



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA BAHIA  
CAMPUS VALENÇA**

**CARLOS FREDERICO DOS SANTOS PELLEGRINI**

**CÓDIGO ABERTO E PIRATARIA: UMA NOVA PERSPECTIVA PARA O ACESSO  
LEGAL À TECNOLOGIA**

**VALENÇA-BA  
2025**

**CARLOS FREDERICO DOS SANTOS PELLEGRINI**

**CÓDIGO ABERTO E PIRATARIA: UMA NOVA PERSPECTIVA PARA O ACESSO  
LEGAL À TECNOLOGIA**

*Trabalho de Conclusão de Curso apresentado ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia da Bahia, como requisito parcial para a obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas.*

*Orientador: Prof. Me. André Luiz Romano Madureira*

**VALENÇA-BA  
2025**

**CARLOS FREDERICO DOS SANTOS PELLEGRini**

**CÓDIGO ABERTO E PIRATARIA: UMA NOVA PERSPECTIVA PARA O ACESSO  
LEGAL À TECNOLOGIA**

*Trabalho de Conclusão de Curso apresentado ao Curso Superior de Tecnologia em  
Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e  
Tecnologia da Bahia, como requisito parcial para a obtenção do grau de Tecnólogo  
em Análise e Desenvolvimento de Sistemas.*

Data de Aprovação: \_\_\_\_\_ / \_\_\_\_\_ / \_\_\_\_\_

**BANCA EXAMINADORA**

---

**Prof. Me. André Luiz Romano Madureira**  
(Orientador - IFBA Campus Valença)

---

**Prof. Me. Joel Eugênio Cordeiro Junior**  
(Avaliador - IFBA Campus Valença)

---

**Prof. Me. Hortevan Marrocos Frutuoso**  
(Avaliador - IFBA Campus Valença)

## **AGRADECIMENTOS**

Primeiramente, dedico minha mais profunda gratidão a Jovana, o amor da minha vida. Sua presença foi meu norte e a inspiração constante que me impulsionou a seguir em frente, mesmo nos momentos mais difíceis.

Agradeço ao meu orientador, Professor Mestre André Luiz Romano Madureira, pela sua disponibilidade, paciência e pelos ensinamentos que foram cruciais para o desenvolvimento e a qualidade desta pesquisa.

Expresso meu sincero agradecimento aos meus amigos e familiares, que, cada um à sua maneira, contribuíram com palavras de incentivo, compreensão e suporte.

Por fim, um agradecimento especial ao trio que tornou os dias mais alegres e memoráveis: Daisy, Lili e Pepe.

## RESUMO

Este trabalho analisa a pirataria de software no Brasil e propõe o uso de softwares de código aberto como alternativa ética, legal e viável. A pesquisa utiliza abordagem exploratória com métodos quantitativos e qualitativos, dividida em duas fases: mapeamento da demanda por pirataria (The Pirate Bay e Google Trends) e testes comparativos de desempenho entre softwares proprietários e livres. Os resultados mostram que, apesar de pequenas limitações em usabilidade, os softwares de código aberto apresentam desempenho técnico competitivo, além de vantagens como gratuidade, segurança e suporte comunitário. Conclui-se que essas soluções podem contribuir para reduzir a pirataria e ampliar a inclusão digital, desde que acompanhadas de políticas públicas, capacitação técnica e maior conscientização social.

**Palavras-chave:** pirataria de software; código aberto; software livre; comparação de desempenho; inclusão digital.

## ABSTRACT

This paper analyzes software piracy in Brazil and proposes the use of open-source software as an ethical, legal, and viable alternative. The research uses an exploratory approach with quantitative and qualitative methods, divided into two phases: mapping the demand for piracy (The Pirate Bay and Google Trends) and comparative performance tests between proprietary and free software. The results show that, despite minor usability limitations, open-source software presents competitive technical performance, in addition to advantages such as being free of cost, security, and community support. It is concluded that these solutions can contribute to reducing piracy and expanding digital inclusion, provided they are accompanied by public policies, technical training, and greater social awareness.

**Keywords:** software piracy; open source; free software; performance comparison; digital inclusion.

# Sumário

<b>Agradecimentos</b>	<b>4</b>
<b>1 INTRODUÇÃO</b>	<b>1</b>
<b>2 OBJETIVOS</b>	<b>1</b>
2.1 Objetivo Geral . . . . .	1
2.2 Objetivos Específicos . . . . .	1
<b>3 JUSTIFICATIVA</b>	<b>2</b>
<b>4 PIRATARIA DE SOFTWARE: UM PROBLEMA MULTIFACETADO</b>	<b>2</b>
4.1 Conceituação e Contexto . . . . .	2
4.2 Tipos Comuns de Pirataria . . . . .	3
4.3 Cenário Global e Brasileiro . . . . .	4
4.4 Impactos Econômicos e Jurídicos . . . . .	5
4.5 Dimensões Sociais e Culturais . . . . .	6
<b>5 O ECOSISTEMA DE CÓDIGO ABERTO</b>	<b>8</b>
5.1 Definições: Software Livre e Código Aberto . . . . .	8
5.2 Histórico e Evolução . . . . .	9
5.3 Licenças: As Regras do Jogo . . . . .	9
5.4 Análise Comparativa: Vantagens e Desvantagens . . . . .	11
5.4.1 Vantagens do Software Livre . . . . .	11
5.4.2 Desvantagens do Software Livre . . . . .	12
5.4.3 Considerações sobre o Software Proprietário . . . . .	12
5.5 Modelos de Negócio Sustentáveis . . . . .	12
<b>6 Procedimentos Metodológicos</b>	<b>13</b>
6.1 Abordagem da Pesquisa . . . . .	13
6.2 Etapa 1: Identificação da Demanda por Pirataria de Software . . . . .	13
6.3 Etapa 2: Análise Comparativa de Desempenho . . . . .	14

<b>7</b>	<b>Resultados e Discussão</b>	<b>14</b>
7.1	Softwares com maior demanda por pirataria . . . . .	14
7.2	Comparações de desempenho por segmento . . . . .	14
7.2.1	Premiere vs Kdenlive (Edição de vídeo profissional) . . . . .	15
7.2.2	FL Studio vs LMMS (Produção musical) . . . . .	16
7.2.3	After Effects vs Blender (Composição e modelagem 3D) . . . . .	17
7.2.4	Photoshop vs GIMP (Edição de imagem) . . . . .	18
7.2.5	Word vs Writer (Pacote de produtividade) . . . . .	19
7.2.6	PowerPoint vs Impress (Pacote de produtividade) . . . . .	20
7.2.7	Excel vs Calc (Pacote de produtividade) . . . . .	21
7.2.8	Acrobat Reader vs Okular (Leitura de PDF) . . . . .	22
7.2.9	Illustrator vs Inkscape (Design vetorial) . . . . .	23
7.2.10	AutoCAD vs FreeCAD (Modelagem técnica e CAD) . . . . .	24
7.2.11	Lightroom vs Darktable (Tratamento de imagem RAW) . . . . .	25
7.3	Comparação de desempenho de sistemas operacionais . . . . .	26
7.4	Síntese da Discussão . . . . .	29
<b>8</b>	<b>Conclusão</b>	<b>29</b>
	<b>Referências</b>	<b>30</b>



# 1 INTRODUÇÃO

O acesso à tecnologia da informação constitui, atualmente, um dos pilares para a inclusão social, a formação educacional e o desenvolvimento profissional. Contudo, em muitos países em desenvolvimento, como o Brasil, esse acesso ainda é limitado por fatores econômicos, sociais e culturais que dificultam a aquisição legal de softwares amplamente utilizados no mercado. Nesse cenário, a pirataria de software tornou-se uma prática recorrente e complexa, envolvendo não apenas a violação de direitos autorais, mas também questões estruturais de desigualdade e exclusão digital.

Apesar dos avanços na legislação e nos mecanismos de fiscalização, os altos custos de licenciamento de programas proprietários, aliados à falta de políticas públicas eficazes de incentivo à tecnologia acessível, contribuem para a perpetuação da informalidade digital. Muitos usuários, especialmente estudantes e profissionais de baixa renda, veem na pirataria a única alternativa para utilizar ferramentas essenciais ao estudo, ao trabalho e à produção cultural. Tal realidade desafia não apenas o campo jurídico, mas também o ético, o econômico e o educacional.

Este trabalho propõe uma análise crítica e comparativa entre softwares proprietários amplamente pirateados e suas alternativas de código aberto. A pesquisa busca compreender os motivos que levam tantos usuários a recorrer à pirataria, bem como avaliar se as soluções livres são, de fato, capazes de substituí-las com qualidade e eficiência. Para isso, serão considerados critérios técnicos, como desempenho e usabilidade, e também aspectos culturais, legais e sociais relacionados ao uso de softwares.

A proposta é fomentar o debate sobre caminhos éticos e sustentáveis para o acesso à tecnologia da informação, destacando o potencial transformador do software livre em contextos de vulnerabilidade digital. Acredita-se que a disseminação e valorização dessas ferramentas podem contribuir para a construção de uma cultura de legalidade, inovação e inclusão, alinhada aos princípios de justiça social e desenvolvimento tecnológico colaborativo.

## 2 OBJETIVOS

### 2.1 Objetivo Geral

Realizar uma análise crítica e comparativa de como os softwares de código aberto podem se apresentar como alternativas tecnológicas viáveis e legítimas em substituição aos softwares proprietários pirateados, examinando aspectos como usabilidade, segurança, acessibilidade, sustentabilidade e implicações legais, com o propósito de fomentar o acesso ético e democrático à tecnologia da informação.

### 2.2 Objetivos Específicos

- Identificar e categorizar os tipos de softwares proprietários mais comumente pirateados no contexto brasileiro e internacional, investigando os fatores socioeconômicos, culturais e tecnológicos que contribuem para essa prática.

- Mapear e destacar alternativas de código aberto que possam substituir os softwares mais pirateados, destacando suas funcionalidades, comunidades de desenvolvimento e modelos de atualização e suporte.
- Realizar uma análise comparativa entre softwares pirateados e suas alternativas open source, com base em critérios técnicos e qualitativos como usabilidade, segurança, estabilidade, interoperabilidade, curva de aprendizado e suporte técnico.
- Discutir os impactos legais, éticos e de segurança decorrentes do uso de softwares pirateados, promovendo uma reflexão sobre o papel do software livre como vetor de inclusão digital, inovação colaborativa e acesso legal à tecnologia.

### **3 JUSTIFICATIVA**

A pirataria de software no Brasil, impulsionada por fatores econômicos e culturais, justifica a busca por alternativas legais e acessíveis. O software de código aberto surge como uma solução viável, oferecendo gratuidade, segurança e autonomia tecnológica, capaz de desmistificar a suposta inferioridade em relação às opções pagas. Este estudo se propõe a analisar, de forma prática, o potencial do software livre para combater a pirataria, promover a inclusão digital e subsidiar a criação de políticas públicas mais eficazes e democráticas.

## **4 PIRATARIA DE SOFTWARE: UM PROBLEMA MULTIFACETADO**

### **4.1 Conceituação e Contexto**

A violação dos direitos de propriedade intelectual associados a programas de computador é denominada "pirataria de software". Segundo Teixeira (2018), essa prática constitui uma forma de contrafação, ou seja, uma falsificação. O autor destaca que, embora a pirataria pudesse ocorrer fora do ambiente digital, foi a internet que potencializou exponencialmente o alcance e a frequência desse delito. A facilidade com que a infração pode ser cometida é um fator agravante, visto que a cópia e a distribuição de programas pirateados podem ser realizadas com conveniência por qualquer agente que disponha de um computador conectado à rede.

A pirataria de software compreende práticas como a cópia e uso não autorizado de programas protegidos por direitos autorais, sendo que a fiscalização e aplicação das leis contra esse tipo de infração podem variar consideravelmente entre diferentes países. Em consonância com essa definição, Waykar e Lad (2019) ressaltam a abrangência da pirataria, enquanto a BSA – THE SOFTWARE ALLIANCE (2017) destaca que até práticas como a duplicação de softwares e sua distribuição entre amigos são ilegais, mesmo quando feitas de maneira informal. As fontes mencionadas também indicam que muitos usuários estão cientes da ilegalidade dessas ações, ainda que as perpetuem.

