

CENTRO UNIVERSITÁRIO UNIDADE DE ENSINO SUPERIOR DOM BOSCO  
CURSO ENGENHARIA DE SOFTWARE

**ITALO RICHARD'SONN DA SILVA NEVES**

**SOFTWARE LIVRE:** uma análise sobre seu papel transformador no cenário digital

São Luís  
2023

**ITALO RICHARD'SONN DA SILVA NEVES**

**SOFTWARE LIVRE:** uma análise sobre seu papel transformador no cenário digital

Monografia apresentada ao Curso de Engenharia de Software do Centro Universitário Unidade de Ensino Superior Dom Bosco como requisito parcial para obtenção do grau de Bacharel em Engenharia de Software.

Orientador: Prof. Me. Allisson Jorge Silva Almeida.

São Luís

2023

Dados Internacionais de Catalogação na Publicação (CIP)

Centro Universitário - UNDB / Biblioteca

Neves, Italo Richard'sonn da Silva

softwarew livre: uma análise sobre seu papel transformador no cenário digital. / Italo Richard'sonn da Silva Neves. \_\_ São Luís, 2023.

62 f.

Orientador: Prof. Me. Allisson Jorge Silva Almeida.

Monografia (Graduação em Administração) - Curso de Engenharia de Software – Centro Universitário Unidade de Ensino Superior Dom Bosco – UNDB, 2023.

1. Software Livre. 2. Sistema operacional. 3. Segurança. 4. Privacidade. I. Título.

CDU 004.4

**ITALO RICHARD'SONN DA SILVA NEVES**

**SOFTWARE LIVRE:** uma análise sobre seu papel transformador no cenário digital

Monografia apresentada ao Curso de Engenharia de Software do Centro Universitário Unidade de Ensino Superior Dom Bosco como requisito parcial para obtenção do grau de Bacharel em Engenharia de Software.

Aprovada em: 04/12/2023.

**BANCA EXAMINADORA:**

---

**Allisson Jorge Silva Almeida (Orientador)**

Mestre em Engenharia de Eletricidade

Centro Universitário Unidade de Ensino Superior Dom Bosco (UNDB)

---

**Prof. Me. Arlison Wady Sousa Martins**

Mestre em Ciência da Computação

Centro Universitário Unidade de Ensino Superior Dom Bosco (UNDB)

---

**Prof. Me. Suzane Carvalho dos Santos**

Mestra em Engenharia de Eletricidade

Centro Universitário Unidade de Ensino Superior Dom Bosco (UNDB)

Dedico a Deus nosso senhor, a  
minha mãe, meu pai, minha  
família.

## **AGRADECIMENTOS**

Agradeço inicialmente a todos que me apoiaram nessa jornada, sempre me dando suporte, conselhos e aprendizado, aos meus amigos e colegas de classe, juntamente ao corpo docente que se fizeram presentes nesses quatro anos e meio de curso, a minha família e a minha amável noiva, de onde parte o meu maior suporte e incentivo.

## RESUMO

Este trabalho tem como foco a exploração e análise dos conceitos e filosofia do Software Livre, com ênfase no sistema operacional GNU/Linux, seu caso de sucesso mais proeminente. Ambos desempenham papéis cruciais na tecnologia atual, promovendo valores de liberdade, segurança e privacidade. O objetivo geral é fornecer uma visão sobre o funcionamento e a influência desses tópicos na sociedade.

Os objetivos específicos incluem a análise da filosofia do Software Livre, a compreensão de suas características e licenças, e a avaliação de sua relação com segurança e privacidade. Além disso, o trabalho busca compreender o crescimento e a aceitação do GNU/Linux, seus impactos sociais e culturais.

Realizando a coleta de dados por meio de documentação, livros e análise de várias fontes.

A metodologia utilizada é predominantemente de pesquisa descritiva, abrangendo diversas áreas, como linguagens de programação, hardware, filosofia do Software Livre, distribuições de sistemas operacionais, arquitetura de sistemas de computação e modelos de negócios relacionados a software. As fontes de dados incluem documentação oficial, livros especializados e artigos científicos relacionados aos tópicos abordados.

Palavras-chave: GNU/Linux, Privacidade, Segurança, Kernel

## **ABSTRACT**

This work focuses on the exploration and analysis of the concepts and philosophy of Free Software, with an emphasis on the GNU/Linux operating system, its most prominent success case. Both play crucial roles in current technology, promoting values of freedom, security, and privacy. The overall goal is to provide an insight into the functioning and influence of these topics in society.

Specific objectives include the analysis of Free Software philosophy, understanding its characteristics and licenses, and evaluating its relationship with security and privacy. Additionally, the work seeks to understand the growth and acceptance of GNU/Linux, its social and cultural impacts. Data collection is conducted through documentation, books, and analysis of various sources.

The methodology employed is predominantly descriptive research, covering diverse areas such as programming languages, hardware, Free Software philosophy, operating system distributions, computer system architecture, and business models related to software. Data sources include official documentation, specialized books, and scientific articles related to the topics addressed.

Keywords: GNU/Linux, Privacy, Security, Kernel



## LISTA DE FIGURAS

|   |    |
|---|----|
| Figura 1 - Preço do Software Mathematica..... | 30 |
| Figura 2 - Planos de Suporte Red Hat.....     | 31 |

## LISTA DE GRÁFICOS

|   |    |
|---|----|
| Gráfico 1 - Número de Linhas por Atualização do Kernel.....                 | 35 |
| .....   | 35 |
| Gráfico 2 - Porcentagem Aproximada de Usuários por Sistema Operacional..... | 37 |
| .....   | 37 |
| Gráfico 3 - Número de Modificações no Kernel por Contribuidores.....        | 38 |
| Fonte - Pingdom, 2012.....  | 38 |

## LISTA DE ABREVIATURAS E SIGLAS

|         |  |
|---------|--|
| CD      | Change Directory (Mudar Diretório).  |
| CMP     | Consent Management Provider (Provedor de Gerenciamento de Consentimento).              |
| CPU     | Central Processing Unit (Unidade Central de Processamento).                            |
| CP      | Copy (Copiar).   |
| CURL    | Client for URLs (Cliente para URLs).   |
| GNU     | GNU 's Not Unix (GNU Não é Unix).  |
| GNU/GPL | GNU General Public License (Licença Pública Geral GNU).                                |
| GNOME   | GNU Network Object Model Environment (Ambiente GNU para Modelagem de Objetos em Rede). |
| KDE     | K Desktop Environment (Ambiente de Área de Trabalho K).                                |
| LS      | List (Listar).   |
| LXQT    | Lightweight Qt Desktop Environment (Ambiente de Área de Trabalho Qt Leve).             |
| MALLOC  | Memory Allocation (Alocação de Memória).   |
| MLS     | Multi Level Security (Segurança de Múltiplos Níveis).                                  |
| MV      | Move (Mover).  |
| NSA     | National Security Agency (Agência de Segurança Nacional).                              |
| RAM     | Random Access Memory (Memória de Acesso Aleatório).                                    |
| SB      | Secure Boot (Inicialização Segura).  |
| SELinux | Security-Enhanced Linux (Linux Reforçado com Segurança).                               |
| SUDO    | Superuser do (Super Usuário Faz).  |
| TOP     | Task Operation Program (Programa de Operações de Tarefas).                             |
| UEF     | Unified Extensible Firmware Interface (Interface de Firmware Extensível Unificada).    |
| WGET    | Web Get (Obtenção da Web).   |
| XFCE    | XForms Common Environment (Ambiente Comum XForms).                                     |

WGET      Web Get (Obtenção da Web).

## SUMÁRIO

|   |    |
|---|----|
| 1 INTRODUÇÃO.....   | 13 |
| 1.1 OBJETIVOS.....  | 14 |
| 1.1.1 OBJETIVOS GERAIS.....   | 14 |
| 1.1.2 OBJETIVOS ESPECÍFICOS.....  | 14 |
| 1.2 DELIMITAÇÃO DO TRABALHO.....  | 15 |
| 1.3 JUSTIFICATIVA.....  | 15 |
| 1.4 METODOLOGIA.....  | 16 |
| 1.4.1 TIPO DE PESQUISA.....   | 16 |
| 1.4.2 ÁREA DE ABRANGÊNCIA.....  | 17 |
| 2 DESENVOLVIMENTO.....  | 18 |
| 2.1 SOFTWARE LIVRE, FILOSOFIA, USO E DIFUSÃO.....                               | 18 |
| 2.1.1 CARACTERIZAÇÃO E DISTRIBUIÇÃO DE SOFTWARE LIVRE.....                      | 23 |
| 2.1.2 LICENÇAS DE USO, GNU GPL E MIT.....                                       | 24 |
| 2.1.3 SOFTWARE LIVRE, SEGURANÇA E SUA PRIVACIDADE.....                          | 26 |
| 2.1.4 OPEN SOURCE.....  | 27 |
| 2.1.4 CONSUMO E MONETIZAÇÃO.....  | 28 |
| 2.2 GNU/LINUX, SEU CRESCIMENTO, ACEITAÇÃO E FUNCIONAMENTO.....                  | 33 |
| 2.2.1 ENTENDO O SOFTWARE E SUA ACEITAÇÃO.....                                   | 33 |
| 2.2.2 IMPACTOS SOCIAIS, CULTURAIS E SOCIOLÓGICOS.....                           | 38 |
| 2.2.3 SEU KERNEL, FUNCIONAMENTO E PROCESSOS.....                                | 39 |
| 2.2.4, CPU, SEU FUNCIONAMENTO E PRINCÍPIOS BÁSICOS EMPREGADOS AO GNU/LINUX..... | 41 |
| 2.2.6 GNU/LINUX, SUAS DISTRIBUIÇÕES, ARQUITETURA E VARIEDADE TÉCNICA.....       | 44 |
| 2.2.7 PRIVACIDADE E SEGURANÇA COM GNU/LINUX.....                                | 48 |
| 3 CONCLUSÃO.....  | 54 |
| REFERÊNCIAS.....  | 57 |

## 1 INTRODUÇÃO

Este estudo focaliza a análise e discussão de temas fundamentais, inicialmente explorando os conceitos e a filosofia subjacentes ao Software Livre. Aprofundaremos nossa investigação no seu principal expoente, o sistema operacional GNU/Linux, destacando suas contribuições essenciais no cenário tecnológico contemporâneo. Em uma era onde a liberdade, segurança e privacidade desempenham papéis cruciais na utilização de softwares, desde corporações até usuários cotidianos, examinaremos como esses elementos fundamentais moldam e influenciam diversas camadas de utilizadores.

