUDIO INSTALADOS

Os pacotes relacionados a seguir, são obrigatórios na instalação do Android Studio (usando o Gerenciador SDK do Android) para o funcionamento e compilação correta da aplicação, por meio de um dispositivo móvel ou virtual:

- **Android SDK Manager:** O SDK Manager é o gerenciador de atualizações da plataforma Android. Ele oferece ferramentas, plataformas e outros componentes do **SDK** necessários para desenvolver os aplicativos (FELLIPE, 2016);
- **Android SDK Platform-Tools 29.0.5:** O SDK Platform-Tools do Android é um componente do SDL do Android. Ele inclui ferramentas que fazem interface com a plataforma Android, como adb, fastboot e systrace. Essas ferramentas são necessárias para o desenvolvimento de apps para Android. Elas também serão necessárias se você quiser desbloquear o carregador de inicialização do dispositivo e instalá-lo com uma nova imagem do sistema (DEVELOPERANDROID, 2020);



- **Android SDK Build Tool 30.0.0 rc1:** O Android SDK Build Tools é um componente do SDK do Android necessário para criar apps para Android (DEVELOPERANDROID, 2020);
- **Android Emulator 30.0.0:** O Android Emulator simula dispositivos Android no seu computador. Assim, você pode testar o aplicativo em diversos dispositivos e níveis de API do Android, sem precisar ter cada dispositivo físico (DEVELOPERANDROID, 2020).

## INFORMAÇÕES DO CÓDIGO FONTE PARA TESTE

Os resultados foram gerados considerando um código fonte distribuído gratuitamente através do GitHub, usado para a execução dos testes comparativos entre os dois sistemas operacionais e o Android Studio.

A função do aplicativo compilado no Android Studio é para registro e controle de gastos de uma frota de veículos para registrar viagens, despesas e analisar métricas. Quando a aplicação é executada contém um total de quatro telas, onde cada tela tem sua função específica, como: cadastrar veículo, registrar viagens ou gastos e resumo das informações do aplicativo.

O principal motivo para a utilização do código na comparação de usabilidade e desempenho, é porque o código possui complexidades na estruturação e no armazenamento de dados gerados pela aplicação, especialmente na utilização do usuário ao realizar funções de registro, cadastro e resumo, permitindo o completo dos recursos das máquinas virtuais, gerando resultados detalhados.

## RESULTADOS

Usando a ferramenta Android Profile (que pertence a aplicação Android Studio) oferecer é possível obter dados em tempo real para ajudar a entender como a aplicação usa os recursos de CPU, memória, rede e consumo de energia.

Os comparativos dos testes foram realizados em ambos SO executados ambiente virtual, com os mesmos recursos de Memória RAM, armazenamento e placa gráfica, além de utilizar o mesmo código da aplicação que está sendo executada em tempo real por um smartphone. A Tabela 3 apresenta os resultados dos testes:

Ação	Windows	Linux
<b>Tempo de Execução do APP:</b> (o app é iniciado pela primeira vez desde que o sistema foi inicializado)	O tempo de inicialização do Android Studio é superior no Windows, chegando a demorar 43 segundos a mais que no Linux, ocasionando travamentos.	O tempo de inicialização no Linux é 43 segundos inferior ao do Windows, sem ocasionar travamentos.
<b>Execução de uma aplicação no Android Studio:</b> (execução de um app direto do código fonte pelo Android Studio)	O Windows possui atraso ao carregar e baixar os scripts e bibliotecas do Android Studio para execução da aplicação no Smartphone.	O Linux possui um pequeno atraso ao carregar e baixar os scripts e bibliotecas do Android Studio, porém o tempo de resposta é menor do que no Windows.