Contudo, para além de uma perspectiva estritamente legal ou técnica, a pirataria pode ser analisada sob um viés socioeconômico. Karaganis (2011, apud (FERES; OLIVEIRA; GONÇALVES, 2017)) argumenta que o fenômeno representa, em sua essência, um “problema mundial

de preços”. Segundo o autor, a sua expansão é impulsionada por uma combinação de fatores, entre os quais se destacam o elevado custo de mídias digitais como softwares, a baixa renda média em países em desenvolvimento, a ampla difusão de tecnologias que facilitam a cópia e as rápidas transformações nos padrões culturais e de consumo.

Apesar da existência de legislação específica para o combate a essas práticas em diversas nações, a efetividade da sua aplicação ainda é um ponto de atenção. Conforme apontam Dornelas e Clem (2024), embora muitos países possuam leis antipirataria, a implementação e fiscalização dessas normas variam significativamente, o que demonstra a complexidade da abordagem global do problema.

## 4.2 Tipos Comuns de Pirataria

A pirataria de software pode se manifestar de diversas formas, muitas vezes adaptadas aos meios tecnológicos e contextos sociais em que ocorrem. Embora a prática como um todo se baseie no uso, distribuição ou modificação de programas sem autorização legal, ela assume formatos específicos que impactam tanto usuários individuais quanto organizações inteiras.

Um dos tipos mais recorrentes é a falsificação (*counterfeiting*), caracterizada pela reprodução não autorizada de softwares, incluindo suas embalagens, manuais e materiais visuais, de forma a simular os produtos originais com alto grau de fidelidade. Muitas vezes, usuários adquirem cópias falsificadas acreditando serem legítimas, o que agrava o problema. Além do prejuízo à indústria, há ainda o risco adicional de infecções por malwares, já que essas versões podem conter códigos maliciosos embutidos (DAS, 2018).

Outra modalidade significativa é o que se denomina pirataria via internet (*internet theft*). A popularização do acesso à internet facilitou enormemente a obtenção de softwares de maneira ilícita, seja por meio de sites que disponibilizam downloads não autorizados, seja por redes *peer-to-peer* (P2P), nas quais usuários compartilham cópias ilegais entre si. Tais cópias, em geral, são modificadas para burlar os mecanismos de proteção, como verificação de licença ou número de série (DAS, 2018; Symantec, ).

A superutilização em ambientes cliente-servidor (*client-server overuse*) é uma forma de violação comum em redes locais, especialmente em organizações. Nessa prática, o software é instalado em um servidor e acessado simultaneamente por múltiplos usuários, muitas vezes em número superior ao permitido pela licença original. Embora aparentemente inofensiva, essa forma de uso desrespeita os termos contratuais e configura violação de direitos autorais (Symantec, ).

Outro tipo recorrente é o chamado *hard disk loading*, que ocorre geralmente no comércio de hardware. Lojas ou técnicos instalam cópias ilegais de softwares nos computadores vendidos, a fim de torná-los mais atrativos ao consumidor. Nesse caso, não são fornecidos manuais, mídias físicas ou licenças legítimas, o que reforça a ilegalidade da prática. Esse tipo de pirataria é mais comum em relação a sistemas operacionais ((DAS, 2018); SIIA, apud (KHADKA, 2015)).

A pirataria por parte do usuário final (*end user piracy*) abrange diferentes formas de uso indevido por indivíduos. Um dos casos é o aluguel não autorizado de software (*renting*), no qual se permite o uso temporário de programas sem consentimento do detentor dos direitos

autorais. Embora menos comum, essa prática é proibida por legislações específicas, como o *Computer Software Rental Amendments Act* (KHADKA, 2015).

Outro exemplo dentro dessa categoria é o uso indevido de licenças acadêmicas, quando softwares fornecidos para fins educacionais são utilizados para propósitos comerciais ou pessoais. Isso infringe os termos contratuais e pode prejudicar instituições que, muitas vezes, são responsabilizadas pelo uso incorreto (KHADKA, 2015).

A prática conhecida como *softlifting* refere-se à cópia e distribuição não autorizada de software a amigos, familiares ou colegas, sem aquisição de licenças adicionais. Trata-se de uma das formas mais comuns de pirataria no ambiente doméstico e em pequenas empresas, especialmente quando estas aumentam o número de estações de trabalho sem ajustar sua base de licenciamento (DAS, 2018). Muitas vezes, os usuários sequer têm consciência de que estão cometendo uma infração legal.

Essas diversas modalidades ilustram a complexidade do fenômeno da pirataria de software, que vai muito além do simples "download ilegal". Elas revelam falhas na conscientização dos usuários, brechas legais, e também desigualdades no acesso à tecnologia, criando um cenário que exige tanto políticas educativas quanto soluções acessíveis e legais.

### 4.3 Cenário Global e Brasileiro

A pirataria de software é uma prática que persiste em diversas regiões do mundo, especialmente em países em desenvolvimento, onde os custos de softwares licenciados ainda representam uma barreira significativa ao acesso tecnológico. Segundo a pesquisa global de software realizada pela Business Software Alliance (BSA), cerca de 37% dos softwares instalados em computadores pessoais no mundo, em 2018, não estavam devidamente licenciados, representando um prejuízo estimado em US\$ 46,3 bilhões (BSA – THE SOFTWARE ALLIANCE, 2018). Apesar de esse número indicar uma redução em relação aos 42% de 2011 e 43% de 2013, ele ainda evidencia a extensão do problema no cenário global.

No que se refere ao Brasil, os índices de pirataria permanecem significativamente altos, ainda que em trajetória de queda nos últimos anos. De acordo com Feres, Oliveira e Gonçalves (2017), cerca de 50% dos softwares utilizados no país em 2013 eram pirateados, contrastando com os 18% registrados nos Estados Unidos, país com uma economia mais desenvolvida e com maior poder aquisitivo médio por parte dos usuários. Esse dado sugere uma relação direta entre o grau de desenvolvimento econômico de uma nação e o índice de pirataria de software.

Adicionalmente, um levantamento mais recente conduzido pela empresa Revenera indica que, em 2024, o Brasil ocupa a quinta posição no ranking dos países com maior uso de software pirata no mundo (GOFF, 2024). Esse posicionamento demonstra que, apesar dos avanços em conscientização e na disponibilização de alternativas legais, como o código aberto, a pirataria ainda representa uma prática amplamente adotada no país.

A persistência da pirataria no Brasil pode ser parcialmente compreendida à luz de sua história de políticas protecionistas no setor de tecnologia. Durante as décadas de 1980 e 1990, o país adotou uma série de medidas restritivas à importação de hardware e software, com o objetivo de estimular o desenvolvimento de uma indústria nacional. No entanto, tais políticas contribuíram para a normalização de práticas informais e o fortalecimento de um "mercado cinzento", especialmente nos segmentos de computadores domésticos e videogames

(TIETZMANN; PASE; PIOVESAN, 2024). A população, diante da dificuldade de acesso a produtos oficiais, passou a recorrer a estratégias de contorno, como o uso de cópias ilegais e modificações técnicas conhecidas como “gambiaras”, o que moldou uma relação peculiar entre o consumidor brasileiro e o entretenimento digital.

Esse contexto histórico também repercutiu nas relações internacionais do Brasil, particularmente com os Estados Unidos. Conforme aponta o estudo do Social Science Research Council (MIZUKAMI et al., 2011), tensões envolvendo propriedade intelectual marcaram a diplomacia entre os dois países, sobretudo nas discussões sobre a proteção da nascente indústria brasileira de informática e, posteriormente, em relação à adoção de patentes farmacêuticas. Embora o Brasil tenha cedido a pressões internacionais em determinados momentos, como ao instituir direitos autorais sobre softwares e flexibilizar restrições de importação, a cultura de informalidade digital já estava amplamente estabelecida no país.

Portanto, o caso brasileiro exemplifica como fatores históricos, culturais, econômicos e políticos se entrelaçam para explicar a persistência da pirataria, mesmo diante de avanços legislativos e tecnológicos. Essa realidade reforça a importância de se considerar alternativas viáveis, como o uso de software de código aberto, não apenas como uma medida técnica, mas também como uma resposta socioeconômica ao problema.

#### **4.4 Impactos Econômicos e Jurídicos**

A pirataria de software tem efeitos amplos e multidimensionais que afetam desde o setor privado até o poder público, passando também pelo consumidor final. No campo econômico, os prejuízos se manifestam de forma direta e indireta. De acordo com a Associação Brasileira da Propriedade Intelectual (ASSOCIAÇÃO BRASILEIRA DA PROPRIEDADE INTELECTUAL – ABPI, 2023), estima-se que a pirataria gere um impacto negativo anual de R\$ 160 bilhões à economia brasileira, atingindo setores diversos como softwares, músicas, filmes, livros e moda. Os prejuízos diretos envolvem a perda de receitas pela comercialização de cópias ilegais, enquanto os indiretos incluem a desvalorização de marcas, perda de confiança por parte dos consumidores e o aumento dos investimentos necessários em segurança digital (FERREIRA; COSTA, 2019).

As perdas não se restringem ao setor privado. O Estado também é diretamente afetado, sobretudo no que diz respeito à arrecadação fiscal. Como produtos e serviços piratas não estão sujeitos à tributação, a evasão fiscal se torna expressiva. Segundo a mesma fonte (ASSOCIAÇÃO BRASILEIRA DA PROPRIEDADE INTELECTUAL – ABPI, 2023), o Brasil deixa de arrecadar cerca de R\$ 30 bilhões por ano em impostos em razão da pirataria. Esse montante poderia ser revertido em políticas públicas voltadas à educação, infraestrutura e saúde, setores frequentemente prejudicados pela limitação orçamentária.

Apesar desses dados, há controvérsias na forma como esses prejuízos são calculados. Mizukami et al. (2011) argumentam que muitos dos relatórios, como os da BSA (Business Software Alliance), partem do pressuposto de que cada cópia pirata corresponde a uma venda legítima perdida, o que nem sempre se confirma na prática. Os autores destacam que, em muitos casos, os usuários que recorrem ao software pirata não o fariam caso fosse necessário pagar pela licença, especialmente em países em desenvolvimento, onde os custos são proibitivos (Tigre e Marques, 2009, apud (FERES; OLIVEIRA; GONÇALVES, 2017)).

Outro aspecto relevante diz respeito à segurança cibernética. O uso de softwares não licenciados está frequentemente associado a riscos tecnológicos e operacionais. De acordo com pesquisa conduzida pela BSA (BSA – THE SOFTWARE ALLIANCE, 2018), há uma chance de uma em três de que softwares não oficiais venham acompanhados de malwares. Cada incidente pode custar, em média, US\$ 2,4 milhões para ser resolvido, além de comprometer dados sensíveis e a reputação das empresas. O custo global com ataques oriundos de softwares não licenciados chega a cerca de US\$ 359 bilhões anuais. Esses riscos têm motivado muitos gestores de TI a priorizarem a regularização dos sistemas utilizados como forma de mitigar perdas associadas à segurança (BSA – THE SOFTWARE ALLIANCE, 2018).

No âmbito do consumidor individual, os danos também são consideráveis. Como aponta Alves (2018), ao utilizar versões *crackeadas* de softwares, os usuários se expõem a malwares que podem comprometer seus sistemas operacionais, causar perda de dados e até mesmo possibilitar sequestros digitais por meio de *ransomware*. Esse risco é especialmente preocupante diante do fato de que muitos recorrem à pirataria como estratégia de inclusão digital e aprendizado, dada a necessidade de familiaridade com determinadas ferramentas exigidas pelo mercado de trabalho.

Do ponto de vista jurídico, a legislação brasileira trata a pirataria como infração penal. A Lei nº 9.609, de 19 de fevereiro de 1998, prevê punições nos artigos 12 a 14 para crimes relacionados a cópias ilegais de software. Além disso, o Superior Tribunal de Justiça, por meio da Súmula 502, reforça que, mesmo em casos de aceitação social da prática, sua repressão deve ser aplicada, considerando-se a materialidade e autoria como elementos suficientes para tipificação do crime (SALDANHA, 2024).

Por fim, vale observar que, diferentemente de jurisdições como a norte-americana, o ordenamento jurídico brasileiro não adota a doutrina do *fair use*, que permite o uso parcial de obras protegidas sob certas condições. No Brasil, as exceções ao direito autoral são limitadas e mais restritivas, conforme disposto na Lei nº 9.610, de 19 de fevereiro de 1998, o que reduz as possibilidades de defesa legal para usuários que utilizam softwares de forma não autorizada, mesmo que para fins educacionais ou de acesso à informação (ALVES, 2024).