Em meio ao cenário contemporâneo, onde softwares e sistemas operacionais desempenham funções cruciais na sociedade e economia, torna-se essencial compreender a influência direta e a relevância do software livre em nossa vida. Este estudo propõe uma análise aprofundada desses tópicos, visando destacar como o Software Livre e o GNU/Linux continuam a moldar de maneira positiva o panorama tecnológico. Através dessa exploração, buscamos evidenciar como esses elementos fortalecem valores fundamentais, como liberdade, conhecimento e segurança, desempenhando um papel vital no desenvolvimento e uso de sistemas na era digital em constante evolução.

A abordagem dos tópicos será estruturada de forma a facilitar a compreensão do leitor, onde começaremos abordando o conceito fundamental sobre o que caracteriza um Software Livre, junto a sua filosofia de disseminação e compartilhamento de conhecimento e ideais nas estruturas seus pilares, com esse contexto, exploraremos a sua história de criação, princípios e as licenças que o regem.

O sistema operacional GNU/Linux, representa de maneira emblemática o que um Software Livre representa, desempenhando um papel de destaque na indústria da tecnologia, com seu constante crescimento e ampla aceitação do público, especialmente em servidores e dispositivos embarcados, refletindo a confiabilidade, flexibilidade e segurança oferecida pelo sistema, com isso nos aprofundaremos na compreensão e análise do seu Kernel, o núcleo do sistema que

gerencia processos e coordena recursos, oferecendo suporte a uma variedade abrangente de hardware.

A abordagem da interseção entre Software Livre, segurança e privacidade em um mundo digital cada vez mais interconectado em todos os âmbitos acaba tornando estes pontos preocupações centrais, com isso destacamos a abordagem do GNU/Linux em relação a segurança, que engloba desde estrito controle de permissões de arquivos, até a implementação de técnicas de criptografia, separação de privilégios e controle rigoroso de permissões e firewall, mostrando como o sistema prioriza a proteção de dados e sistema, garantindo que apenas usuários autorizados possam acessar ou modificar recursos do sistema.

## **1.1 OBJETIVOS**

Este trabalho tem como propósito analisar e compreender a influência do Software Livre, concentrando-se no GNU/Linux, nos aspectos de segurança, privacidade e seus impactos nas esferas social e cultural.

### **1.1.1 OBJETIVOS GERAIS**

Explorar a relevância do Software Livre, especialmente do GNU/Linux, na sociedade atual, destacando seus impactos em aspectos culturais, sociais, filosóficos, e financeiros, bem como na privacidade. Contrastar essas escolhas com práticas corporativas, analisando seu impacto na segurança do usuário.

### **1.1.2 OBJETIVOS ESPECÍFICOS**

Filosofia e evolução do software livre, investigar as raízes e a evolução do movimento de Software Livre, analisando não apenas o significado de software livre, mas também como essa abordagem revolucionou a relação com a tecnologia, influenciando aspectos culturais.

Direito a privacidade no mundo digital, baseando-se nos fundamentos delineados por Warren e Brandeis em 1890 para definir o conceito de privacidade,

destacar sua crescente importância no mundo digital e explorar como o software livre pode contribuir para a transparência e controle.

GNU/Linux como case de sucesso, investigar a história do GNU/Linux, personificado por Linus Torvalds e examinar como a comunidade desempenha um papel fundamental em seu desenvolvimento, compreender as contribuições específicas do Linux para a segurança, privacidade e seus reflexos nas esferas sociais e culturais

Impactos sociais e culturais do Software Livre, analisar a adoção de Software Livre, exemplificado pelo GNU/Linux e seu profundo impacto nas esferas sociais e culturais, destacar como a liberdade de acesso e modificação estimula a inovação e a cultura colaborativa.

## **1.2 DELIMITAÇÃO DO TRABALHO**

O trabalho se delimita a exemplificar e discorrer sobre a utilização do software, seu sucesso, filosofia e distribuição e a comparação direta com software proprietário, com o discorrimento sobre seu maior case, o GNU/Linux

## **1.3 JUSTIFICATIVA**

O software livre desempenha um papel fundamental na sociedade atual, abrangendo uma variedade de áreas sociais e culturais. Em um mundo cada vez mais diversificado e interconectado, a privacidade e a segurança tornaram-se fatores determinantes na utilização de aplicativos e sistemas que lidam com essas questões sensíveis. A capacidade de garantir a proteção das informações pessoais e a integridade dos sistemas é um diferencial significativo na era digital. Portanto, a disseminação e compreensão dos benefícios e vantagens oferecidos por aplicações que priorizam a excelência nesses aspectos são cruciais.

Neste contexto, este trabalho se propõe a investigar a filosofia e os princípios do software livre, assim como seu funcionamento e sua relevância na garantia de privacidade e segurança. A compreensão desses tópicos é essencial para capacitar a sociedade a fazer escolhas informadas sobre as ferramentas



tecnológicas que utiliza. Além disso, ao explorar o crescimento e aceitação do sistema GNU/Linux, bem como seu impacto social e cultural, este estudo visa fornecer uma visão abrangente das implicações do software livre.

Dessa forma, este estudo visa fundamentar a importância de difundir o conhecimento acerca do software livre, evidenciando sua capacidade de salvaguardar a privacidade e a segurança dos usuários, além de contribuir para uma sociedade mais consciente e interconectada. A promoção desses princípios revela-se crucial para possibilitar que um público mais abrangente desfrute das vantagens proporcionadas por aplicações que primam pela excelência nessas áreas críticas.

## **1.4 METODOLOGIA**

A metodologia utilizada na construção deste trabalho se dá pela pesquisa e desenvolvimento de software, por meio de documentação e livros, abrangendo áreas de debate sobre GNU/Linux, construção e compartilhamento de software, distribuições, hardware e filosofia. Trata-se da apresentação de um ponto de vista sobre o uso, funcionamento e conceitos de sistemas operacionais e software.

Utilizando a metodologia de coleta de dados, por meio de diversas fontes e realizando sua análise para a geração de informações sobre vários temas relacionados aos pontos apresentados, o objetivo é apresentar e estudar os aspectos e nuances presentes, relacionando os temas para obter um melhor resultado didático. Isso envolve a aplicação conjunta de conhecimento prático e teórico para a resolução dessas questões, a fim de elucidar pontos e processos que serão apresentados ao longo do trabalho.

### **1.4.1 TIPO DE PESQUISA**

Pesquisa descritiva onde se possui o objetivo principal de descrever as características dos tópicos abordados no contexto apresentado, buscando compreender e retratar eventos e seu devido contexto, analisando dados,

documentação e livros. Analisando e interpretando as informações, entendo a abordagem e identificando lacunas , fundamentando e contextualizando os tópicos abordados.

#### **1.4.2 ÁREA DE ABRANGÊNCIA**

Esta monografia abrange diversas áreas, incluindo:

1. Linguagens de programação computacional.
2. Funcionamento de hardware.
3. Filosofia relacionada ao software livre.
4. Distribuições de sistemas operacionais.
5. Arquitetura de sistemas de computação.
6. Modelos de negócios relacionados a software.

Fontes de dados utilizadas incluem documentação oficial de sistemas operacionais, linguagens de programação e hardware, livros especializados em sistemas operacionais, linguagens de programação, segurança e software livre, bem como artigos científicos relacionados a sistemas operacionais.

## **2 DESENVOLVIMENTO**

O desenvolvimento deste trabalho tem como base uma imersão aprofundada nos intrincados domínios do software livre, especialmente no contexto do sistema operacional GNU/Linux. A intenção é desvendar as implicações diretas dessas tecnologias na segurança, privacidade e cultura dos usuários na era digital.

Ao explorar a escolha entre o Software Livre, exemplificado pelo GNU/Linux, e os softwares proprietários, almejamos entender como essa decisão molda de maneira substancial o cenário tecnológico contemporâneo. Nessa análise abrangente, dedicaremos atenção especial ao papel fundamental da transparência e controle proporcionados pelo Software Livre na promoção da segurança cibernética.

A conscientização sobre as divergências entre Software Livre e proprietário também será examinada, destacando como essa compreensão pode influenciar decisões informadas dos usuários em relação à segurança digital. Adicionalmente, abordaremos a maneira como a falta de discernimento técnico pode resultar em escolhas inconscientes, colocando em risco o direito à privacidade dos usuários.

Este desenvolvimento não se limita a uma análise técnica, mas busca desbravar as interconexões profundas entre tecnologia, sociedade e cultura. A intenção é revelar o impacto multifacetado do Software Livre na nossa era digital em constante evolução.

### **2.1 SOFTWARE LIVRE, FILOSOFIA, USO E DIFUSÃO.**

Para exemplificar como se caracteriza um Software Livre, no projeto GNU, primeiramente necessitamos de forma obrigatória decorrer sobre UNIX, um sistema operacional multiusuário e multitarefas, que possui características importantes como arquitetura modular, sistema de arquivos hierárquico, interface de linha de comando e portabilidade, estes princípios de arquitetura influenciaram o movimento de software livre com a criação de um sistema operacional livre.

"O UNIX tornou-se a língua franca na maioria dos departamentos de ciência da computação, e os estudantes logo se tornaram proficientes em sua

manipulação. Em seguida, em 1978, a Bell distribuiu seu programa UUCP (UNIX-to-UNIX copy), permitindo que computadores copiassem arquivos entre si. Com base no UUCP, em 1979, quatro estudantes [...] desenvolveram um programa para comunicação entre computadores UNIX. Uma versão aprimorada desse programa foi distribuída gratuitamente em uma conferência de usuários UNIX em 1980. Isso permitiu a formação de redes de comunicação de computadores, como a Usenet News [...], ampliando significativamente a prática da comunicação de computadores." (Castells, 2001: 13, tradução nossa).<sup>1</sup>

Na década de 80, com o surgimento do projeto GNU, liderado por Richard Stallman, se originando da crescente frustração com a comercialização de Software, onde este possuía o objetivo inicial de desenvolvimento de um sistema operacional completo e comunitário, compatível com Unix. O Projeto possuía incluído uma variedade de utilitários e aplicativos destinados a serem distribuídos gratuitamente a população como "Software Livre", "O projeto para desenvolver o sistema GNU é chamado de 'Projeto GNU'. O Projeto GNU foi concebido em 1983 como uma forma de trazer de volta o espírito cooperativo que prevaleceu na comunidade de computação em dias anteriores." (GNU/project, tradução nossa)<sup>2</sup>

Com o desenvolvimento do projeto em 1984, foi publicado de maneira oficial o manifesto GNU, onde veio a ser argumentado a filosofia do Software Livre, com suas ideias básicas e fundamentalistas, associadas a liberdade e compartilhamento de conhecimento de maneira livre e irrestrita por qualquer parte e argumentação contra as ideias de Software Proprietário. As ideias apresentas podem ser definidas em 4 pilares para a elaboração do movimento, crescimento e acolhimento por toda a sua comunidade.