<p><b>Navegação e Usabilidade:</b> (utilizando e aplicando algumas funções do Android Studio para testar desempenho)</p>	<p>Ao navegar pelo Android Studio e tentar realizar alguns testes, o Windows fechava corriqueiramente somente ao tentar executar alguma função ou abrir algumas ferramentas, ocorrendo em alguns casos “Tela Azul” (fechamento do sistema).</p>	<p>A navegação pelo Linux, ocorre alguns travamentos, porém, em todos os testes feitos, não houve nenhum problema de fechamento da aplicação (Android Studio), ou problemas relacionados ao sistema operacional.</p>
<p><b>Consumo da CPU (processador):</b> (consumo da CPU com a aplicação sendo executada pela smartphone)</p>	<p>Sendo executada a aplicação no Windows, o programa se apresentou instável em sem muita perda de processamento, consumindo em média entre 10 e 14% no uso do processador.</p>	<p>Executando a mesma aplicação no Linux, apresentou a mesma estabilidade de uso que no Windows, onde está sendo consumido entre 10 a 15% do processador.</p>
<p><b>Uso da Memória RAM:</b> (o Android Studio consome uma quantidade razoável de RAM, ao executar uma aplicação pareada com smartphone)</p>	<p>Testado uma aplicação que consome mais recursos do computador no Android Studio, o Windows se apresentou mais estável consumindo somente 119MB de memória RAM, no qual não influencia no funcionamento da máquina e que não ocasiona em travamentos no programa.</p>	<p>Porém, o Linux, consome mais memória RAM para tentar executar de forma mais fluida no sistema, ele consome 136,5MB de RAM (na mesma aplicação do teste do Windows), mas isto gradativamente ocasiona em pequenas oscilações na execução da aplicação, o que não atrapalha na usabilidade dela.</p>

Tabela 3: resultados comparativos no Android Studio entre o Windows e Linux

Com os resultados obtidos se observa que um dos sistemas operacionais se manteve estável e apresentou melhores resultados na execução do código fonte no Android Studio, tendo melhor desempenho para execução e programação no aplicativo virtualizado.

- **Tempo de Execução do aplicativo:** o tempo de inicialização do Android Studio, mostra-se inferior no Linux, que apresenta 43 segundos a menos no tempo de abertura da aplicação em comparação ao Windows.
- **Execução de uma aplicação no Android Studio:** em comparação ao tempo de execução da aplicação entre os dois sistemas, o Linux apresenta melhores resultados devido ao baixo tempo de resposta em comparação ao Windows.
- **Navegação e Usabilidade:** no sistema de navegação e usabilidade no Android Studio entre os sistemas operacionais, o Windows apresenta uma grande instabilidade de performance e desligamento da máquina, fator que não ocorre no Linux, onde, apresenta boa instabilidade com leves travamentos, mas não houve desligamento da máquina.
- **Consumo da CPU (processador):** nesse fator de consumo do processador, os dois sistemas operacionais têm o mesmo resultado (diferença entre 1% na utilização da CPU nos sistemas) que apresentam uma média entre 10 a 15% de uso da CPU.
- **Uso da Memória RAM:** o uso de memória RAM tem uma diferença de utilização na aplicação, onde apresentou uma melhor performance no

Windows, que consome menos RAM do que o Linux, pois isso influencia no funcionamento da aplicação quando o código está em execução, que no Windows apresenta menos oscilações e travamentos do que no Linux.

## CONCLUSÃO

LINUX/WINDOWS se tornou mais viável para a utilização do programa Android Studio via máquina virtual, pois, os elementos apresentados para testes foram de mais eficácia, melhor usabilidade em geral, menos consumo dos componentes e menor tempo de execução para inicialização do aplicativo no sistema.

Manter o ambiente de programação para o desenvolvimento de um aplicativo implica considerar o funcionamento e desempenho da máquina, fatores nos quais são relevantes a utilização de um sistema operacional, pois, o mesmo depende de vários fatores para executar tal tarefa sem ocorrer complicações na execução da aplicação, considerando os componentes utilizados na máquina.

Os resultados obtidos servem de alicerce para programadores ou até mesmo usuários que buscam melhorar o desempenho e agilidade em um sistema com máquina virtual, considerando a preparação adequada do ambiente de desenvolvimento do sistema operacional utilizando.

O sistema operacional Linux (Ubuntu) possui maiores vantagens na utilização de aplicações para o desenvolvimento de aplicações mobile em relação ao Windows, porem a instalação do Android Studio no Linux é mais complexa que no Windows, o que se torna uma dificuldade para alguns usuários programarem no ambiente Linux.