## 4.5 Dimensões Sociais e Culturais

Embora a pirataria de software seja caracterizada juridicamente como uma infração, diversas análises indicam que, em determinadas realidades sociais, ela pode adquirir contornos de aceitação cultural. Saldanha (2024) argumenta que a alta prevalência de práticas de consumo ilegítimo, mesmo diante de políticas de repressão e da condenação pelas indústrias criativas, sugere um nível considerável de tolerância social. Tal aceitação parece decorrer de múltiplos fatores: para alguns usuários, o principal atrativo reside nos preços significativamente mais baixos em relação aos praticados no mercado formal; para outros, trata-se da única forma de acesso a conteúdos ou programas indisponíveis por vias legais. Há ainda quem veja na pirataria uma ferramenta de inclusão, especialmente entre parcelas da população que, de outro modo, estariam totalmente excluídas do universo digital e cultural.

Essa percepção também se manifesta em discursos que relativizam os danos sociais da pirataria. Cruz (), em estudo com moradores de regiões de menor poder aquisitivo, aponta que os entrevistados tendem a justificar o consumo de software e jogos piratas com base na alta carga tributária dos produtos oficiais e na desconfiança quanto à real destinação desses tributos. Além disso, há a crença de que o impacto financeiro causado às empresas detentoras dos direitos não

compromete sua lucratividade de forma significativa, especialmente quando se trata de grandes corporações internacionais. Entre os consumidores, também é comum o argumento de que a produção de mídias originais, muitas vezes, já causa danos ambientais, não sendo a pirataria o único vetor de prejuízo socioambiental.

A cultura do "jeitinho brasileiro" é outro elemento frequentemente associado à banalização da pirataria. Casali et al. (2010) destacam que muitos indivíduos apresentam predisposição ao uso de software não licenciado mesmo reconhecendo sua ilegitimidade, o que revela uma dissociação entre atitude ética e intenção de uso. Essa percepção é reforçada pelos achados de Grohmann et al. (2015), que, ao analisarem o comportamento de jovens brasileiros, identificaram que a decisão de piratear não necessariamente está vinculada à avaliação moral da prática, mas sim a fatores como sensação de impunidade, contexto econômico e normalização cultural do ato.

Nesse cenário, a decisão de piratear software transcende uma simples análise econômica, sendo profundamente influenciada por percepções de justiça e por um complexo cálculo de custo-benefício. Estudos como o de Fine et al. (2019), propõem um modelo de escolha do consumidor que integra múltiplas teorias para explicar esse comportamento. Segundo os autores, a intenção de piratear é fortemente determinada pelas atitudes e percepções éticas do indivíduo sobre o ato. Contudo, essa avaliação moral é ponderada por uma análise racional dos riscos percebidos — como a possibilidade de punição ou a infecção por malware — e dos benefícios tangíveis, como o valor de mercado, a qualidade e a facilidade de acesso ao software por vias não oficiais (FINE et al., 2019). Dessa forma, a pirataria emerge não apenas como uma resposta a preços considerados injustos, mas como uma decisão calculada onde a utilidade e os resultados esperados podem superar as barreiras morais e os riscos associados, especialmente em contextos culturais com menor aversão à incerteza.

Tais práticas e justificativas não surgem isoladamente. Conforme observa Almeida (2007), existe uma correlação entre os altos índices de corrupção percebida no Brasil e a internalização do "jeitinho" como traço cultural. Esse fenômeno acaba por legitimar pequenas transgressões cotidianas, como o uso de software pirata, que se tornam toleradas — e até mesmo normalizadas — no imaginário coletivo.

Em uma perspectiva mais simbólica, a figura do "pirata" também exerce influência sobre a forma como essas práticas são percebidas socialmente. De acordo com Konstam (2008), o pirata ocupa um lugar ambíguo no imaginário ocidental: simultaneamente condenado pela ordem legal e celebrado como ícone de rebeldia e resistência à autoridade. Esse imaginário contribui para moldar uma visão cultural da pirataria que vai além de sua definição legal, apresentando-a como uma forma de transgressão socialmente aceitável ou até romantizada. Dessa forma, os estigmas culturais em torno da pirataria de software não apenas explicam sua persistência, como também revelam a complexidade do fenômeno, que deve ser analisado à luz das dinâmicas sociais, econômicas e simbólicas que moldam o comportamento dos usuários.

## 5 O ECOSISTEMA DE CÓDIGO ABERTO

### 5.1 Definições: Software Livre e Código Aberto

A compreensão conceitual das expressões software livre e código aberto é essencial para este estudo, uma vez que ambas sustentam propostas alternativas ao modelo proprietário de distribuição de software. Embora frequentemente utilizadas como sinônimos, as duas expressões possuem origens, motivações e nuances distintas.

O termo software livre é historicamente associado ao movimento liderado por Richard Stallman e à Free Software Foundation (FSF), criado na década de 1980. Segundo esse entendimento, um programa só pode ser considerado livre quando respeita integralmente as liberdades fundamentais do usuário: a liberdade de executar o programa para qualquer propósito (liberdade 0); de estudar seu funcionamento e adaptá-lo conforme necessário (liberdade 1), o que exige o acesso ao código-fonte; de redistribuir cópias do software (liberdade 2); e de aprimorar o programa e compartilhar essas melhorias com a comunidade (liberdade 3) (STALLMAN; Free Software Foundation, 2015).

Dessa forma, o software livre não se refere apenas à gratuidade, mas à garantia de controle, transparência e colaboração no uso e no desenvolvimento das ferramentas digitais. Trata-se, portanto, de uma proposta ética, centrada na liberdade dos usuários e na promoção de um ambiente de compartilhamento e cooperação tecnológica.

Por outro lado, a noção de código aberto (ou *open source*) passou a ganhar destaque a partir de 1998, quando foi formulada a chamada Open Source Definition (OSD) pela Open Source Initiative (OSI). Esse documento, inspirado nas diretrizes do projeto Debian (Debian Free Software Guidelines), estabelece um conjunto de dez critérios que devem ser atendidos para que um software seja considerado de código aberto (COLEMAN; HILL, 2004).

Embora compartilhe com o software livre muitos princípios técnicos, como o acesso ao código-fonte e a possibilidade de modificação e redistribuição, o movimento open source adota uma abordagem mais pragmática e voltada ao mercado. Segundo análise de Sudibjo (2017), a popularização do termo "código aberto" representou uma reinterpretação estratégica dos ideais do software livre, buscando maior aceitação por parte do setor corporativo. Nessa perspectiva, as liberdades técnicas continuaram presentes, mas passaram a ser justificadas principalmente por seus benefícios práticos, como o aumento da eficiência, a redução de custos e a aceleração da inovação por meio da colaboração descentralizada.

Assim, autores como Rossi e Bochnia (2023) destacam que o open source se apresenta menos como um movimento ético e mais como uma proposta de modelo de desenvolvimento e distribuição viável economicamente. Essa transformação contribuiu para que empresas tradicionalmente associadas ao software proprietário passassem a utilizar, desenvolver e distribuir softwares com base em licenças abertas.

Dessa forma, embora haja sobreposição nos aspectos técnicos entre o software livre e o código aberto, suas motivações e formas de articulação social e política divergem significativamente. Enquanto o primeiro está alicerçado em princípios filosóficos e de liberdade, o segundo enfatiza a eficiência técnica e o potencial de mercado do desenvolvimento colaborativo.



## 5.2 Histórico e Evolução

O surgimento do movimento Open Source está diretamente relacionado à história do Software Livre, embora seus fundamentos filosóficos e objetivos estratégicos apresentem importantes distinções. Para compreender essa evolução, é necessário contextualizar os dois movimentos e as razões que levaram à sua separação conceitual.

O movimento do Software Livre surgiu na década de 1980, tendo como principal articulador Richard Stallman, programador e ativista que fundou a Free Software Foundation (FSF) em 1985. De acordo com Aguiar (2009), esse movimento possui um caráter essencialmente ético e político, defendendo que os usuários devem ter liberdade para executar, copiar, distribuir, estudar, modificar e aprimorar o software. O uso de programas proprietários, sob essa perspectiva, é visto como um entrave à liberdade individual e um problema de ordem social.

A partir dessa visão, Stallman argumenta que a principal motivação por trás do desenvolvimento de software não deveria ser a obtenção de lucro, mas sim o compartilhamento de conhecimento e o fortalecimento da comunidade (PEIRANO; UGARTE, 2004). Dessa forma, o Software Livre foi estabelecido como um movimento social, cujos princípios estão mais ligados à filosofia hacker de colaboração, transparência e liberdade do que a estratégias de mercado.

Contudo, em 1998, parte da comunidade envolvida com o desenvolvimento colaborativo de software percebeu a necessidade de reformular a imagem do Software Livre, especialmente para torná-lo mais atrativo ao setor empresarial. Essa iniciativa deu origem ao termo “Open Source”, promovido por figuras como Eric Raymond e Bruce Perens. Segundo Aguiar (2009), a mudança não foi técnica, mas sim de posicionamento: os proponentes do Open Source optaram por enfatizar os benefícios práticos e econômicos do desenvolvimento aberto — como a qualidade do código, a confiabilidade, a escalabilidade e a redução de custos — ao invés de basear seus argumentos em questões éticas.

Dessa forma, o movimento Open Source estabeleceu-se como uma vertente mais pragmática, com foco na eficiência do modelo colaborativo e na adoção empresarial. A criação da Open Source Initiative (OSI), também em 1998, consolidou esse novo direcionamento, promovendo uma definição mais neutra de software aberto, baseada em critérios técnicos e licenças compatíveis.

Portanto, embora os dois movimentos compartilhem práticas semelhantes — como o acesso ao código-fonte e a liberdade de modificação —, eles divergem em seus pressupostos fundamentais. Enquanto o Software Livre se ancora em valores morais e sociais, o Open Source busca legitimar a abertura do código a partir de sua eficácia técnica e utilidade econômica no contexto corporativo.

## 5.3 Licenças: As Regras do Jogo

O licenciamento é um dos pilares fundamentais para a estrutura legal e filosófica do software livre, pois define os termos sob os quais um programa pode ser utilizado, modificado e redistribuído. Existem diferentes categorias de licenças dentro do universo do software livre, sendo as mais relevantes aquelas que se dividem entre modelos recíprocos (ou copyleft) e permissivos. Cada modelo impõe condições distintas sobre como o software e suas versões derivadas podem ser reutilizados por terceiros.

As chamadas licenças recíprocas — também conhecidas como copyleft — estabelecem que qualquer trabalho derivado de um software licenciado sob esse modelo deve necessariamente ser redistribuído sob os mesmos termos da licença original. Conforme exposto pela Free Software Foundation (FSF), a filosofia por trás desse tipo de licenciamento é assegurar que as liberdades concedidas pelo software original sejam preservadas nas versões modificadas, impedindo que futuras adaptações imponham restrições adicionais ao uso ou distribuição (SABINO, 2011).

Essa abordagem visa fortalecer o ecossistema do software livre como um todo, promovendo um ciclo de colaboração contínuo, onde as melhorias são mantidas acessíveis à comunidade. Espera-se que, com isso, a base de software livre se expanda progressivamente, beneficiando tanto usuários quanto desenvolvedores. Além disso, a obrigatoriedade de manter os mesmos termos de licenciamento pode favorecer a compatibilidade entre diferentes versões de um sistema, ao evitar a fragmentação ocasionada por funcionalidades fechadas e isoladas (SABINO, 2011).

No entanto, esse modelo não é isento de críticas. Há debates internos na própria comunidade de software livre quanto ao grau de rigidez do copyleft, uma vez que ele pode dificultar a integração com projetos que utilizam licenças menos restritivas. Na prática, é possível incorporar software de licenças permissivas dentro de projetos copyleft — mas o oposto não é permitido, o que pode limitar a interoperabilidade entre projetos que seguem filosofias distintas (SABINO, 2011).

A principal licença que representa o modelo copyleft é a GNU General Public License (GPL), desenvolvida pela Free Software Foundation. Desde sua criação, essa licença passou por diversas versões, sendo as mais relevantes a GPL versão 2 (GPLv2), a GPL versão 3 (GPLv3) e a Affero General Public License (AGPL), cada uma com aprimoramentos e adaptações legais para acompanhar os desafios contemporâneos relacionados à distribuição e uso de software em ambientes diversos, como redes e serviços online.