#### 1. Liberdade de executar o programa para qualquer propósito.

---

<sup>1</sup> UNIX became the *lingua franca* of most computer science departments, and students soon became adept at its manipulation. Then, in 1978 Bell distributed its UUCP program (UNIX-to-UNIX copy) allowing computers to copy files from each other. On the basis of UUCP, in 1979, four students [...] designed a program for communication between UNIX computers. An improved version of this program was distributed freely at a UNIX users' conference in 1980. This allowed the formation of computer communication networks, Usenet News [...] thus considerably broadening the practice of computer communication

<sup>2</sup>The project to develop the GNU system is called the "GNU Project." The GNU Project was conceived in 1983 as a way of bringing back the cooperative spirit that prevailed in the computing community in earlier days

2. Liberdade para modificar e aperfeiçoar programa, seja para adequá-lo para o uso próprio ou pelo aperfeiçoamento e modificações que de algum modo, toda a comunidade usuário desse sistema venha a ser beneficiada com tais mudanças.

3. Liberdade para redistribuir cópias de modo gratuito, distribuição de licenças ou doações.

4. Liberdade de estudar como o programa funciona, ter acesso total ao código-fonte, para que assim possa cumprir as liberdades anteriormente apresentadas.

Desde o seu início como um movimento e comunidade, a sua visibilidade junto ao grande público e massas vem crescendo e evoluindo de forma exponencial, onde a comunidade de usuários e contribuidores, se aglomeram e organizam de forma simples e eficaz, espalhados por todo o mundo, atualmente acredita-se que mais de cem mil programadores e projetistas cooperam de forma ativa e em sua maioria, de forma voluntária em projetos Software Livre, onde estes contribuem para o crescimento de determinados sistemas.

Estima-se que mais de dez milhões de pessoas utilizam de forma diária, aplicações que são distribuídas desta forma. O uso dessas aplicações por inúmeros usuários, pode caracterizar-se por variados motivos seja ele, financeiro, filosófico quando observamos pela óptica dos que acreditam fielmente na filosofia proposta e seguem utilizando apenas softwares que seguem tais mandamentos ou pela óptica de motivos operacional, quando tratamos de organizações que fazem uso de sistemas de tal natureza.

Visando essa lógica apresentada, o software proprietário é normalmente projetado com o objetivo de produzir lucro para a organização e apenas isso, independente de outros sub-motivos que possam vir a ser apresentados e por este motivo, pode estar sujeito a segmentos impostos pelo mercado e sua flutuação, com a inclusão de funcionalidades que em sua maioria não agregam no desenvolvimento do software para o público, seguido normalmente por uma obsolescência programada, falta de suporte para versões antigas e atualizações que necessitam de

um aprimoramento de hardware, com o único objetivo de atender as necessidades e demandas do mercado.

"Software Proprietário é um programa de computador autorizado sob as vantagens seletivas e confiáveis associadas ao detentor dos direitos autorais. O Software Proprietário é criado por um indivíduo ou empresa que possui o privilégio de utilizar hardware existente ou recém-desenvolvido para criar novos programas. A principal característica do Software Proprietário é que seu código-fonte é mantido em segredo do mundo exterior. Portanto, a estrutura interna do software exclusivo não é revelada. As restrições ao software proprietário geralmente são impostas por meio de um documento chamado EULA (Contrato de Licença de Usuário Final), ao qual os usuários são obrigados a concordar antes de usar o software. Ele funciona de maneira semelhante a um contrato por escrito de uso, estabelecendo as condições entre o usuário e o vendedor."(Singh, Amandeep, R.K Bansal, Ph.D, e Neetu Jha. vol. 114, no. 18, março de 2015, tradução nossa)<sup>3</sup>

No Software Livre, todo o seu desenvolvimento e atualizações são pautadas e orientadas pelas necessidades dos usuários e principalmente por aqueles que colaboram de forma ativa em seu desenvolvimento, agindo como uma forma mais democrática, onde programadores de todo mundo introduzem melhorias no sistema, desde a sua versão original até novos braços e versões que podem vir a surgir, atraindo um novo público e desenvolvedores para o sistema.

No caso de um sistema operacional, o código-fonte – as instruções de programação implícitas no sistema – é livre. Qualquer pessoa pode melhorá-lo, transformá-lo, explorá-lo. Porém, essas melhorias, transformações e explorações precisam ser disponibilizadas livremente. Pense Zen. O projeto não pertence a ninguém e pertence a todos. Ao abri-lo a todos, há um aperfeiçoamento rápido e contínuo. Com equipes de colaboradores trabalhando em paralelo, os resultados podem acontecer muito mais depressa e com muito mais sucesso do que se estivessem sendo conduzidos a portas fechadas. (Torvalds 2001, p. 261, tradução nossa).<sup>4</sup>

---

<sup>3</sup>Proprietary Software is machine programming authorized under the selective trustworthy advantages associated with the copyright holder. Proprietary Software is created by an individual or firm who has privileges of utilizing pre-existing or flourishing brand new equipment to make new programming. The principle character of Proprietary Software is that its source code is dependably kept a mystery from the outside world. Accordingly, the inside structure of exclusive programming will be not revealed. The Confinements on restrictive programming are by and large forced through a report called EULA (End-user license agreement) to which clients will be suspected to concur just before utilizing the programming. It meets expectations, simply like a written agreement of good use, ailments between the client and merchant.

Tendo inúmeras vantagens, abrangendo diversas áreas, sociais, financeiras e questões como privacidade.

A) Possuindo um custo social baixo, diferente do software proprietário que é orientado ao benefício do fabricante visando o capital, o software livre visa o benefício do usuário, priorizando as funcionalidades atribuídas ao sistema e o seu uso.

B) Não é refém de tecnologias proprietárias, onde o sistema não se ver refém de organizações, tanto governamentais ou comerciais privadas.

C) Possui a independência de fornecedores, como este não pertence a uma entidade ou ser, que detenha os direitos e propriedade sobre o software, não existe de maneira grande, a possibilidade que determinado produto seja descontinuado, a comunidade usuária do sistema colabora de forma ativa para a sua manutenção e atualização.

D) Não ocorre a obsolescência programada do hardware, pois se ver livre do corrimento da pratica intencional de fabricar e comercializar produtos de forma que estes possuam um tempo de vida limitado, tornando-se inutilizável após um certo período de tempo, juntamente com o lançamento constante de versões quando produto com funcionalidades que não agregam ao usurário.

E) Os sistemas desenvolvidos como Software Livre, são amplamente conhecidos por sua robustez quando se trata da sua segurança, essa qualidade é um resultado direto dos processos de desenvolvimento, como ocorre o processo de revisão e depuração constante do código fonte, por meio de diversas pessoas especializadas na área a localização de erros e falhas de segurança, são identificadas rapidamente e logo corrigidas.

---

<sup>4</sup> In the case of an operating system, the source code - the programming instructions embedded in the system - is free. Anyone can improve it, modify it, explore it. However, these improvements, modifications, and explorations must be made available freely. Think Zen. The project belongs to no one and belongs to everyone. By opening it up to everyone, there is rapid and continuous improvement. With teams of collaborators working in parallel, results can happen much more quickly and with much more success than if they were being conducted behind closed doors.

### 2.1.1 CARACTERIZAÇÃO E DISTRIBUIÇÃO DE SOFTWARE LIVRE

Em tese, a definição de todos os processos presentes na distribuição e desenvolvimento de um sistema Software Livre, proposto pelo Projeto GNU, podendo ter sua definição em pontos de ciclos de desenvolvimento e vida de um software, se iniciando com o autor escrevendo a primeira versão do programa e publicando o seu código fonte, com é concedido o acesso a terceiros, para que seja possível a correção de erros básicos ou críticos que possam vir a ser visualizados, sugerir implementações e melhorias que podem vim a ser aceitas pelo autor e implementadas em uma nova versão, atualizando a sua documentação de maneira constante para o uso de futuros colaboradores.

“Existe uma razão específica pela qual a liberdade de modificar é crucial para a documentação de software livre. Quando as pessoas exercem seu direito de modificar o software e adicionam ou alteram suas funcionalidades, se forem conscientes, também modificaram o manual para poder fornecer uma documentação precisa e utilizável junto com o programa modificado.” (GNU/project, 2023, tradução nossa).<sup>5</sup>

Detalhando ainda mais a sua definição proposta pelo Projeto GNU, as etapas de desenvolvimento de software livre e sua distribuição, para o reconhecimento e compartilhamento por toda a comunidade, podendo ser definidas como:

- a) Pacote: consiste em um aglomerado de arquivos que foram distribuídos pelo autor do projeto, onde este é composto de informações, principalmente sobre o código fonte do programa, documentação e possivelmente uma versão compilada e executável doo programa.
- b) Autor: a pessoa, grupo ou organização que compõem os responsáveis pela fabricação da versão original do autor.

---

<sup>5</sup> Is a particular reason why the freedom to modify is crucial for documentation for free software. When people exercise their right to modify the software, and add or change its features, if they are conscientious they will change the manual too so they can provide accurate and usable documentation with the modified program.



- c) Versão original: Sendo o programa que foi originalmente distribuído pelo autor sem a existência da modificação externa, produzida por terceiros.
- d) Trabalho derivado: Sendo o resultado das modificações introduzidas de forma externa através dos processos de colaboração do Software Livre, sobre a versão original, modificações essas que alteram ou adicionam funcionalidades do programa e correção de erros eventuais.
- e) Licença: Termo que habilita os direitos sobre o software, definido pelo autor.
- f) Código fonte: Versão produzida pelo autor, declarando o comportamento do programa, suas funções e nuances que este possui.
- g) Código executável: Uma versão do programa que se encontra de forma binária, ideal para ser executada pelo usuário para que o mesmo possa usufruir das funcionalidade e propósitos do software.
- h) Distribuição: A distribuição consiste em todos os itens citados acima, devidamente organizados e armazenados de forma eficiente, sendo em um meio físico ou virtual.