No Linux percebeu-se que existe maior rapidez e fluidez ao utilizar o Android Studio, porem o Linux consome mais memória RAM do que no Windows, mas a aplicação é executada com menos tempo no Linux e outros dois fatores importantes como: navegação e tempo de resposta, o que de modo em geral o torna melhor para utilização do que Windows numa máquina virtual.

## REFERÊNCIAS

**5 PRINCIPAIS diferenças entre Windows e Linux.** Impacta, 2016. Disponível em: <<https://www.impacta.com.br/blog/2016/08/03/5-principais-diferencas-entre-windows-e-linux/>>. Acesso em: 03 de abr. de 2020.

**APT-GET.** guialinux, 2013. Disponível em: <<https://guialinux.uniriotec.br/apt-get/>>. Acesso em: 17 de abr. de 2020.

**CONHEÇA o Android Studio.** developer.android, 2020. Disponível em: <<https://developer.android.com/studio/intro?hl=pt-br/>>. Acesso em: 05 de jun. de 2020.

CORDEIRO, Fellipe. **Gerenciamento de Pacotes com Android SDK Manager.** androidpro, 2016. Disponível em: <[https://www.androidpro.com.br/blog/android-studio/android-sdk-manager/#O\\_que\\_e\\_o\\_Android\\_SDK\\_Manager/](https://www.androidpro.com.br/blog/android-studio/android-sdk-manager/#O_que_e_o_Android_SDK_Manager/)>. Acesso em: 03 de out. de 2020.

**DIFERENÇA entre o Windows e o Linux.** todadiferença, 2018. Disponível em: <<https://www.todadiferenca.com/diferenca/diferenca-entre-windows-e-o-linux/>>. Acesso em: 12 de abr. de 2020.

**Executar apps no Android Emulator.** developer.android, 2020. Disponível em: <<https://developer.android.com/studio/run/emulator?hl=pt-br/>>. Acesso em: 03 de out. de 2020.

**EXE extensão de arquivo.** reviversoft, 2020. Disponível em: <<https://www.reviversoft.com/pt/file-extensions/exe/>>. Acesso em: 21 de ago. de 2020.

HIDEKI, Erick. **Por que programadores preferem Linux para trabalhar?**. Ubuntudicas, 2013. Disponível em: <<https://www.ubuntudicas.com.br/2013/08/por-que-programadores-preferem-linux-para-trabalhar/>>. Acesso em: 17 de mar. de 2020.

**HISTÓRIA dos Sistemas Operacionais.** uab.ifsul, 2017. Disponível em: <<http://uab.ifsul.edu.br/tsiad/conteudo/modulo1/sop/ua/at2/03.html/>>. Acesso em: 15 de set. de 2020.

KEIZER, Gregg. **Microsoft soma 800 milhões de usuários com Windows 10.** Computerworld, 2019. Disponível em: <<https://computerworld.com.br/2019/03/23/microsoft-soma-800-milhoes-de-usuarios-com-windows-10/>>. Acesso em: 17 de fev. de 2020.

**Notas das versões do SDK Platform-Tools.** developer.android, 2020. Disponível em: <<https://developer.android.com/studio/releases/platform-tools/>>. Acesso em: 15 de out. de 2020.

**Notas das versões do SDK Build Tools.** developer.android, 2020. Disponível em: <<https://developer.android.com/studio/releases/build-tools/>>. Acesso em: 10 de out. de 2020.

**O QUE posso melhorar no meu ambiente de programação.** Programação Progressiva, 2017. Disponível em: <<https://www.programacaoprogessiva.net/2018/05/O-Que-Posso-Melhorar-Ambiente-Programacao.html/>>. Acesso em: 02 de mar. de 2020.

SOUSA, Luís Carlos. **A importância do Sistema Operacional para o bom funcionamento da máquina.** luiscarlosousaj, 2014. Disponível em: <<http://luiscarlosousaj.blogspot.com/2014/04/a-importancia-do-sistema-operacional.html/>>. Acesso em: 17 de fev. de 2020.