Em contraste com o modelo copyleft, encontram-se as licenças permissivas, que oferecem maior flexibilidade para os desenvolvedores. Nesse tipo de licença, o código-fonte pode ser reutilizado e redistribuído sob diferentes termos, inclusive sob licenças proprietárias, desde que sejam mantidos os créditos de autoria e o aviso de direitos autorais original. A licença MIT/X11, por exemplo, é amplamente utilizada e considerada compatível com a GPL. Segundo a Free Software Foundation, essa licença é uma alternativa viável para quem deseja maior liberdade na distribuição e incorporação de seu software, sem impor as obrigações recíprocas características do copyleft (SABINO, 2011).

Outra licença amplamente adotada é a Apache License 2.0, que, apesar de também ser considerada uma licença de software livre, apresenta cláusulas que a tornavam inicialmente incompatível com a GPL versão 2. Entre essas cláusulas destacam-se: a possibilidade de encerramento automático da licença em caso de litígios relacionados a patentes e a obrigatoriedade de indenização aos colaboradores caso o software seja distribuído com garantias legais. Tais exigências não constavam nas versões anteriores da GPL, o que impedia a redistribuição do software Apache sob os termos da GPLv2. Contudo, a GPLv3 foi reformulada de modo a tornar-se compatível com essas cláusulas, permitindo, assim, uma integração mais ampla entre diferentes projetos de software livre (SABINO, 2011).

O estudo das licenças demonstra que o licenciamento não é apenas uma questão técnica ou legal, mas também filosófica. Ele reflete visões distintas sobre liberdade, controle, colaboração e sustentabilidade no desenvolvimento de tecnologias abertas. A escolha da licença adequada

depende não apenas dos objetivos técnicos do projeto, mas também da postura que se deseja adotar em relação à comunidade e ao ecossistema do software livre como um todo.

## **5.4 Análise Comparativa: Vantagens e Desvantagens**

O debate entre software livre e software proprietário tem sido objeto de diversas análises, especialmente no que se refere às vantagens e limitações percebidas por usuários, técnicos e especialistas da área de tecnologia da informação. De acordo com Garcia et al. (2010), os aspectos mais frequentemente apontados como positivos no software livre incluem o baixo custo de aquisição, a possibilidade de customização, a liberdade de uso e modificação, e a segurança associada à transparência do código-fonte. Por outro lado, também são relatadas desvantagens relacionadas à usabilidade, à carência de suporte técnico estruturado e à disponibilidade limitada de determinados aplicativos.

### **5.4.1 Vantagens do Software Livre**

O aspecto econômico é frequentemente citado como uma das principais motivações para a adoção do software livre. Segundo Gay (2007), a gratuidade de muitos desses sistemas representa uma alternativa viável em contextos onde os altos custos de licenciamento de software proprietário são proibitivos. Cerioni (2003) complementa essa perspectiva ao destacar que o software livre permite o reaproveitamento de equipamentos considerados obsoletos, prolongando sua vida útil e ampliando o acesso à tecnologia. Didio (2005), por sua vez, argumenta que essa economia se traduz em benefícios financeiros mais amplos, especialmente em ambientes corporativos e governamentais.

Outro ponto relevante é a flexibilidade e possibilidade de customização. Para Gay (2007), a abertura do código possibilita a adaptação dos programas conforme as necessidades específicas dos usuários, o que representa um diferencial importante em relação ao modelo fechado e padronizado dos softwares proprietários. Serrano, Guerreiro e Caldeira (2004) também apontam essa flexibilidade como um atributo essencial para organizações que buscam soluções tecnológicas sob medida.

A liberdade de ação proporcionada pelo software livre é considerada uma vantagem significativa. Essa liberdade refere-se tanto ao acesso ao código-fonte (SALEH, 2004; GAY, 2007), quanto à autonomia frente a fornecedores, reduzindo a dependência de empresas detentoras dos direitos autorais do software (SILVEIRA, 2005). Para além do aspecto técnico, essa autonomia também é interpretada como um valor social, uma vez que promove a soberania tecnológica e a inclusão digital (SILVEIRA, 2005).

No que se refere à segurança, Hexsel (2002) defende que os softwares livres tendem a apresentar menor vulnerabilidade a vírus e invasões, justamente pela possibilidade de auditoria constante e coletiva do código. Esse argumento encontra respaldo na conhecida "Lei de Linus", segundo a qual "com olhos suficientes, todos os bugs são superficiais", reforçando a ideia de que a abertura do código contribui para uma identificação mais rápida de falhas e vulnerabilidades.

Além disso, o software livre é frequentemente associado à colaboração e ao desenvolvimento coletivo. Castells (1999) e Hexsel (2002) destacam a existência de comunidades ativas que contribuem para o aprimoramento contínuo dos sistemas, promovendo a cooperação entre

usuários e desenvolvedores e beneficiando a sociedade como um todo por meio da partilha de conhecimento.

#### **5.4.2 Desvantagens do Software Livre**

Apesar das inúmeras vantagens apontadas, o software livre também apresenta limitações que dificultam sua adoção em determinados contextos. Hexsel (2002) observa que a ausência de suporte técnico estruturado pode representar um entrave para usuários iniciantes, especialmente em ambientes corporativos que demandam atendimento rápido e eficaz. Ainda segundo o autor, a adoção de software livre exige maior capacitação técnica por parte dos usuários, o que pode gerar barreiras em organizações com equipes pouco familiarizadas com esses sistemas.

Campos (2006) também destaca a escassez de softwares voltados ao entretenimento e a jogos como uma limitação significativa, particularmente em contextos domésticos ou educacionais. Adicionalmente, a documentação técnica nem sempre é acessível ou suficientemente detalhada para auxiliar na resolução de problemas mais complexos, conforme apontado por Didio (2005).

No campo da segurança, embora existam argumentos favoráveis ao software livre, Serrano, Guerreiro e Caldeira (2004) sugerem que o código fechado, característico do software proprietário, também pode representar uma vantagem, na medida em que restringe o acesso a informações sensíveis e potencialmente exploráveis.

#### **5.4.3 Considerações sobre o Software Proprietário**

O modelo proprietário, por sua vez, também apresenta pontos positivos frequentemente associados à facilidade de uso, compatibilidade e suporte técnico. Cerioni (2003) e Cardozo (1965) ressaltam que esses sistemas costumam ser mais amigáveis ao usuário final, com processos de instalação e configuração simplificados. Campos (2006) reforça que a ampla compatibilidade com aplicativos e equipamentos é uma das razões pelas quais o software proprietário mantém forte presença no mercado.

No entanto, esse modelo implica uma série de restrições que, segundo Hexsel (2002), vão desde a dependência em relação ao fornecedor até a impossibilidade de modificação do código. Cerioni (2003) também menciona a escassez de aplicativos personalizados ou especializados em algumas áreas. Essas limitações podem restringir a liberdade dos usuários e dificultar a adaptação de soluções tecnológicas a contextos específicos.

### **5.5 Modelos de Negócio Sustentáveis**

Embora o software de código aberto esteja frequentemente associado à gratuidade, seu desenvolvimento pode ser parte integrante de estratégias comerciais bem definidas. Empresas têm adotado diferentes abordagens para monetizar soluções baseadas em OSS, conciliando a abertura do código com objetivos financeiros e sustentação econômica.

Um dos modelos mais conhecidos é o licenciamento duplo (dual licensing), no qual uma mesma aplicação é disponibilizada sob duas licenças distintas: uma de caráter copyleft, como

a GNU General Public License (GPL), e outra proprietária, direcionada a clientes que desejam utilizar o software em contextos comerciais mais amplos. Conforme explica Riehle, a licença copyleft exige que quaisquer modificações ou integrações ao software sejam redistribuídas sob os mesmos termos da licença original. Isso cria uma restrição para empresas que pretendem incorporar o OSS em soluções comerciais fechadas, o que, por sua vez, as leva a optar pela versão licenciada comercialmente. Um exemplo notável dessa estratégia é o banco de dados MySQL, amplamente citado como um caso de sucesso no uso do modelo de licenciamento duplo.

Outro modelo relevante é o chamado modelo de núcleo aberto (open core). Nesse formato, uma parte central do software é disponibilizada sob uma licença de código aberto, enquanto funcionalidades adicionais, geralmente voltadas para uso corporativo ou recursos mais avançados, são licenciadas de forma proprietária. De acordo com Riehle, tais componentes pagos incluem, com frequência, ferramentas indispensáveis à aplicação empresarial, como integrações especializadas, módulos de segurança aprimorados ou interfaces de administração avançadas.

Esses modelos revelam que o código aberto não representa, necessariamente, uma ruptura com os princípios de mercado, mas sim uma adaptação estratégica. A abertura do código pode coexistir com práticas comerciais consolidadas, possibilitando tanto a sustentabilidade econômica do projeto quanto o acesso mais amplo à tecnologia. Ao integrar elementos gratuitos e pagos, tais modelos oferecem uma alternativa híbrida entre o software proprietário tradicional e o software livre integral, equilibrando liberdade de uso com incentivos à inovação e ao suporte corporativo.

## **6 Procedimentos Metodológicos**

### **6.1 Abordagem da Pesquisa**

Este trabalho adotou uma abordagem exploratória, com métodos mistos (quantitativos e qualitativos), visando compreender o fenômeno da pirataria de software e a viabilidade de alternativas de código aberto como solução legítima de acesso à tecnologia. A estratégia metodológica foi dividida em duas etapas complementares: a identificação da demanda por pirataria de software e a realização de testes comparativos de desempenho entre softwares proprietários amplamente pirateados e suas respectivas alternativas livres.

### **6.2 Etapa 1: Identificação da Demanda por Pirataria de Software**

A primeira etapa consistiu em mapear os softwares mais buscados em contextos de pirataria. Inicialmente, foi coletada uma amostra do site *The Pirate Bay*, especificamente a seção “Top 100” da categoria “Applications”, no dia 5 de junho de 2025. Após a remoção de duplicidades e variantes de versão, bem como a exclusão de softwares classificados como livres, foi elaborada uma lista consolidada de softwares proprietários.

Posteriormente, utilizou-se a ferramenta Google Trends para aferir o interesse por pirataria digital por meio de buscas contendo o nome do software seguido do termo “crack” (ex.: “Photoshop crack”). As buscas consideraram os últimos 12 meses em escala global. Os resultados foram normalizados em uma escala de 0 a 100, permitindo calcular uma média de interesse e compor um ranking com os softwares com maior associação à pirataria.

### 6.3 Etapa 2: Análise Comparativa de Desempenho

Na segunda etapa, foram realizados testes práticos para comparar o desempenho de softwares proprietários amplamente pirateados e suas contrapartes de código aberto. Os critérios avaliados incluíram:

- Tempo de inicialização (em segundos);
- Pico de uso de memória RAM (em MB);
- Pico de uso de CPU (em %);

Os testes foram realizados na mesma máquina (*Dell G15 5510* com Windows 11 Home Single Language, Intel i5-10500H, 12 GB RAM, SSD NVMe, NVIDIA GTX 1650), com controle de variáveis externas. Cada cenário foi repetido três vezes para cálculo de média, desvio padrão e coeficiente de variação (CV).

Como métrica principal de estabilidade, utilizou-se o CV com os seguintes critérios:

- $CV \leq 10\%$ : Variação excelente;
- $10\% < CV \leq 20\%$ : Variação aceitável;
- $CV > 20\%$ : Variação potencialmente inaceitável.

Para valores absolutos baixos (ex.: tempos de inicialização  $< 10$  s), também se considerou a variação absoluta como critério de aceitação ( $\leq 1$  s para tempo,  $\leq 200$  MB para RAM,  $\leq 5\%$  para CPU).

## 7 Resultados e Discussão

Com base nas métricas do Google Trends e nos dados extraídos do The Pirate Bay, foram identificados os softwares com maior associação a práticas de pirataria.

### 7.1 Softwares com maior demanda por pirataria

A Tabela 1 apresenta os softwares com maior demanda, indicando uma forte associação da pirataria com ferramentas profissionais de criação, escritório e sistemas operacionais. Isso sugere uma carência por acesso a soluções produtivas de alto custo.