Com estes pontos, a popularidade e conseqüentemente a atração de novos usuários para o software cresce de maneira exponencial, onde novos programadores passam a colaborar diretamente no desenvolvimento, atribuindo qualidade, confiança e segurança ao produto. Essa crescente em sua qualidade deve-se especialmente ao grande número de desenvolvedores que estão diretamente envolvidos em todo o processo, já que uma vez publicado, dependendo da visibilidade e popularidade do programa, todos esses processos se dão de maneira rápida e efetiva.

### **2.1.2 LICENÇAS DE USO, GNU GPL E MIT.**

Tendo o copyleft como uma prática de licenciamento de software quando se trata de termos específicos de distribuição, compartilhamento e propriedade intelectual do software. Possuindo o objetivo claro e central de retirar as barreiras legais e exigir que todas as versões, desde a original até os trabalhos derivados

possuam tais pontos protegidos e permaneçam livres e abertos a toda a comunidade, usando norma e legislação como um pilar para garantia das quatro liberdades apresentadas anteriormente.

“No Projeto GNU, nossa intenção é dar a *todos* os usuários a liberdade de redistribuir e modificar software GNU. Se intermediários pudessem retirar as liberdades, nós poderíamos ter muitos usuários, mas não teríamos liberdade. Por isso, em vez de colocar o GNU em domínio público, nós o tornamos “copyleft”. Copyleft diz que qualquer um que redistribui o software, com ou sem modificações, tem que passar adiante as liberdades de fazer novas cópias e modificá-las. O Copyleft garante que todos os usuários tenham liberdade” (GNU project, 2022, tradução nossa).<sup>6</sup>

Para a Free Software Foundation, trata-se de liberdade e não preço sobre o produto, com isso a licença apropriada e correta, segundo estes, se torna a Licença Pública Geral GNU (GNU GPL), para ser utilizada e incluída em manuais, código-fonte e documentação, publicada em 1989 e desenhada com o objetivo de ser facilmente aplicada ao programa, onde para usufruir da sua utilização e termos é necessário apenas incluir notas que se refiram a GPL.

“As licenças para a maioria dos softwares e outros trabalhos práticos são projetadas para tirar sua liberdade de compartilhar e alterar os trabalhos. Por outro lado, a Licença Pública Geral GNU destina-se a garantir a sua liberdade de compartilhar e alterar todas as versões de um programa – para garantir que ele permaneça software livre para todos os seus usuários. Nós, a Free Software Foundation, usamos a Licença Pública Geral GNU para a maioria dos nossos softwares; aplica-se também a qualquer outro trabalho divulgado desta forma pelos seus autores. Você também pode aplicá-lo aos seus programas (GNU project, 2023, tradução nossa).<sup>7</sup>

Tendo uma filosofia similar a utilização da GNU(GPL), a licença MIT abrange diversas áreas que trazem consigo o poder de permitir que o software seja tratado sem restrições para modificações, distribuição e uso. Possuindo um alto

---

<sup>6</sup> In the GNU Project, our intention is to give all users the freedom to redistribute and modify GNU software. If intermediaries could take away these freedoms, we could have many users, but we would not have freedom. That's why, instead of placing GNU in the public domain, we made it "copyleft." Copyleft says that anyone who redistributes the software, with or without modifications, must pass on the freedoms to make new copies and modify them. Copyleft ensures that all users have freedom

<sup>7</sup>Most software and other practical works' licenses are designed to take away your freedom to share and modify those works. On the other hand, the GNU General Public License is intended to ensure your freedom to share and modify all versions of a program - to ensure that it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it also applies to any other work released in this way by its authors. You can also apply it to your programs.

valor permissivo, trazendo consigo diversos pontos de extrema importância, como a redistribuição livre, aviso de direitos autorais e licença

Não possuindo garantia e compatibilidade com outras licenças de software livre onde, Permissão é concedida, gratuitamente, a qualquer pessoa que obtenha uma cópia deste software e dos arquivos de documentação associados (o 'Software'), para lidar com o Software sem restrições, incluindo, entre outras coisas, os direitos de usar, copiar, modificar," ( St. Laurent, Andrew M. Entendimento de Licenciamento de Software Livre e de Código Aberto. 21 de setembro de 2004, p. 22, tradução nossa)<sup>8</sup>

Sua principal característica acaba sendo a sua a sua simplicidade, que permite uma colaboração concisa e bastante flexível em seu uso, sem cláusulas que podem acarretar dificuldade ou empecilhos jurídicos, como a sua ausência de garantia e responsabilidade, protegendo diretamente os detentores dos direitos autorais dos projetos disponíveis sobre questões legais e processos judiciais que podem vir a surgir envolvendo o uso do software por terceiro, possuindo estes determinados pontos e ideais, essa licença acaba sendo amplamente recomendado por instituições que pregam esta filosofia, como a Free Software Foundation.

### **2.1.3 SOFTWARE LIVRE, SEGURANÇA E SUA PRIVACIDADE**

Atualmente, estamos inseridos cada vez mais em um ambiente totalmente digital e integrado, que podemos até caracterizar como ameaçador, onde nossa segurança e integridade digital passa a ser uma prioridade, assim como a nossa segurança física, diversos serviços se destacam por oferecer soluções de destaque nesta área, principalmente soluções que empregam o Software Livre.

Toda a filosofia apresentada e debatida anteriormente desempenha um papel fundamental na segurança dos sistemas que optam pelo uso da licença GNU/GPL ou mesmo são apenas sistemas de código aberto. Considerando esses dois espectros, a transparência do código é um dos fatores principais para que

---

<sup>8</sup> Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify,

esses tipos de software sejam tão eficazes quando se trata da segurança do sistema e dos usuários. Especialistas podem examinar todas as nuances do código e identificar vulnerabilidades e problemas ocultos, conforme o projeto GNU.

“Isso não implica que o software livre seja imune a erros ou vulnerabilidades. A diferença entre o software livre e o software proprietário nesse aspecto está no tratamento dos erros: os usuários de software livre podem estudar o programa e/ou corrigir os erros que encontrarem, frequentemente em comunidades, pois podem compartilhar o programa, enquanto os usuários de programas proprietários são obrigados a depender do desenvolvedor do programa para correções” (GNU project 2023, tradução nossa)<sup>9</sup>

Ocorrendo uma alta taxa de correção e velocidade razoavelmente rápida, uma vez que essa vulnerabilidade é identificada, a comunidade que utiliza esse software e necessita que ele seja o mais seguro possível tenta, de maneira rápida e eficiente, corrigir todos os problemas encontrados e distribuir a atualização corrigida para os demais usuários

A distinção entre Software Livre e Open Source, embora frequentemente utilizados como sinônimos, reflete nuances significativas na filosofia e nas liberdades concedidas aos usuários. O Software Livre, fundamentado nas ideias de Richard Stallman e na Free Software Foundation, prioriza a liberdade dos usuários, enfatizando a capacidade de executar, modificar e redistribuir o código-fonte. Isso visa garantir não apenas a utilidade imediata, mas também a autonomia do usuário sobre o software.

#### **2.1.4 OPEN SOURCE**

Por outro lado, o termo Open Source, introduzido posteriormente por figuras como Eric S. Raymond, foca mais na acessibilidade ao código-fonte para aprimoramentos colaborativos. Embora compartilhe alguns princípios com o Software Livre, a abordagem Open Source não enfatiza estritamente as liberdades

---

<sup>9</sup> This doesn't imply that free software is immune to bugs or insecurities. The difference between free and proprietary software in this respect is the handling of the bugs: free software users are able to study the program and/or fix the bugs they find, often in communities as they are able to share the program, while proprietary program users are forced to rely on the program's developer for fixes.

éticas do usuário. Esse modelo concede mais flexibilidade para a colaboração, muitas vezes em ambientes empresariais, sem necessariamente impor a mesma ênfase nos valores éticos defendidos pelo movimento do Software Livre.

Portanto, a principal diferença reside nas prioridades: o Software Livre enfatiza a liberdade do usuário, enquanto o Open Source valoriza a abertura do código para benefícios práticos, mesmo que as liberdades do usuário não sejam tão rigorosamente preservadas.

#### **2.1.4 CONSUMO E MONETIZAÇÃO**

Observa-se que Software Livre não significa necessariamente que este seja gratuito ou que não possa se obter lucro, seja por meio de doações, venda, cobrança de licença e uso, toda a sua filosofia se trata primordialmente da distribuição de conhecimento e livre acesso à informação, com o não impedimento sobre os princípios e direitos básicos, seguindo a sua filosofia.

Conforme evidenciado ao longo do texto, o Software Proprietário prioriza, acima de tudo, o lucro em detrimento das necessidades reais do usuário. Vale ressaltar que essa abordagem não implica necessariamente em uma execução deficiente de suas funções, como afirmou Richard Stallman “O software proprietário é um esquema para tornar os ricos mais ricos à custa de usuários insatisfeitos” (Richard Stallman, tradução própria)<sup>10</sup>, O ponto crítico aqui discutido não se relaciona diretamente à qualidade intrínseca do software, mas sim à natureza do código-fonte fechado, sem a possibilidade de auditoria, estudo e disseminação do conhecimento nele contido. É crucial compreender que o conhecimento é uma entidade destinada a ser compartilhada e difundida, conforme expresso por Moshe Dayan: “A liberdade é o oxigênio da alma” (My Life, Moshe Dayan, 1976).

O impacto do Software Livre no meio acadêmico, especialmente entre pesquisadores brasileiros, é notável diante da significativa desvalorização do setor. Diante dos recursos precários fornecidos por órgãos governamentais e da ausência

---

<sup>10</sup>Proprietary software is a scheme to make the rich richer, at the expense of dissatisfied users.

de incentivo financeiro adequado para pesquisas em desenvolvimento tecnológico e científico, surge a questão: como pesquisadores ou instituições podem viabilizar soluções elaboradas, eficazes e sofisticadas que, muitas vezes, envolvem custos elevados sem o suporte adequado?

“O software de pesquisa inclui código-fonte, algoritmos, scripts, fluxos de trabalho computacionais e executáveis criados durante o processo de pesquisa ou com pesquisa como seu propósito. Ele é tipicamente específico para determinado domínio. A pesquisa também emprega diretamente fluxos de trabalho computacionais e de processamento de dados, que podem combinar várias peças de software e outros recursos, além de blocos de notas executáveis autocontidos que integram dados, métodos analíticos, procedimentos, código, resultados, visualizações e narrativas. O software de pesquisa pode ser tratado como um recurso do sistema operacional que é gerenciado junto com outras partes da pesquisa, mas ainda assim precisa ser distinguido como uma entidade separada de um tipo especial. Além disso, todos os seus elementos podem ser agregados em um produto de pesquisa autônomo do sistema operacional, especialmente se for utilizado por vários projetos de pesquisa” (NI\$OS, Software e Software Livre e de Código aberto na Ciência Aberta, Fev 2023, tradução nossa)<sup>11</sup>

Verificaremos como exemplo um software de natureza totalmente proprietária, chamado Wolfram Mathematica, este trabalha realizando operações matemáticas de grande complexidade, atualmente custando o valor de uso mensal de três mil dólares americanos, como demonstrado na figura 1, logo observamos que ele se trata de um software que não é acessível a toda uma comunidade e por se tratar de um software proprietário, não possuímos o conhecimento da sua lógica matemática utilizada na realização dos cálculos, e com isso a auditoria destes resultados se torna limitada, prejudicando o processo científico.

---





<sup>11</sup> Research software includes source code, algorithms, scripts, computational workflows and executables created during the research process or with research as its purpose. It is typically domain-specific. Research also directly employs computational and data processing workflows, which may combine several pieces of software and other resources, and self-contained executable notebooks that integrate data, analytical methods, procedures, code, results, visualizations and narrative. Research software can be treated as an OS resource that is managed together with other parts of research but it still needs to be distinguished as a separate entity of a special type. Furthermore, all its elements may be aggregated in a standalone OS research product, especially if it is to be used by several research projects.

## Figura 1 - Preço do Software Mathematica

Mathematica Pricing >  
**Industry**  
 Mathematica licensing for professional<sup>®</sup> and personal use.  
 Already a customer? [Renew service or upgrade.](#)  
 Find out if you have [access through your organization](#) >

Individuals Groups & Organizations

Brazil

| Standard Cloud   | Standard Desktop   | MOST POPULAR Standard Desktop + Cloud  | Mathematica Enterprise Edition   |
|--|--|--|--|
| <br><b>Mathematica   Online</b><br>• Upgrades<br>• Standard support<br><br>\$1,660 /year<br><input type="button" value="Subscribe"/><br>\$200 /month <input type="button" value="Subscribe"/> | <br><b>Mathematica   Desktop</b><br>with 1 year of <b>Premier Service</b><br>• Upgrades<br>• Standard support<br>• Personal-computer license<br><br>\$3,320<br><input type="button" value="Buy Now"/><br>\$1,660 /year <input type="button" value="Subscribe"/> | <br><b>Mathematica   Desktop</b><br>with 1 year of <b>Premier Service Plus</b><br>• Mathematica Online<br>• Upgrades<br>• Standard support<br>• Personal-computer license<br><br>\$3,652<br><input type="button" value="Buy Now"/><br>\$1,826 /year <input type="button" value="Subscribe"/> | <br><b>Mathematica   Desktop</b><br>with 1 year of <b>Premier Service Plus</b><br>• Mathematica Online<br>• Upgrades<br>• 24/7 Enterprise support<br>• Personal-computer license<br>• Enterprise-level development<br><br>\$9,300<br><input type="button" value="Buy Now"/> |

Fonte - [Wolfram Computation Meets Knowledge](#)

Logo que uma vez disponíveis tais regras e lógicas de código, isso permitiria que outras empresas criassem soluções equiparadas ou melhores, gerando uma competição, que afetaria diretamente a empresa de forma negativa e de forma positiva o meio acadêmico.

Observa-se que uma das principais soluções é recorrer diretamente a softwares que se caracterizam como Software Livre, onde a disponibilização do seu código fonte, permite que o usuário tenha conhecimento exato sobre o que o sistema está realizando e seu comportamento, sua lógica e caminhos, eliminando suspeitas como no caso do Mathematica, a confiabilidade acaba colocada novamente ao todo, onde resultados podem ser auditados, testados e contestados de forma direta e efetiva, elevando consideravelmente o nível de rigorosidade.

Entende-se que uma das maneiras mais comuns de se monetizar um Software Livre, é oferecendo suporte e serviços, podendo ser citada a Red Hat, uma empresa de tecnologia, conhecida pela distribuição Red Hat Enterprise Linux

(RHEL), uma das mais populares distribuições Linux que temos atualmente, onde esta funciona oferecendo suporte técnico para os detentores do sistema, onde usaremos este para exemplificar um determinado modelo de monetização.

A empresa oferece várias camadas de suporte técnico, onde por meio destas é possível para a empresa escolher o nível de assistência que mais se adequa a realidade da sua necessidade, oferecendo planos, básicos, padrão, premium e suporte empresarial, como pode ser observado na figura 2.

**Figura 2 - Planos de Suporte Red Hat**

| SELF-SUPPORT (1 YEAR)  | STANDARD (1 YEAR)   | PREMIUM (1 YEAR)  |
|--|---|---|
| <ul style="list-style-type: none"> <li>Does not include Red Hat customer support.</li> <li>Does not include Red Hat Enterprise Linux Atomic Host.</li> <li>Can only be deployed on physical systems.</li> <li>Cannot be stacked with other subscriptions.</li> <li>Is not intended for production environments.</li> </ul> | <ul style="list-style-type: none"> <li>For all support cases, standard business hours of coverage are either 9 a.m. to 6 p.m. (0900-1800) local time for North America or 9 a.m. to 5 p.m. (0900-1700) for outside North America. Standard business hours do not include weekends or local public holidays.</li> <li>Web and phone support channels.</li> <li>Unlimited support cases.</li> </ul> | <ul style="list-style-type: none"> <li>24x7 for severity 1 and 2 cases.</li> <li>Standard business hours of coverage for severity 3 and 4 cases are either 9 a.m. to 6 p.m. (0900-1800) local time for North America or 9 a.m. to 5 p.m. (0900-1700) for outside North America. Standard business hours do not include weekends or local public holidays.</li> <li>Web and phone support channels.</li> <li>Unlimited support cases.</li> </ul> |
| <p><input type="checkbox"/> Satellite (bundle) US\$699</p> <p><b>Total US\$349</b></p>   | <p><b>Select an Add-On</b></p> <p><input type="checkbox"/> Satellite (bundle) US\$1,149</p> <p><input type="checkbox"/> High Availability US\$399</p> <p><input type="checkbox"/> Resilient Storage US\$799</p> <p><input type="checkbox"/> Extended Update Support US\$249</p> <p><b>Total US\$799</b></p>   | <p><b>Select an Add-On</b></p> <p><input type="checkbox"/> Satellite (bundle) US\$1,649</p> <p><input type="checkbox"/> High Availability US\$399</p> <p><input type="checkbox"/> Resilient Storage US\$799</p> <p><b>Total US\$1,299</b></p>   |
| <b>Add to cart</b>   | <b>Add to cart</b>  | <b>Add to cart</b>  |

Fonte: [Red Hat Services](#)

Com isso é garantido que as empresas tenham acesso a atualizações de segurança, suporte especializado e correção de erros, além de oferecer juntamente serviços de consultoria, como migração, integração, possibilitando a integração facilitada de outras tecnologias a empresa, avaliação, planejamento e desenvolvimento de soluções totalmente personalizadas.

Existem diversos outros modelos que são passíveis de renda para o desenvolvedor ou empresa, pode-se destacar os principais, onde este se assemelham a um plano de negócio de um software proprietário, com a principal



diferença sendo que estes modelos estão servido a um software livre, onde mesmo que possuam tais formas de pagamento ou financiamento, o seu código fonte continua sendo livre para todo o público, com isso destacamos.

- A) Doações e financiamento coletivo, muito usado por desenvolvedores individuais
- B) Planos de assinatura
- C) Venda de hardware e serviços relacionados, como Raspberry pi, um hardware que conta com uma distribuição GNU/Linux pré instalada.
- D) Crowdsourcing de Recursos, onde este funciona através do trabalho voluntário de desenvolvedores, permitindo que o projeto tenha um custo reduzido e mão de obra abrangente.

## **2.2 GNU/LINUX, SEU CRESCIMENTO, ACEITAÇÃO E FUNCIONAMENTO**

O GNU/Linux, sistema operacional de código aberto, destaca-se pelo seu notável crescimento, ampla aceitação e sólido funcionamento. Desde o seu surgimento, liderado por Linus Torvalds, o Linux tornou-se um caso de sucesso no universo do Software Livre.

Sua aceitação crescente é impulsionada pela comunidade global de desenvolvedores que colaboram para aprimorar continuamente o sistema. A robustez do GNU/Linux é evidenciada pela sua eficiência como sistema operacional, garantindo estabilidade e segurança aos usuários. Este tópico explora o fenômeno do GNU/Linux, abordando seu crescimento exponencial, a aceitação generalizada e a operacionalidade que o transformaram em uma referência no mundo da computação.

### **2.2.1 ENTENDO O SOFTWARE E SUA ACEITAÇÃO**

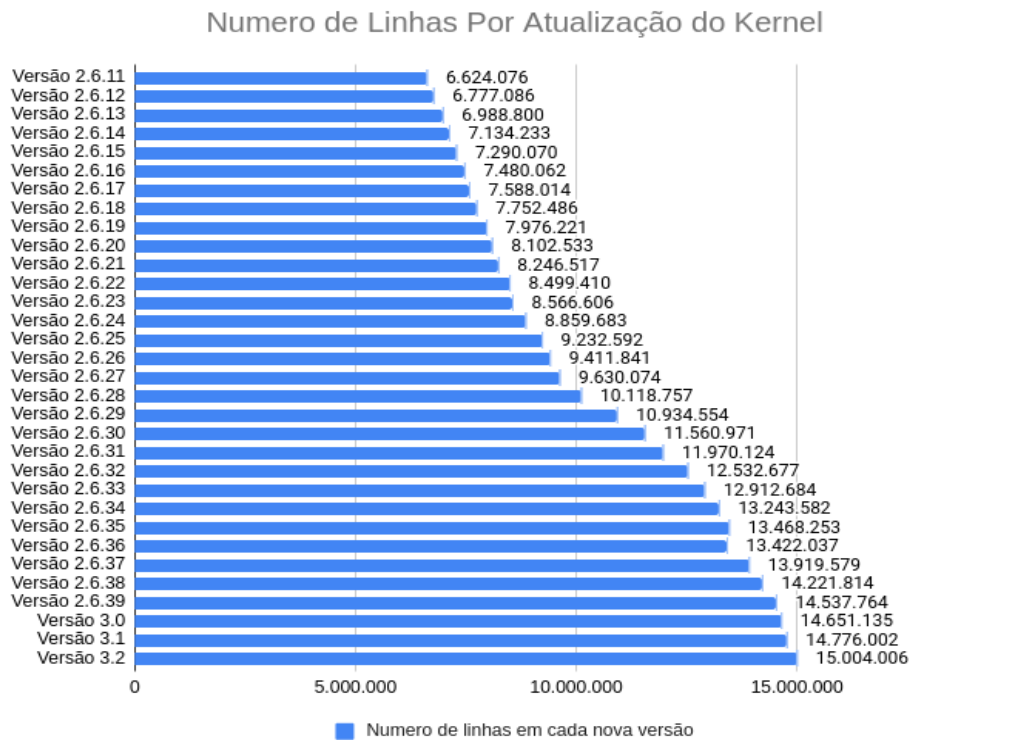
Sendo um dos principais Sistemas Operacionais existentes atualmente, o sistema GNU/Linux é uma combinação amistosa do Kernel Linux, com as ferramentas e utilitários do projeto GNU, com sua constante influência, aceitação e prestígio, adjetivos estes que podem ser atribuídos a um leque de combinações e características, como seus valores filosóficos e rigorosidade técnica. Para entender o motivo desse sistema ser amplamente dissipado é necessário examinar suas características mais fundamentais e entender a sua diferenciação dos outros sistemas existentes.