### 7.2 Comparações de desempenho por segmento

A seguir, apresenta-se uma síntese comparativa dos testes realizados entre os softwares proprietários listados e suas alternativas livres, cujos resultados detalhados estão nas figuras a seguir.

Tabela 1: Softwares com maior demanda por pirataria.

<b>Software</b>	<b>Ocorrências</b>
Premiere	87.98
FL Studio	85.81
After Effects	83.20
Photoshop	82.83
Office	82.30
Acrobat	81.81
Ilustrator	81.50
AutoCAD	79.67
Windows 11	79.37
Windows 7	79.07
Lightroom	79.00

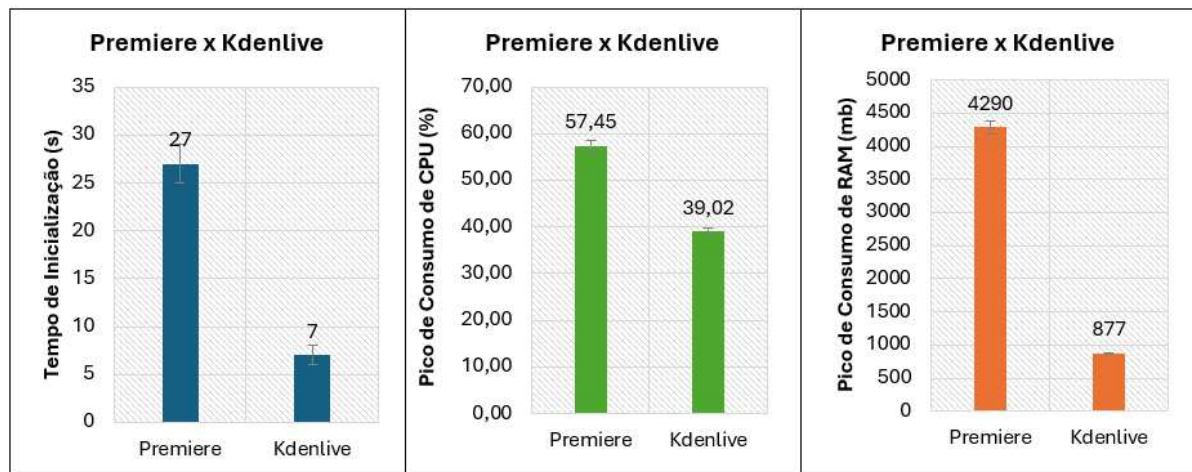
*Fonte: Elaboração própria (2025).*

### 7.2.1 Premiere vs Kdenlive (Edição de vídeo profissional)

A tarefa consistiu no corte e renderização dos 5 minutos iniciais de um vídeo de 15 minutos. Conforme demonstrado na Figura 1, Kdenlive apresentou menor consumo de RAM e um tempo de inicialização consideravelmente menor (7 s em média, contra 27 s do Premiere). Ambos os softwares tiveram desempenho estável, com Coeficiente de Variação (CV) igual ou inferior a 14% ( $CV \leq 14\%$ ).

Figura 1: Resultados dos testes comparativos entre Premiere e Kdenlive.

Premiere	Tempo (S)	RAM(MB)	CPU (%)	Kdenlive	Tempo (S)	RAM(MB)	CPU (%)
Teste 1	25	4388	56,34	Teste 1	6	880	38,59
Teste 2	29	4286	58,22	Teste 2	8	882	38,68
Teste 3	27	4196	57,79	Teste 3	7	869	39,79
Média	27	4290	57,45	Média	7	877	39,02
D.Padrão	2	96,06	0,98	D.Padrão	1	7	0,66
CV	7%	2%	2%	CV	14%	1%	2%



Fonte: Elaboração própria (2025).

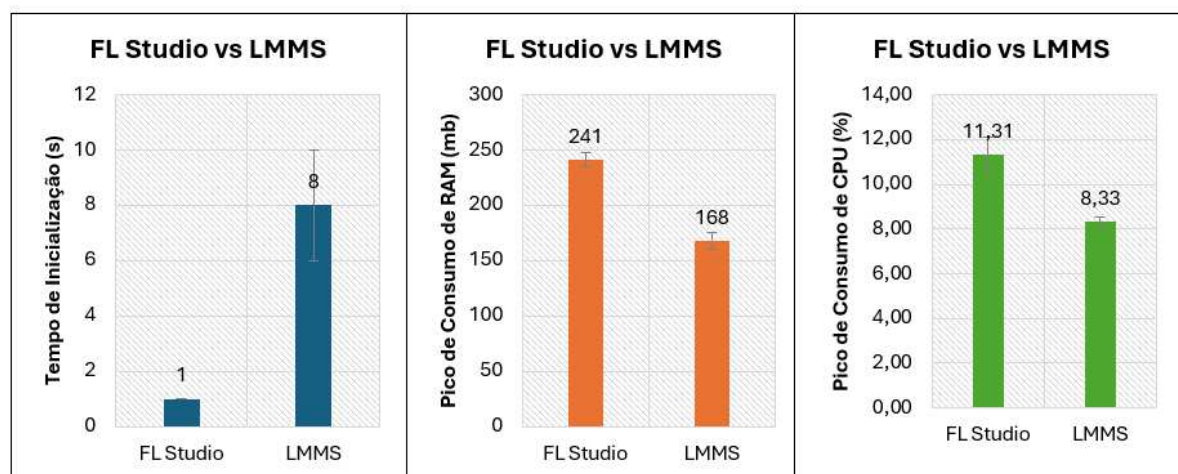
## 7.2.2 FL Studio vs LMMS (Produção musical)

A tarefa foi a criação de uma batida com 5 toques iguais em um loop simples. Nos testes apresentados na Figura 2, FL Studio inicializou mais rápido (1 s vs 8 s), mas LMMS teve menor consumo de RAM. As variações de desempenho foram consideradas aceitáveis, com exceção da instabilidade no tempo de inicialização do LMMS (CV = 25%).



Figura 2: Resultados dos testes comparativos entre FL Studio e LMMS.

FL Studio	Tempo (S)	RAM(MB)	CPU (%)	LMMS	Tempo (S)	RAM(MB)	CPU (%)
Teste 1	1	243	11,38	Teste 1	8	170	8,47
Teste 2	1	246	12,05	Teste 2	10	174	8,45
Teste 3	1	234	10,50	Teste 3	6	160	8,07
Média	1	241	11,31	Média	8	168	8,33
D.Padrão	0	6,2	0,8	D.Padrão	2,00	7,21	0,23
CV	0%	3%	7%	CV	25%	4%	3%



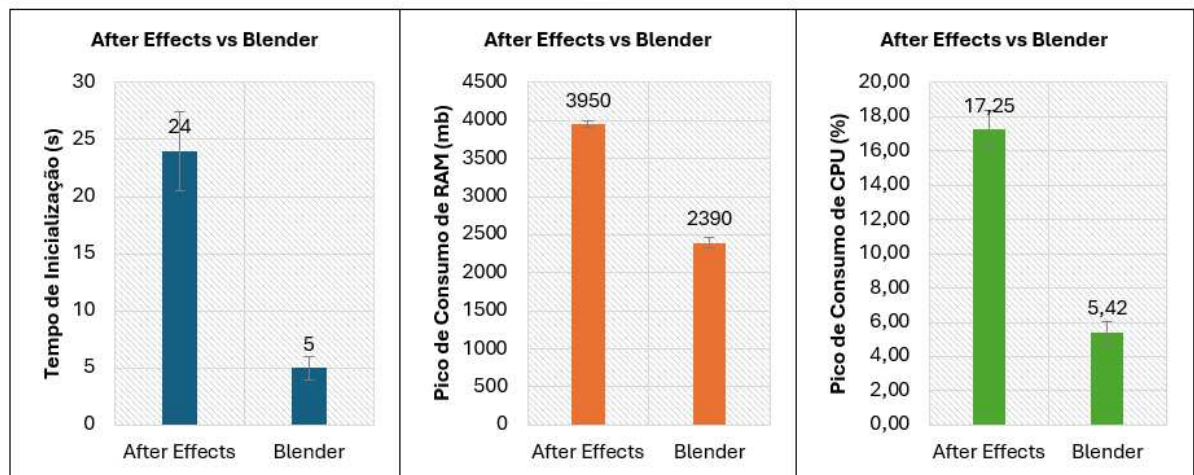
Fonte: Elaboração própria (2025).

### 7.2.3 After Effects vs Blender (Composição e modelagem 3D)

A tarefa foi a criação de uma animação de 3 segundos com um círculo movendo-se em 90 graus. A Figura 3 mostra que o Blender apresentou menor uso de CPU e RAM, porém com uma inicialização mais lenta (5 s vs 24 s). Os coeficientes de variação estiveram dentro dos limites aceitáveis, exceto para o tempo do Blender (CV = 20%).

Figura 3: Resultados dos testes comparativos entre After Effects e Blender.

After Effects	Tempo (S)	RAM(MB)	CPU (%)	Blender	Tempo (S)	RAM(MB)	CPU (%)
Teste 1	22	3995	18,38	Teste 1	5	2314	6,03
Teste 2	22	3911	17,15	Teste 2	6	2422	5,49
Teste 3	28	3944	16,22	Teste 3	4	2434	4,74
Média	24	3950	17,25	Média	5	2390	5,42
D.Padrão	3,46	42,3	1,1	D.Padrão	1,00	66,09	0,65
CV	14%	1%	6%	CV	20%	3%	12%

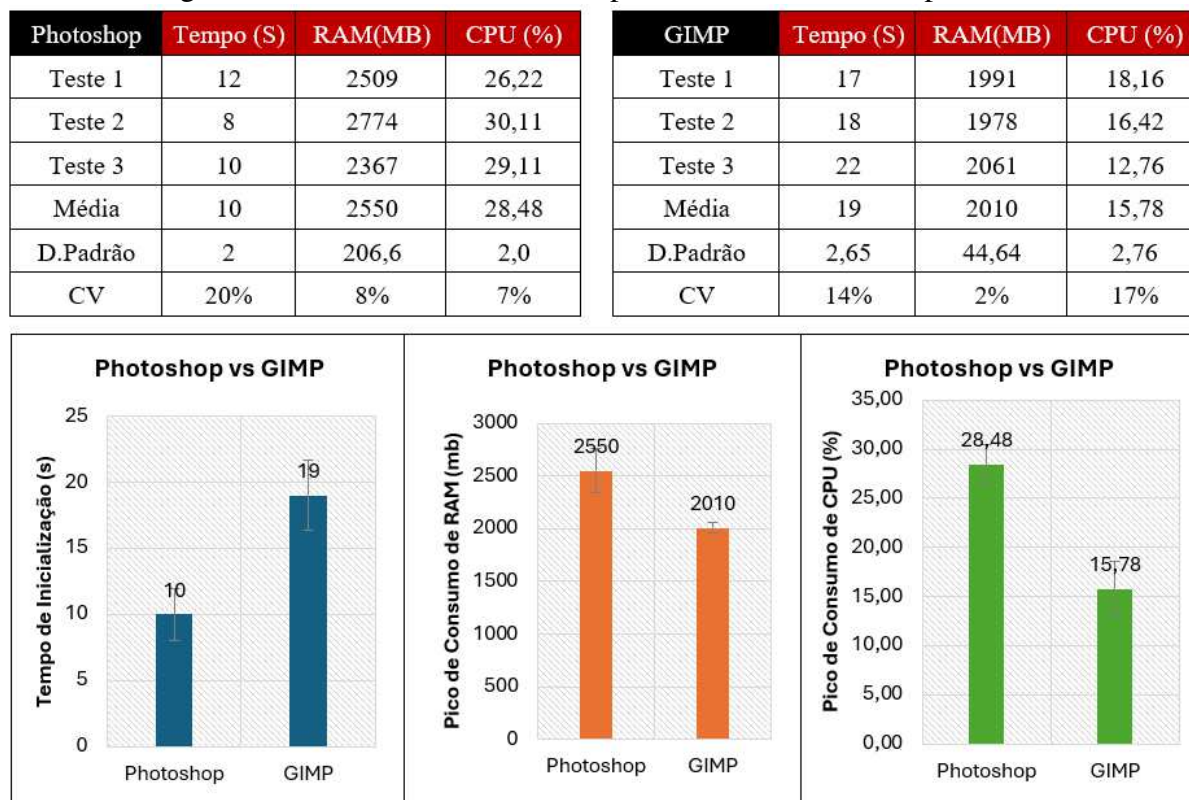


Fonte: Elaboração própria (2025).