O Kernel Linux inicialmente desenvolvido por Linus Torvalds em 1991, estudante da universidade de Helsinque, construído principalmente utilizando a linguagem de computação C, já que, essa sendo uma linguagem de alto nível, permite a execução de tarefas extremamente importantes para a criação de um Kernel, como a alocação de memória e outras funções presentes na linguagem. O sistema foi inicialmente pensado e desenvolvido para ser utilizado por pessoas que se identificavam como entusiastas em computadores pessoais.

Observa-se que em sua distribuição original, o próprio usuário deveria procurar programas que empregavam as funcionalidades desejadas a este, compilar e realizar as configurações necessárias para que o sistema se adaptasse aos seus anseios e executasse da maneira mais cômoda, algo totalmente diferente do que encontramos com as distribuições atuais, onde algumas oferecem um leque de configurações e programas já previamente instalados, tornando todo o processo de uso mais simplificado e convidativo.

Com outras distribuições prezando pela “instabilidade” e a total personalização, onde estas necessitam um conhecimento mais avançado sobre o sistema e gerenciamento de pacotes, para que o usuário deixe a sua máquina de maneira pessoal, instalando os recursos necessários seguindo seu uso.

Com seu constante crescimento tanto em ambientes domésticos, mas principalmente em servidores e empresas, as atualizações se mantêm de forma constante e consistente, o seu Kernel se mantêm em manutenção, ativa, crescendo em funcionalidade e segurança, como o Gráfico 1 apresentando uma visão do número de linhas adicionadas ao Kernel do Sistema Operacional GNU/Linux a cada nova versão até a versão 3.1.

**Gráfico 1 - Número de Linhas por Atualização do Kernel**

**Fonte** - Pigdom, 2012.

A importância do Software Livre para o crescimento do sistema, ocorre de forma direta, principalmente com toda a comunidade de usuários e desenvolvedores que trabalham e usam ativamente o sistema, alguns pesquisadores como Francisco Coelho dos Santos, defendem este ponto como ideia para explicar o diferencial do Linux como um Sistema Operacional comparado a outros modelos existentes, o fato de outros modelos não passarem a formação de um lado social, um senso de pertencimento e comunidade algo que é impregnado em toda esta comunidade.

Inúmeros pontos são capazes de explicar o seu crescimento e aceitação como um Sistema Operacional. Sua natureza modular, permite sua alta customização e adaptação a diferentes necessidades, capacidades que buscam atender as necessidades individuais ou organizacionais, é uma das principais razões

para o sistema ser amplamente aceito em diferentes setores, empresariais, domésticos e governamentais e pela comunidade de Software Livre, com isso podemos citar.

**Colaboração e Contribuição:** Com o seu Código-fonte livre, o sistema sempre permitiu que a comunidade global de desenvolvedores e entusiastas contribuíssem de forma direta na correção de erros, atualizações de recursos e melhorias constantes, trazendo com isso um sentimento de pertencimento sobre o sistema.

**Customização e Adaptação:** Com a sua licença de Código Livre a customização é uns dos principais pontos visto pelos usuários do Linux, com distribuições como Arch Linux, disponibilizando inúmeras maneiras de “Construir” seu próprio sistema, configurá-lo de maneira único, e oferecendo suporte direto através da sua extensa e detalhada documentação.

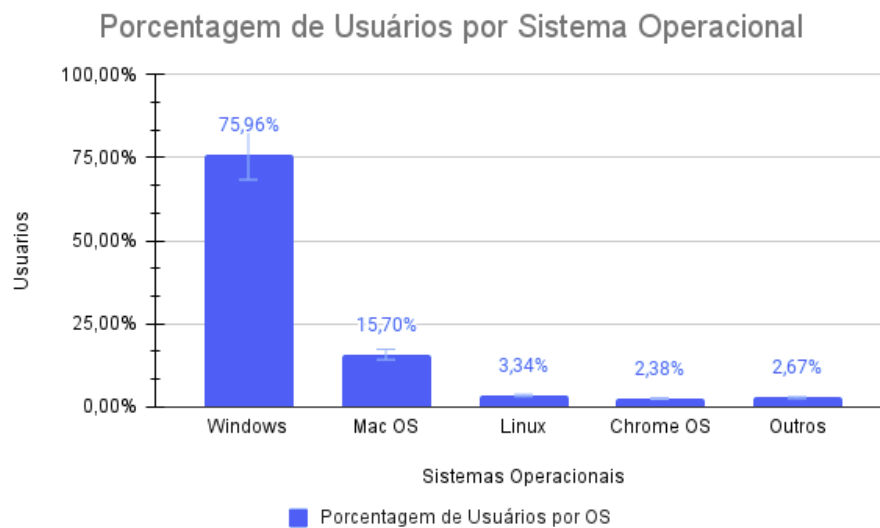
Isso gera uma liberdade, liberdade esta que pode ser explorada, por empresas, usuários e times, que personalizam e entregam um sistema de maneira customizada para as necessidades do usuário leigos, através de várias distribuições presentes no mercado da tecnologia.

**Inovação Rápida:** A natureza do Linux com sua extensa comunidade de usuário e seu código totalmente disponível para o público, permite que a inovação ocorra de maneira rápida, inteligente e abrangente, onde distribuições podem optar por estabilidade e liberar recursos novos, depois que eles forem testados e corrigidos pela comunidade e outras optam por ser instáveis, trazendo atualizações de forma constante, onde qualquer pessoa com o conhecimento necessário pode contribuir de maneira ativa para a melhoria e manutenção do sistema.

A cultura de compartilhamento de informações e conhecimento, gera uma crescente linear visível de aprendizado, criando ambientes em que tanto um iniciante, querendo descobrir este sistema e entender seu funcionamento e uso, assim como um especialista, consigam se sentir aceitos e incentivados a contribuir para o crescimento deste sistema, criando um elo forte, entre usuário e distribuições e suas ferramentas ofertadas, moldando todo um cenário tecnológico global, se

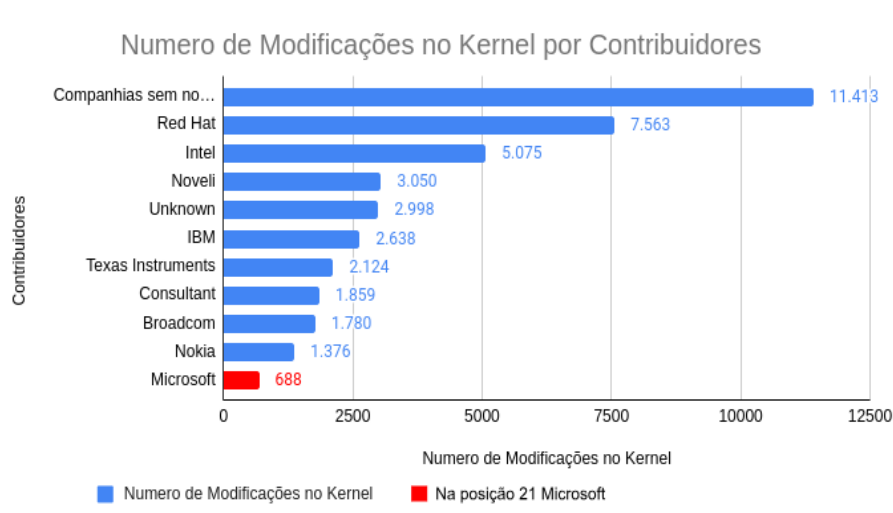
tornando amplamente usado como demonstrado no gráfico 2, comparando o número de usuários por sistema operacional.

**Gráfico 2** - Porcentagem Aproximada de Usuários por Sistema Operacional



**Fonte:** Pingdom, 2012

O sistema desempenha um papel crucial em operações de negócio em todo mundo, dependentes da sua estabilidade, confiança e flexibilidade gerada, acaba se tornando a escolha óbvia e para empresas de todos os tamanhos, podendo ser aplicado desde servidores até soluções em nuvem, com isso a colaboração de grandes empresas, para a manutenção do sistema é constante, como a Microsoft, mesmo possuindo seu próprio sistema operacional (Windows), ajuda na manutenção do Kernel Linux como exibido no Gráfico 3, com o número de modificações feita no kernel por grandes empresas.

**Gráfico 3 - Número de Modificações no Kernel por Contribuidores.**

Fonte - Pingdom, 2012

## 2.2.2 IMPACTOS SOCIAIS, CULTURAIS E SOCIOLÓGICOS.

Observa-se que o sistema GNU/Linux ultrapassa o seu mero papel como um completo sistema operacional existente nos dias atuais, acaba sendo amplamente mais profundo, influenciando a nossa sociedade cultural e digital, e o relacionamento interpessoal dos seres humanos com as tecnologias empregadas por este, com isso a elucidação do seu papel na promoção, colaboração e inclusão, não se dar apenas em avanços tecnológicos.

Tendo como uma das características mais marcantes sua transparência e cultura de colaboração promovida de forma totalmente orgânica por desenvolvedores ao redor do globo, essa diversidade é uma característica fundamental, pois proporciona um ambiente onde as contribuições são amplamente valorizadas e cada qual possui o seu devido mérito.

Como exemplificado, a sua natureza permite que uma variedade de pessoas colaborem independente da sua origem ou nível de experiência em desenvolvimento e outras áreas que fundamentam o que conhecemos como um sistema operacional, como seu Kernel, design ou segurança. Seu modelo elegante

de desenvolvimento descentralizado, tem influenciado diversos projetos e empresas a adotar abordagens semelhantes, essa cultura de colaboração não apenas leva a uma tecnologia mais robusta, avançada e exemplar que acaba sendo um padrão para o mundo, mas também promove a diversidade na indústria da tecnologia.

Com a sua disponibilidade afetando diretamente de forma benéfica um dos princípios estipulados por Richard M. Stallman. Onde onde este evento acaba por impactar de maneira significativa estudantes, professores e pesquisadores da área, que pode de forma simples acessar, modificar e estudar o código presente nas distribuições formadas, podendo assim criar a sua própria ou acrescentar melhorias, garantindo uma continuidade no desenvolvimento, oportunidade de aprendizado de forma teórica e prática, juntamente com a sua experimentação. Ocorrendo a democratização do acesso ao conhecimento tecnológico e permite que as pessoas compreendam não apenas como uma determinada tecnologia funciona, mas também como ela pode ser moldada de acordo com as necessidades dos indivíduos.

Tendo o seu sucesso inspirado o crescimento de inúmeros projetos em diferentes áreas, além dos sistemas operacionais e suas inúmeras distribuições que atendem diversos nichos específicos para uma variedade de usuários. Projetos como Apache, FireFox e MySQL prosperam juntamente com a iniciativas de hardware de código aberto, como projetos com arduinos e movimento Open Compute, que tem como iniciativa projeta hardware mais eficiente e que ao mesmo tempo seja de alguma forma mais acessível, personalizado e ecológico. Estes são exemplos de como toda essa filosofia pode se estender além do software.

Sua descentralização reflete em tendências amplas em direção a autonomia, sendo visível que sistemas com a mesma raiz ideologia, que optam em seguir tais rumos vivenciam uma maior liberdade, o sistema GNU/Linux acaba não sendo apenas um sistema operacional, mas sim uma ideologia e cultura, um exemplo inspirador do poder de colaboração, aberturas das portas tecnológicas e uma bandeira de luta.

### **2.2.3 SEU KERNEL, FUNCIONAMENTO E PROCESSOS**



Para entendermos o seu funcionamento e sua importância, necessitamos primeiramente entender os conceitos básicos de Hardware e como ocorre a comunicação do Kernel com ele. Sabemos que um Sistema Operacional necessita trabalhar efetivamente junto ao Hardware, onde a sua implementação de forma correta acaba sendo imperceptível para o usuário, como exemplificado por Bovet e Cesati, "O kernel é o coração do sistema operacional. Ele é responsável por tudo, desde a inicialização do sistema até a execução de aplicações." (BOVET; CESATI, 2005, p. 2, tradução nossa)<sup>12</sup>

Sem a existência de um Software, o computador não passa apenas de um gigante peso de papel sem grande utilidade e usabilidade. Podemos caracterizar um sistema operacional como uma grande coleção de programas de sistemas, que permitem ao usuário final executar operações desejadas, o próprio GNU/Linux é composto por variadas partes, onde vários programas que em conjunto e se integram para formar o Sistema Operacional sendo o Kernel se configurando apenas como uma destas partes.

Tendo funções primordiais, como Gerenciamento de memória, onde ocorre todo o monitoramento da quantidade de memória que está sendo utilizada para o armazenamento de dados, em variados tipos de ambientes. Atuando também como intérprete entre os processos executados e o hardware, através de Drives do dispositivo, e a sua segurança, Torvalds descreve, "O Kernel Linux é a parte mais fundamental de um sistema operacional", gerenciando recursos de hardware, escalonando processos e proporcionando uma base sólida para a execução de programas (Torvalds, 2001, p. 12, tradução nossa)<sup>13</sup>

Com toda a comunicação entre o Sistema Operacional e o Hardware sendo efetuada através solicitações de processos, se dá a execução das ações e atividades deferidas por meio do código previamente definido pelo desenvolvedor do software, conversando diretamente com a CPU, que se caracteriza como o coração

---

<sup>12</sup> The kernel is the heart of the operating system. It is responsible for everything, from system boot-up to running applications.

<sup>13</sup> The Linux Kernel is the most fundamental part of an operating system, managing hardware resources, scheduling processes, and providing a solid foundation for running programs.

de qualquer sistema computacional, onde o microprocessador calcula e realiza operações de forma lógica e objetiva, gerencia o fluxo de dados, lendo e executando funções da memória, de acordo com Hennessy e Patterson "A unidade central de processamento (CPU) é o cérebro do computador. É responsável por executar as instruções do programa de computador." (Hennessy & Patterson, 1989, p. 1, tradução nossa)<sup>14</sup>

Sua estrutura de dados é formada por um conjunto destas, o que torna possível a realização de gerenciamento de recursos do sistema e união entre os componentes do kernel. Com isso podemos destacar inicialmente estas elencadas abaixo.

- 1) Listas Encadeadas, onde permite que os elementos sejam removidos e adicionados de forma simples e efetiva, usada principalmente para gerenciamento de dispositivos e listas de tarefas.
- 2) Árvores binárias de pesquisas são utilizadas para otimizar a pesquisa de elementos em um grande conjunto de dados, para garantir que as operações vão ser realizadas de forma eficiente já que em tese, o Kernel e todo o sistema operacional são um conjunto de lista de arquivos, que trazem configurações responsáveis por determinados pontos não passam de uma lista de arquivos que dentro deste trazem configurações responsáveis por determinados pontos, e com isso o sistema é construído de forma que um humano consiga utilizar de forma elucidada.
- 3) hash table é usada para mapear chaves e valores e com isso permitir o acesso eficiente e rápido em estruturas como cache e alocação de recursos, como veremos nos próximos tópicos com a utilização da linguagem C para a realização de tais atividades.

#### **2.2.4, CPU, SEU FUNCIONAMENTO E PRINCÍPIOS BÁSICOS EMPREGADOS AO GNU/LINUX**

---

<sup>14</sup>The central processing unit (CPU) is the brain of the computer. It is responsible for carrying out the instructions of the computer program.

Sabe-se que a execução de um processador se dá pela governança de um relógio externo, onde esse relógio do sistema gera pulsos regulares para o processador e em cada pulso o processador realiza algum trabalho programado previamente, um processador poderia executar uma instrução a cada pulso do relógio, onde como exemplo um processador de 100MHz, receberá 100.000.000 pulsos a cada segundo. Segundo Rusling, "A CPU, ou seja, o microprocessador, é o coração de qualquer sistema de computador. O microprocessador realiza cálculos, realiza operações lógicas e gerencia fluxos de dados lendo instruções da memória e, em seguida, executando-as." (RUSLING, David A. The Linux Kernel. 1996. p. 8, tradução nossa)<sup>15</sup>

As instruções executadas pelo Processador se caracterizam de maneira muito simples, onde se podemos escrever da seguinte forma "Leia o conteúdo da memória na posição 1 no registrador B", Os Registradores são o armazenamento interno do microprocessador, usado para armazenar dados executar operações sobre eles, como Hennessy e Patterson descrevem

"Registradores são a unidade de armazenamento primária de um computador. Eles são usados para armazenar dados e instruções durante a execução de um programa. Registradores são tipicamente pequenos, de alta velocidade e de acesso direto. Eles são usados para armazenar dados que estão sendo usados atualmente pelo processador, como variáveis locais, endereços de memória, resultados intermediários de cálculos e instruções." (Hennessy & Patterson, 2019, p. 33-34, Computer Architecture: A Quantitative Approach, tradução nossa)<sup>16</sup>

Constata-se que os registradores trabalham operando com dados binários, que por sua vez é a linguagem de máquina pura, não entendível para os humanos em sua forma total, mas de forma importante e concisa é fundamental o entendimento desse tipo de tecnologia para o profissional da tecnologia, entender como a arquitetura e funcionamento da máquina funciona é ideal para o programador execute seu trabalho com a maior precisão e otimização possível e realizar a construção de um sistema operacional eficiente.

---

<sup>15</sup>The CPU, or rather microprocessor, is the heart of any computer system. The microprocessor calculates, performs logical operations and manages data flows by reading instructions from memory and then executing them.

<sup>16</sup>Registers are the primary storage unit of a computer. They are used to store data and instructions during program execution. Registers are typically small, high-speed, and directly accessible. They are used to store data that is currently being used by the processor, such as local variables, memory addresses, intermediate calculation results, and instructions.

O sistema binário faz a utilização de apenas dois dígitos em sua totalidade (0, 1) para representar seus valores, como definidos por Petterson e Henssey “ O sistema binário é a linguagem dos computadores. É um sistema de dois dígitos, 0 e 1, que representam os dois estados de um interruptor: ligado ou desligado.” (Patterson & Hennessy, 2019, p. 24, Computer Organization and Design: The Hardware/Software Interface, tradução nossa)<sup>17</sup>. O processo básico de tradução de números decimais para binários que usaremos nesse exemplo, para fundamentar o conhecimento sobre a linguagem de máquina, se baseia em dividir repetidamente o número decimal por 2 e coletar os restos das divisões para que assim de forma lógica cheguemos a obter a sua representação binária.

Neste exemplo, vamos converter o número 25 para binário. Primeiramente, dividimos 25 por 2, obtendo um resto de 1. Em seguida, dividimos o quociente, que é 12, por 2 novamente, resultando em um resto de 0. Continuamos esse processo, realizando divisões consecutivas por 2, até que o quociente não seja mais divisível. Assim, dividimos o resultado anterior, que é 6, por 2, e obtemos um resto de 0. Depois, dividimos o quociente da operação anterior, que é 3, por 2, resultando em um resto de 1. Por fim, dividimos o quociente, que é 1, da operação anterior, e recebemos um resto de 0.

Pela realização deste cálculo conseguimos obter o resto das divisões, e lendo de maneira invertida sobre o valor da última divisão até a primeira, agrupando o resto de todas as divisões conseguimos obter o binário 11001, que em decimal significa 25, a divisão sucessiva por 2 é a base do processo de conversão decimal para binário, uma forma eficiente e simples de encontrar o binário determinado número, mas que pode se tornar complexo à medida que os números crescem em ordem de grandeza, sendo definido por Rusling como.

“Esses uns e zeros correspondem a interruptores elétricos que estão ligados ou desligados. Assim como 42 é um número decimal, o que significa '4 dezenas e 2 unidades', um número binário é uma série de dígitos binários, onde cada um representa uma potência de 2. Nesse contexto, uma potência significa o número de vezes que um número é multiplicado por ele mesmo.