#### 7.2.4 Photoshop vs GIMP (Edição de imagem)

A tarefa consistiu no corte de  $\frac{1}{4}$  de uma foto e na aplicação de um filtro sépia. De acordo com os dados da Figura 4, o GIMP utilizou menos memória RAM, mas seu tempo de inicialização foi superior ao do Photoshop (19 s vs 10 s). Ambos os softwares exibiram um desempenho consistente nos testes.

Figura 4: Resultados dos testes comparativos entre Photoshop e GIMP.



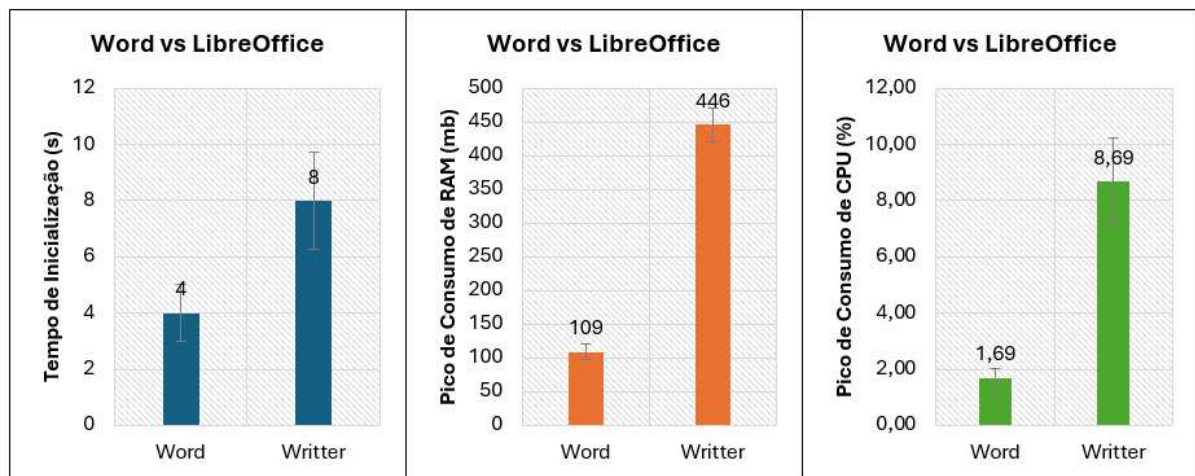
Fonte: Elaboração própria (2025).

### 7.2.5 Word vs Writer (Pacote de produtividade)

A tarefa envolveu a inserção de 10 parágrafos de "Lorem Ipsum", a aplicação de negrito nos parágrafos ímpares e a impressão do documento. O Writer teve maior uso de recursos, mas com um tempo de inicialização aceitável, como pode ser visto na Figura 5. Embora o CV para o tempo de inicialização do Word tenha sido alto (25%), a variação absoluta foi de apenas 1 segundo.

Figura 5: Resultados dos testes comparativos entre Word e LibreOffice Writer.

Word	Tempo (S)	RAM(MB)	CPU (%)	Writer	Tempo (S)	RAM(MB)	CPU (%)
Teste 1	4	111	1,68	Teste 1	9	474	10,01
Teste 2	5	120	2,02	Teste 2	9	438	9,04
Teste 3	3	96	1,37	Teste 3	6	426	7,02
Média	4	109	1,69	Média	8	446	8,69
D.Padrão	1	12,1	0,3	D.Padrão	1,73	24,98	1,53
CV	25%	11%	19%	CV	22%	6%	18%



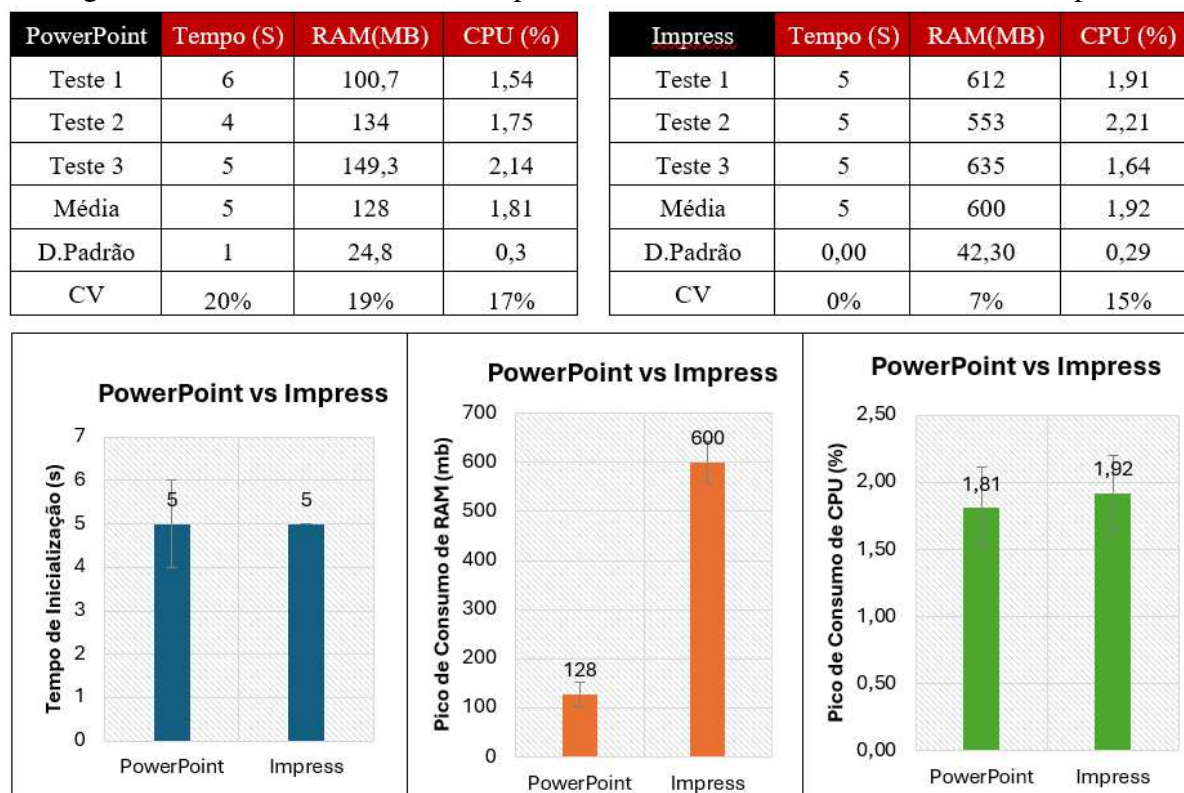
Fonte: Elaboração própria (2025).

## 7.2.6 PowerPoint vs Impress (Pacote de produtividade)

A tarefa foi a criação de 7 slides: o primeiro e o último com títulos, e os demais com imagens e parágrafos. Os resultados na Figura 6 indicam que o Impress apresentou um consumo de RAM mais elevado, mas seu tempo de resposta foi similar ao do PowerPoint (ambos com média de 5 s). Todas as variações de desempenho observadas ficaram dentro dos limites aceitáveis.



Figura 6: Resultados dos testes comparativos entre PowerPoint e LibreOffice Impress.



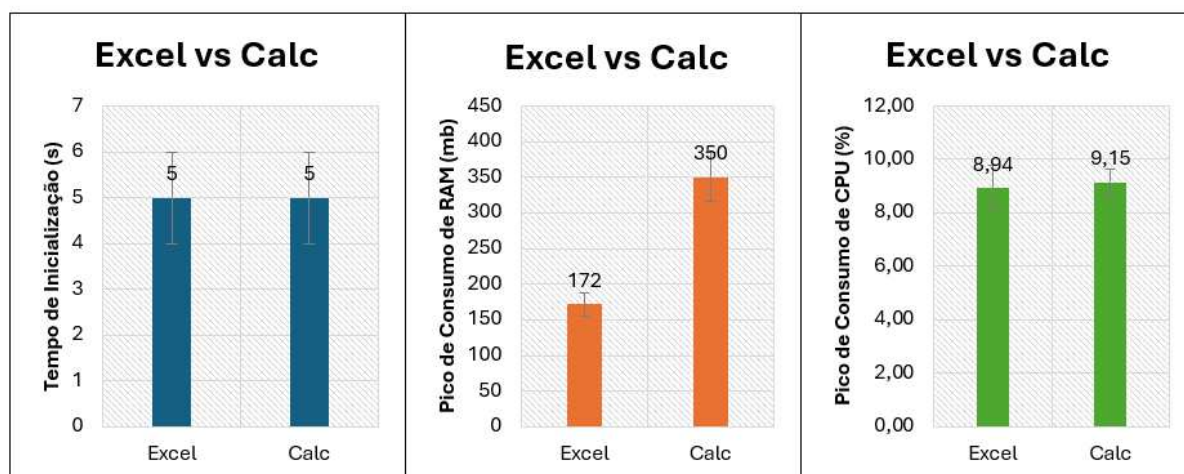
Fonte: Elaboração própria (2025).

## 7.2.7 Excel vs Calc (Pacote de produtividade)

A tarefa consistiu na importação de uma tabela com 2 colunas e 50 valores, seguida do uso da função SOMA. Na comparação exibida na Figura 7, o Calc apresentou maior consumo de RAM e CPU. Os tempos de inicialização foram idênticos e os coeficientes de variação, aceitáveis.

Figura 7: Resultados dos testes comparativos entre Excel e LibreOffice Calc.

Excel	Tempo (S)	RAM(MB)	CPU (%)	Calc	Tempo (S)	RAM(MB)	CPU (%)
Teste 1	4	188	8,14	Teste 1	6	321	9,72
Teste 2	5	156	9,79	Teste 2	4	388	8,88
Teste 3	6	172	8,89	Teste 3	5	341	8,85
Média	5	172	8,94	Média	5	350	9,15
D.Padrão	1	16,0	0,8	D.Padrão	1,00	34,39	0,49
CV	20%	9%	9%	CV	20%	10%	5%



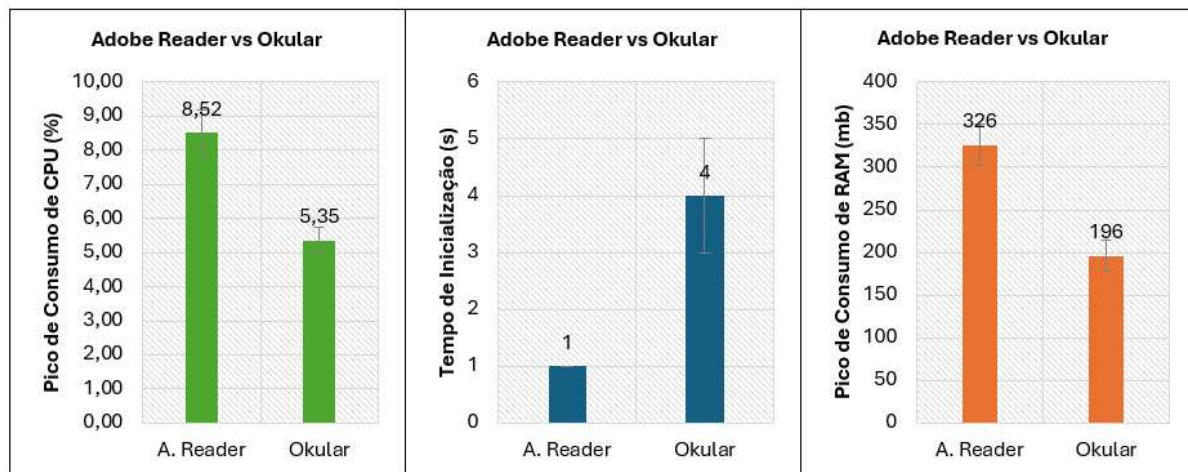
Fonte: Elaboração própria (2025).

## 7.2.8 Acrobat Reader vs Okular (Leitura de PDF)

A tarefa foi a abertura de um arquivo PDF de 100 MB com 50 páginas, seguida de rolagem completa e marcações nas páginas ímpares. A análise da Figura 8 revela que o Okular demonstrou menor uso de RAM e CPU, mas com um tempo médio de inicialização superior (4 s vs 1 s). Os CVs foram considerados bons, com exceção da variação no tempo de inicialização do Okular (CV = 25%).

Figura 8: Resultados dos testes comparativos entre Adobe Reader e Okular.

A. Reader	Tempo (S)	RAM(MB)	CPU (%)	Okular	Tempo (S)	RAM(MB)	CPU (%)
Teste 1	1	333	7,76	Teste 1	3	184	5,74
Teste 2	1	299	8,95	Teste 2	5	187	4,98
Teste 3	1	346	8,85	Teste 3	4	217	5,33
Média	1	326	8,52	Média	4	196	5,35
D.Padrão	0	24,3	0,7	D.Padrão	1,00	18,25	0,38
CV	0%	7%	8%	CV	25%	9%	7%



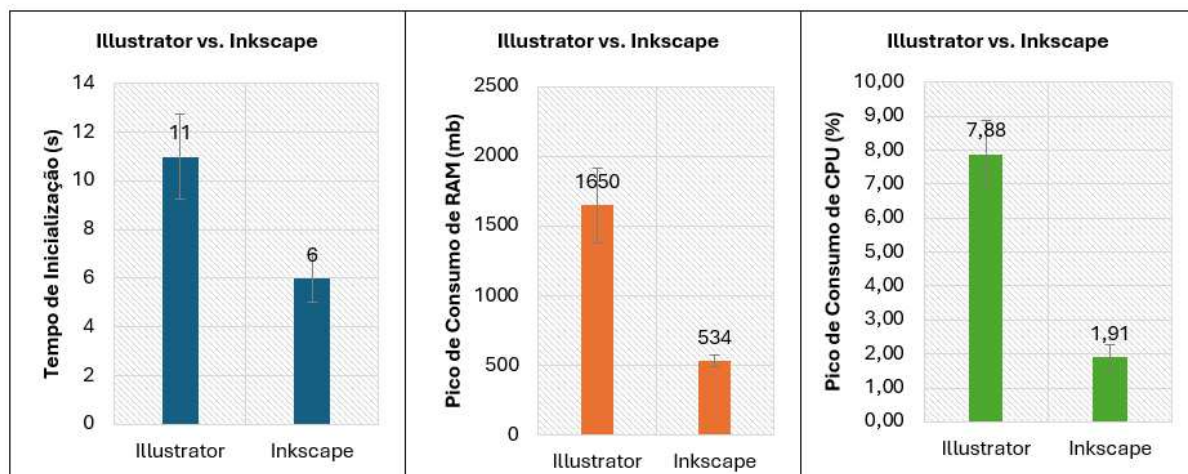
Fonte: Elaboração própria (2025).

### 7.2.9 Illustrator vs Inkscape (Design vetorial)

A tarefa foi a criação de uma ilustração vetorial de um cubo. Conforme os dados da Figura 9, o Inkscape consumiu menos RAM, mas teve um desempenho levemente inferior em termos de uso de CPU. Ambos os softwares demonstraram estabilidade satisfatória durante os testes.

Figura 9: Resultados dos testes comparativos entre Illustrator e Inkscape.

Illustrator	Tempo (S)	RAM(MB)	CPU (%)	Inkscape	Tempo (S)	RAM(MB)	CPU (%)
Teste 1	12	1790	9,01	Teste 1	5	489	1,68
Teste 2	9	1820	7,32	Teste 2	7	552	2,35
Teste 3	12	1340	7,31	Teste 3	6	561	1,70
Média	11	1650	7,88	Média	6	534	1,91
D.Padrão	1,732051	268,9	1,0	D.Padrão	1,00	39,23	0,38
CV	16%	16%	12%	CV	17%	7%	20%



Fonte: Elaboração própria (2025).

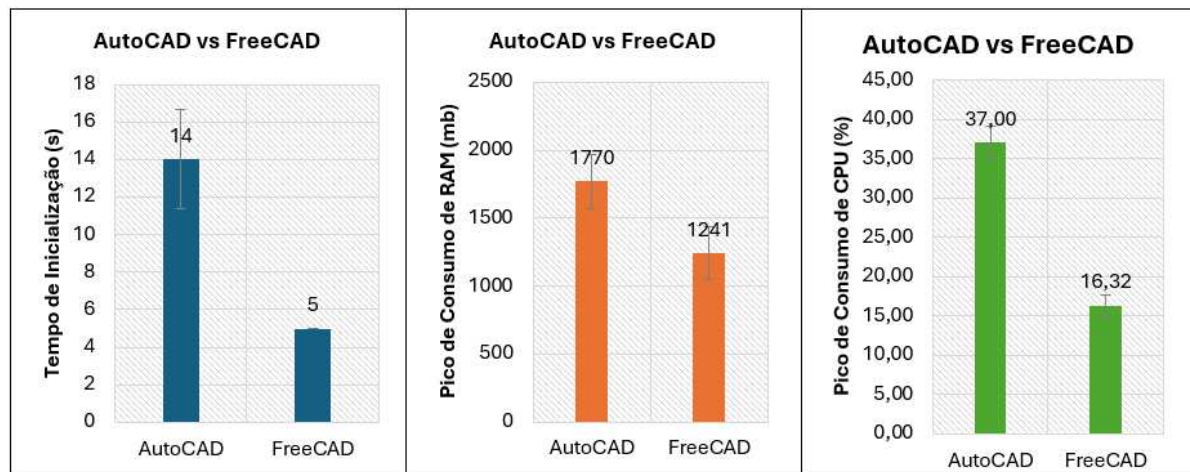
#### 7.2.10 AutoCAD vs FreeCAD (Modelagem técnica e CAD)

A tarefa consistiu na criação de um modelo 3D de um cubo. A Figura 10 demonstra que o AutoCAD foi mais rápido na inicialização (14 s vs 5 s), porém o FreeCAD se mostrou mais econômico no uso de recursos de hardware. O CV de RAM no FreeCAD (16%) foi considerado dentro da faixa aceitável.



Figura 10: Resultados dos testes comparativos entre AutoCAD e FreeCAD.

AutoCAD	Tempo (S)	RAM(MB)	CPU (%)	FreeCAD	Tempo (S)	RAM(MB)	CPU (%)
Teste 1	15	1712	35,33	Teste 1	5	489	1,68
Teste 2	11	1991	39,45	Teste 2	7	552	2,35
Teste 3	16	1607	36,22	Teste 3	6	561	1,70
Média	14	1770	37,00	Média	6	534	1,91
D.Padrão	2,64	198,5	2,2	D.Padrão	1,00	39,23	0,38
CV	19%	11%	6%	CV	17%	7%	20%

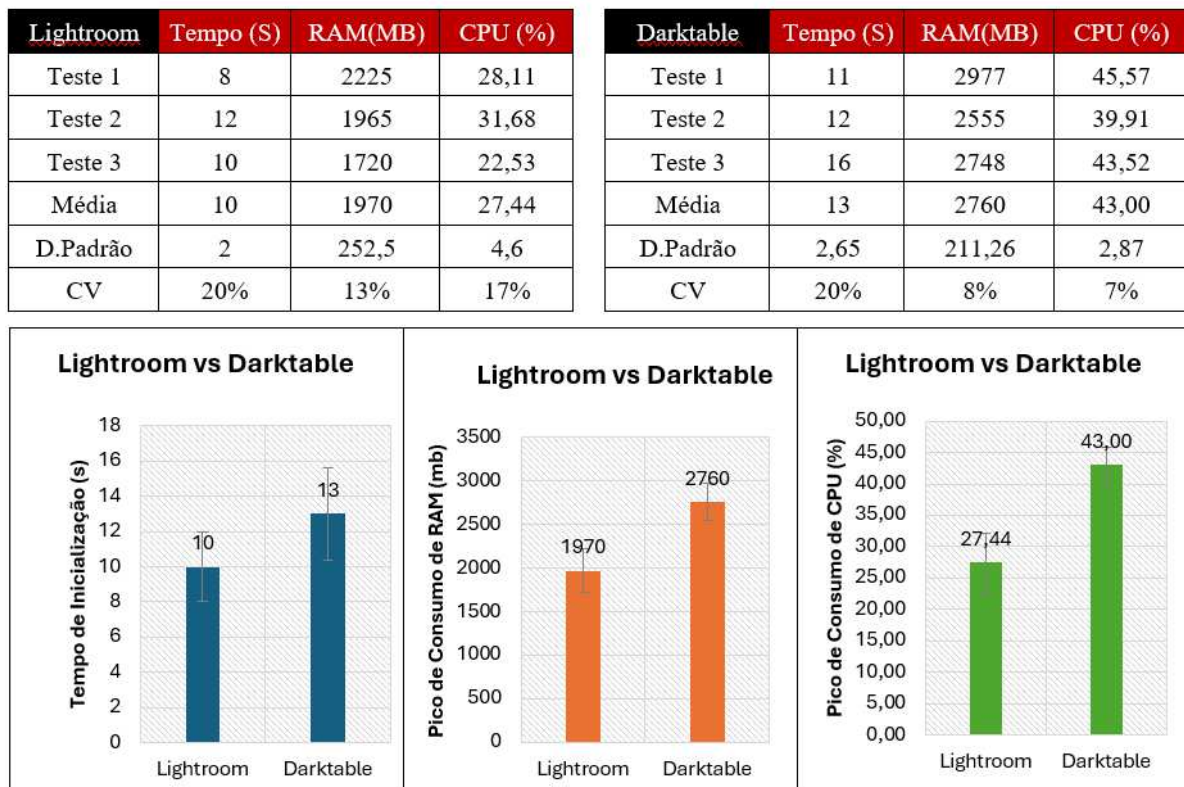


Fonte: Elaboração própria (2025).

### 7.2.11 Lightroom vs Darktable (Tratamento de imagem RAW)

A tarefa foi a aplicação de um filtro sépia e a impressão da imagem. Os testes (Figura 11) mostram que o Darktable consumiu mais RAM que o Lightroom, mas a variação absoluta foi considerada aceitável. Com tempos de inicialização similares, o desempenho técnico de ambos foi julgado compatível.

Figura 11: Resultados dos testes comparativos entre Lightroom e Darktable.



Fonte: Elaboração própria (2025).

### 7.3 Comparação de desempenho de sistemas operacionais

Nesta etapa do trabalho, foi realizada uma análise comparativa entre dois sistemas operacionais: Windows 11 (proprietário) e Zorin OS (baseado em Linux, de código aberto). Ambos foram instalados em *dual boot* na mesma máquina, garantindo que o hardware utilizado fosse exatamente o mesmo para todos os testes, o que elimina variáveis externas e assegura a confiabilidade dos resultados.

Os testes foram divididos em dois blocos, com os resultados compilados na Figura 12:

- **Tempo de Inicialização (Boot Time):** Cronometrado manualmente do momento da escolha no GRUB até a tela totalmente carregada.
- **Benchmark de Desempenho (Geekbench 6.4.0):** Foram utilizadas seis métricas de desempenho:
  - **Single-Core Score:** Avalia a capacidade de processamento de um único núcleo da CPU.
  - **Multi-Core Score:** Mede o desempenho total quando todos os núcleos do processador são utilizados.
  - **Clang Compilation:** Simula tarefas de compilação de software, avaliando o desempenho em tarefas típicas de desenvolvedores.
  - **File Compression:** Mede a velocidade e eficiência na compressão de arquivos.

- **Ray Tracing:** Avalia o desempenho gráfico com renderização baseada em simulação de luz (importante para jogos e modelagem 3D).

O Windows obteve desempenho superior na maioria dos testes de CPU e gráficos. Notavelmente, o Zorin OS se destacou na renderização com Ray Tracing (10033 pontos). Apesar da performance inferior do Zorin OS em alguns benchmarks, a diferença não inviabiliza seu uso para tarefas cotidianas, especialmente ao considerar seu custo zero e filosofia de código aberto.

Figura 12: Resultados dos testes comparativos entre Windows 11 e Zorin OS.

Windows 11	Tempo	Single-Core Score	Multi-Core Score	Clang	File Compression	Ray Tracer
Teste 1	8	1522	5028	8879	2448	9165
Teste 2	11	1535	4917	8887	2441	9091
Teste 3	11	1485	5352	9219	2518	9278
Média	10	1514	5099	8995	2469	9178
Desvio Padrão	1,73	25,94	226,02	194,03	42,58	94,18
CV	17,32%	1,71%	4,43%	2,16%	1,72%	1,03%

Zorin OS	Tempo	Single-Core Score	Multi-Core Score	Clang	File Compression	Ray Tracer
Teste 1	13	1163	4442	9299	1612	10114
Teste 2	14	1089	4399	9275	1771	9999
Teste 3	18	1219	4473	9278	1567	9986
Média	15	1157	4438	9284	1650	10033
Desvio Padrão	2,65	65,21	37,16	13,08	107,18	70,45
CV	17,64%	5,64%	0,84%	0,14%	6,50%	0,70%



Fonte: Elaboração própria (2025).

## 7.4 Síntese da Discussão

Os dados apresentados nesta seção indicam que as alternativas de código aberto oferecem desempenho comparável às soluções proprietárias em várias tarefas, ainda que com variações em tempo de resposta e uso de recursos. Em muitos casos, como no uso de RAM ou CPU, os softwares livres foram mais eficientes. Já em tarefas mais exigentes, como edição de vídeo ou modelagem 3D, os softwares proprietários ainda oferecem vantagens em tempo de execução ou estabilidade.

Essas observações sustentam a ideia de que o código aberto pode mitigar a pirataria, oferecendo alternativas viáveis especialmente em contextos educacionais, profissionais emergentes e países com barreiras econômicas ao licenciamento de software.

## 8 Conclusão

A pesquisa constatou que a pirataria de software é um problema de múltiplas camadas, cujas raízes não são apenas legais ou técnicas, mas também econômicas, sociais e culturais. No Brasil, essa prática se agrava devido ao difícil acesso a programas proprietários e à normalização de hábitos como o "jeitinho" e a informalidade digital.

Como contraponto, as soluções de código aberto surgem como um caminho viável, ético e sustentável. A análise comparativa revelou que, em critérios como estabilidade e performance, muitas ferramentas open source são equivalentes às suas contrapartes pagas. Seus maiores trunfos são a gratuidade, a flexibilidade, a segurança e o suporte colaborativo da comunidade.

Além disso, o incentivo ao software livre representa um passo importante para a soberania tecnológica, a inclusão digital e a diminuição da dependência de práticas ilegais. Ao ser visto como uma solução sistêmica — e não apenas uma substituição técnica —, o ecossistema open source promove um acesso à tecnologia fundamentado na liberdade e na cooperação.

Conclui-se, assim, que é fundamental uma mobilização conjunta entre políticas públicas, instituições de ensino e a sociedade para fomentar a adoção do código aberto. Tal esforço, focado em capacitação e conscientização, não apenas reduzirá os danos da pirataria, mas também fortalecerá uma cultura de inovação e cidadania digital.

## Referências

AGUIAR, V. M. d. O. *Software livre, cultura hacker e o ecossistema da colaboração*. São Paulo: Momento Editorial, 2009.

ALMEIDA, A. C. *A cabeça do brasileiro*. Rio de Janeiro: Record, 2007.

ALVES, P. *Seis riscos de usar programas crackeados*. 2018. Techtudo. Disponível em: <<https://www.techtudo.com.br/listas/2018/12/seis-riscos-de-usar-programas-crackeados.ghml>>. Acesso em: 18 jul. 2025.

ALVES, V. H. L. d. M. Inteligência artificial generativa e direito autoral: investigando os limites do uso justo na era da tecnologia. *Revista de Direito, Inovação, Propriedade Intelectual e Concorrência*, v. 10, n. 1, p. 01–19, 2024. Jan./jul. Disponível em: <<https://doi.org/10.5281/zenodo.10581245>>. Acesso em: 18 jul. 2025.

ASSOCIAÇÃO BRASILEIRA DA PROPRIEDADE INTELECTUAL – ABPI. *Impacto econômico da pirataria no Brasil*. São Paulo: [s.n.], 2023. Disponível em: <<https://exame.com/economia/em-2023-27-das-vendas-de-bebidas-no-brasil-foram-ilicitas-e-perda-com-pirataria-cresceu-20/>>. Acesso em: 10 jun. 2025.

BSA – THE SOFTWARE ALLIANCE. *The compliance gap: BSA Global Software Survey*. Washington, D.C.: [s.n.], 2017. Disponível em: <<https://gss.bsa.org/2016/>>. Acesso em: 15 jul. 2025.

BSA – THE SOFTWARE ALLIANCE. *Software management: security imperative, business opportunity – Global software survey in brief*. Washington, DC: [s.n.], 2018. Disponível em: <[https://globalstudy.bsa.org/2018/downloads/studies/2018GlobalSoftwareSurvey\\_inbrief\\_en.pdf](https://globalstudy.bsa.org/2018/downloads/studies/2018GlobalSoftwareSurvey_inbrief_en.pdf)>. Acesso em: 18 jul. 2025.

CAMPOS, A. *O que é Linux*. Florianópolis: [s.n.], 2006. BR-Linux. Disponível em: <<http://brlinux.org/faq-linux>>. Acesso em: 20 jun. 2025.

CARDOZO, R. N. An experimental study of consumer effort, expectation and satisfaction. *Journal of Marketing Research*, v. 2, n. 3, p. 244–249, 1965.

CASALI, R. R. B. et al. Pirataria de software: uma análise da relação entre comportamento ético, atitude e intenção do consumidor. In: *Anais do ENCONTRO NACIONAL DA ASSOCIAÇÃO NACIONAL DE PÓS-GRADUAÇÃO E PESQUISA EM ADMINISTRAÇÃO*. Rio de Janeiro: ANPAD, 2010.

CASTELLS, M. *A sociedade em rede*. 2. ed. São Paulo: Paz e Terra, 1999.

CERIONI, T. A. *Principal preocupação dos executivos de TI é gestão, e não tecnologia*. 2003. Learning & Performance Brasil. Disponível em: <<http://www.dicas-l.com.br/20050418.html>>. Acesso em: 29 jun. 2025.

COLEMAN, B.; HILL, M. How free became open and everything else under the sun: Introduction. *M/C Journal*, v. 7, n. 3, 2004. Disponível em: <<https://journal.media-culture.org.au/index.php/mcjournal/article/view/2352>>.



CRUZ, L. F. P. d. R. A. *Games, pirataria e sociedade...* Monografia (Trabalho de Conclusão de Curso) — Universidade Federal do Rio de Janeiro.

DAS, M. *Software Piracy*. Monografia (Bachelor's Thesis), 2018.

DIDIO, L. *North American Linux and Windows TCO comparison, Part 1*. [S.l.], 2005. Disponível em: <[https://download.microsoft.com/download/2/8/7/287fda62-1479-48b7-808c-87333312b93e/Yankee\\_TCO.pdf](https://download.microsoft.com/download/2/8/7/287fda62-1479-48b7-808c-87333312b93e/Yankee_TCO.pdf)>. Acesso em: 22 mai. 2025.

DORNELAS, G. C.; CLEM, T. M. d. S. *Cibersegurança nas relações internacionais: um estudo de continuidades*. Monografia (Trabalho de Conclusão de Curso (Graduação em Relações Internacionais)), Belo Horizonte, 2024.

FERES, M. V. C.; OLIVEIRA, J. V. d.; GONÇALVES, D. D. Robin hood às avessas: software, pirataria e direito autoral. *Revista de Direito GV*, São Paulo, v. 13, n. 1, p. 71–90, 2017. Disponível em: <<https://doi.org/10.1590/2317-6172201704>>. Acesso em: 18 jul. 2025.

FERREIRA, M. C.; COSTA, S. M. A. A pirataria e seus impactos na competitividade das empresas brasileiras. *Revista Brasileira de Propriedade Intelectual*, v. 18, n. 34, p. 23–40, 2019.

FINE, C. et al. To buy or to pirate? a model of consumer choice for intellectual property. *Journal of Research in Interactive Marketing*, v. 13, n. 4, p. 555–572, 2019. Disponível em: <<https://doi.org/10.1108/JRIM-07-2018-0093>>.

GARCIA, M. N. et al. Software livre em relação ao software proprietário: aspectos favoráveis e desfavoráveis percebidos por especialistas. *Gestão & Regionalidade*, v. 26, n. 78, p. 106–121, 2010.

GAY, J. *Free software, free society: selected essays of Richard M. Stallman*. Boston: GNU Press, 2007. Disponível em: <<http://www.leonardi.adv.br/blog/wp-content/uploads/2007/06/rms-essays.pdf>>. Acesso em: 8 mai. 2025.

GOFF, M. *Software Piracy Statistics – 2025 Outlook*. 2024. Revenera Blog. Disponível em: <<https://www.revenera.com/blog/software-monetization/software-piracy-stat-watch/>>.

GROHMANN, M. Z. et al. Os jovens e a pirataria: investigando o uso não autorizado de softwares. *Revista de Administração da UNIMEP*, Santa Bárbara d'Oeste, v. 13, n. 1, p. 81–106, 2015.

HEXSEL, R. A. *Propostas de ações de Governo para incentivar o uso de software livre*. Curitiba, 2002. Disponível em: <[http://www.inf.ufpr.br/info/techrep/RT\\_DINF004\\_2002.pdf](http://www.inf.ufpr.br/info/techrep/RT_DINF004_2002.pdf)>. Acesso em: 29 mai. 2025.

KHADKA, I. *Software Piracy: A study of causes, effects and preventive measures*. 2015.

KONSTAM, A. *Piracy: the complete history*. New York: Osprey, 2008.

MIZUKAMI, P. N. et al. Chapter 5: Brazil. In: KARAGANIS, J. (Ed.). *Media piracy in emerging economies*. New York: Social Science Research Council, 2011. Disponível em: <<http://piracy.ssrc.org/>>. Acesso em: 19 jul. 2025.

PEIRANO, M.; UGARTE, D. d. *Richard Stallman: software livre não é pela direita nem pela esquerda*. 2004. Linux Logic. Acesso em: 18 mai. 2025.

RIEHLE, D. The economic motivation of open source: Stakeholder perspectives. *IEEE Computer*, v. 40, n. 4, p. 25–32, 2007. Disponível em: <<https://doi.org/10.1109/MC.2007.139>>. Acesso em: 19 jul. 2025.

ROSSI, G.; BOCHNIA, L. *Chamando cada coisa pelo nome correto: as diferenças entre o Software Livre e o Código Aberto*. 2023. Latinoware.

SABINO, V. C. *Um estudo sistemático de licenças de software livre*. Monografia (Dissertação (Mestrado)), São Paulo, 2011.

SALDANHA, A. Pirataria digital e função social dos direitos autorais na hiperconectividade. *REJUR – Revista Jurídica da UFERSA*, Mossoró, v. 8, n. 15, p. 67–88, 2024. ISSN 2526-9488.

SALEH, A. M. *Adoção de tecnologia: um estudo sobre o uso de software livre nas empresas*. Monografia (Dissertação (Mestrado)), São Paulo, 2004.

SERRANO, A.; GUERREIRO, A.; CALDEIRA, M. *Gestão de sistemas e tecnologias de informação*. Lisboa: FCA-Informática, 2004.

SILVEIRA, S. A. d. Inclusão digital, software livre e globalização contra-hegemônica. *Parcerias Estratégicas*, n. 20, p. 211–221, 2005.

STALLMAN, R.; Free Software Foundation. *Free software, free society: selected essays of Richard M. Stallman*. Cambridge: [s.n.], 2015.

SUDIBJO, M. N. S. *The Proliferation of Open-Source Licenses: The Cooptation of an Industrial Transition Movement by Multiple Institutions*. 2017. Disponível em: <<https://ir.vanderbilt.edu/handle/1803/13784>>.

Symantec. *Types of Piracy*. Disponível em: <<https://www.symantec.com/en/uk/about/legal/anti-piracy/types-piracy>>.

TEIXEIRA, T. *Direito empresarial sistematizado: doutrina, jurisprudência e prática*. 7. ed. São Paulo: Saraiva Educação, 2018.

TIETZMANN, R.; PASE, A. F.; PIOVESAN, G. P. Um meio moldado pela gambiarra: como trapanças e regulações econômicas nos videogames moldaram a cultura digital nacional. *Zanzalá*, v. 12, n. 1, 2024.

WAYKAR, Y.; LAD, K. Software piracy: A global to local scenario. *International Journal for Research in Engineering Application & Management (IJREAM)*, v. 4, n. 11, 2019.