---

<sup>17</sup>the binary system is the language of computers. It is a system of two digits, 0 and 1, that represent the two states of a switch: on or off.

10 elevado à potência 1 (10<sup>1</sup>) é 10, 10 elevado à potência 2 (10<sup>2</sup>) é 10x10, 10<sup>3</sup> é 10x10x10, e assim por diante. O binário 0001 é o decimal 1, o binário 0010 é o decimal 2, o binário 0011 é o 3, o binário 0100 é o 4 e assim por diante. Portanto, o 42 em decimal é 101010 em binário (ou seja, 2 + 8 + 32 ou 2<sup>1</sup> + 2<sup>3</sup> + 2<sup>5</sup>). Em vez de usar binário para representar números em programas de computador, é comum utilizar outra base, chamada hexadecimal. “ (RUSLING, David A. The Linux Kernel. 1996. p. 8, tradução nossa) <sup>18</sup>

### 2.2.6 GNU/LINUX, SUAS DISTRIBUIÇÕES, ARQUITETURA E VARIEDADE TÉCNICA.

As distribuições, normalmente conhecidas pelos usuários como distros, representam um aspecto fundamental que caracteriza o GNU/Linux. Onde fundamentalmente, são conjuntos de pacotes de software previamente selecionados que incluem o Kernel Linux. Em tese, a única semelhança entre todas as distribuições é o compartilhamento do mesmo kernel em diferentes ou semelhantes versões, sendo diferentes em todos os outros aspectos. Mesmo as distribuições que derivam de outras distribuições, como o Ubuntu, que é derivado do Debian, acabam atendendo a uma ampla gama de necessidades do usuário e suas preferências, como definido por Love.

Uma distribuição Linux é um conjunto de software que inclui o kernel Linux, um ambiente de desktop, aplicativos e ferramentas. As distribuições Linux são criadas por uma variedade de organizações, incluindo empresas, organizações sem fins lucrativos e indivíduos (LOVE, 2022, p. 2)

Verifica-se que podemos definir cada distribuição como uma concretização viva da filosofia e dos princípios do Software Livre. A arquitetura que compõe uma distribuição pode ser dividida em três principais componentes que, em conjunto, formam todo o sistema operacional. Segundo Torvalds, “A arquitetura do Linux é modular, o que significa que o sistema é dividido em componentes que podem ser facilmente modificados ou substituídos” (Torvalds, 2005, p. 2). Esses

<sup>18</sup> These ones and zeros correspond to electrical switches being either on or off. Just as 42 is a decimal number meaning “4 10s and 2 units”, a binary number is a series of binary digits each one representing a power of 2. In this context, a power means the number of times that a number is multiplied by itself. 10 to the power 1 ( 10<sup>1</sup> ) is 10, 10 to the power 2 ( 10<sup>2</sup> ) is 10x10, 10<sup>3</sup> is 10x10x10 and so on. Binary 0001 is decimal 1, binary 0010 is decimal 2, binary 0011 is 3, binary 0100 is 4 and so on. So, 42 decimal is 101010 binary or (2 + 8 + 32 or 2<sup>1</sup> + 2<sup>3</sup> + 2<sup>5</sup> ). Rather than using binary to represent numbers in computer programs, another base, hexadecimal is usually used.

componentes são o seu kernel, os utilitários e ferramentas do sistema, e o gerenciador de pacotes. Já apresentamos o kernel e demonstramos sua importância neste trabalho; agora, trataremos dos outros dois componentes.

Observa-se que os utilitários e ferramentas são um conjunto de partes do sistema ou aplicativos. Esses programas podem incluir utilitários de linha de comando, listas de pacotes, bibliotecas e sistemas de inicialização, onde cada um desempenha um papel específico no funcionamento do sistema como um todo.

Entende-se que os utilitários de linha de comando fornecem funcionalidades que podem ser executadas através do terminal de comando, onde definido foram definidos pela Fundação Linux como, “Os utilitários e ferramentas do sistema Linux fornecem funções básicas para o gerenciamento do sistema, como gerenciamento de arquivos, gerenciamento de usuários e gerenciamento de serviços” (The Linux Foundation, 2023, p. 2). Eles abrangem um amplo leque de tarefas, como o gerenciamento de arquivos com comandos como “cd”, “cp”, “mv”, “ls”, a administração do sistema com “sudo”, “top”, “systemctl” e a navegação na web com comandos como “curl” e “wget”.

Definem-se bibliotecas compartilhadas como um conjunto de códigos que podem ser compartilhados por vários programas, visando economizar espaço em disco e melhorar a eficiência do sistema. Isso é um ponto fundamental do GNU/Linux, que se destaca por sua eficiência e agilidade na execução de tarefas, mesmo em máquinas mais antigas e com pouco poder de processamento.

Sabe-se que os sistemas de inicialização desempenham um papel crítico na inicialização do sistema, gerenciando a execução de serviços e aplicativos. Eles são o primeiro processo a ser executado quando ligamos nossa máquina. Atualmente, o sistema mais utilizado é o Systemd, projetado para permitir a inicialização paralela de serviços, o controle de recursos e registros centralizados. Com isso, ele oferece diferentes tipos de ferramentas, como inicialização rápida, gerenciamento centralizado e recursos avançados. Através de cada uma destas, a inicialização se comporta de maneiras diferentes, executando funções específicas, onde “ O sistema de inicialização é responsável por carregar o kernel do Linux na memória e inicializar o sistema operacional.” (The Linux Foundation, 2023, p. 3).

Entende-se que os gerenciadores de pacotes são componentes que simplificam a instalação, atualização e remoção de softwares do sistema de forma mais eficiente, garantindo que todas as dependências do programa sejam tratadas de forma automatizada, tanto na sua instalação quanto na remoção. Com isso, acabamos tendo diversas opções de gerenciadores que atendem aos usuários de formas específicas, mas cumprem a mesma função. Isso é definido pelo fato de que “Os gerenciadores de pacotes são ferramentas que automatizam o processo de instalação, atualização e remoção de software em sistemas Linux.” (The Linux Foundation, 2023, p. 4).

Existem diversos meios e soluções para a utilização deste recurso, que podem ser simples aos olhos dos leigos, como o Advanced Package Tool, ou APT, usado principalmente no Debian e em distribuições derivadas. Ele faz a utilização de arquivos de pacote que possuem a extensão .deb, onde este é conhecido pelos usuários pela sua facilidade de uso. Ou até mesmo soluções mais complexas aos mesmos olhos, como o sistema de gerenciamento da distribuição Gentoo, conhecida por seu sistema de gerenciamento de pacotes único.

Observa-se que na distribuição Gentoo, não são fornecidos pacotes binários pré-compilados. Em vez disso, ela utiliza um sistema de compilação chamado Portage, definido como 'O Gentoo é uma distribuição Linux baseada no Portage, um sistema de gerenciamento de pacotes que permite aos usuários compilar o software de acordo com suas necessidades.' (The Linux Foundation, 2023, p. 5).

Nesse sistema de gerenciamento de pacotes, o Portage é responsável por compilar e instalar os programas. O Portage mantém um banco de dados de metadados que contém informações sobre os pacotes disponíveis e suas dependências, juntamente com instruções sobre como compilá-los. Além disso, faz uso das Use Flags, que são variáveis que permitem ao usuário definir configurações de compilação específicas para determinado pacote, permitindo a personalização conforme sua necessidade. Dessa forma, “O Gentoo é otimizado para o hardware específico do usuário” (The Linux Foundation, 2023, p. 5).

Ponderamos que os utilitários e ferramentas do sistema, bem como os gerenciadores de pacotes, são peças fundamentais nas distribuições GNU/Linux. Eles garantem que o sistema seja eficiente, seguro e de fácil manutenção, ao mesmo tempo em que fornecem uma base sólida para instalar e manter software, com uma ampla variedade de ferramentas. Isso traz liberdade e acessibilidade, desde o desenvolvimento do software até a instalação.

Observa-se que uma das diferentes razões pelas quais as distribuições são tão diversas é devido à sua variedade técnica, em relação aos componentes e configurações existentes. Por meio delas, é possível personalizar e aprimorar seu sistema, adaptando-o de acordo com suas preferências. Você pode torná-lo mais leve para dispositivos com menos poder computacional ou mais robusto para máquinas avançadas. Afinal, “A arquitetura modular do Linux permite que os componentes do sistema sejam facilmente modificados ou substituídos.” (The Linux Foundation, 2023).

O ambiente de trabalho presente na distribuição é uma das peças fundamentais para alguns usuários realizarem a sua escolha de utilização, além de outras ferramentas previamente instaladas. Existe a necessidade de se sentir confortável com o sistema utilizado, se sentir aceito e instigado a usar o sistema, com isso várias distribuições variam amplamente em relação ao seu ambiente de trabalho.

“Um ambiente de desktop Linux é um conjunto de software que fornece uma interface gráfica de usuário para o sistema operacional Linux. Os ambientes de desktop Linux incluem janelas, menus, barras de ferramentas e outros elementos visuais que permitem aos usuários interagir com o sistema operacional” (HELMKE, 2022, p. 1).

Fundamentalmente, podemos caracterizar os quatro principais ambientes como GNOME, KDE, XFCE e LXQT, sendo estes os mais importantes e populares. Cada um deles possui características próprias, podendo ser mais leves e robustos ou ter uma interface mais chamativa para novos usuários. “O Linux é um sistema operacional flexível, o que significa que pode ser personalizado para atender às necessidades específicas de cada usuário.” (Ubuntu Documentation, 2023).



Algumas distribuições, como o Debian, permitem que o usuário escolha a interface durante a instalação do sistema. Durante o processo de instalação, é apresentada uma lista de interfaces, e o usuário pode escolher aquela que mais lhe agrada. Além disso, ele tem a flexibilidade de modificar essa escolha no futuro, uma vez que o GNU/Linux é um sistema totalmente maleável, não sendo algo sólido e imutável.

Com essas variantes o modo de liberação de versões também varia de acordo com a distribuição, sendo que cada uma adota um modelo diferente de lançamento. Algumas optam pelo lançamento apenas de versões estáveis para todo o seu público, o que se torna convidativo quando se fala sobre a utilização do Linux em servidores, onde a margem de risco deve ser mínima, e uma distribuição com versões estáveis, sem bugs desconhecidos, acaba sendo a melhor opção. Por outro lado, outras distribuições optam por atualizar o sistema de forma constante, o que pode levar a problemas ocasionais, e o usuário, com seu conhecimento, acaba tendo que realizar os ajustes necessários.

O esclarecimento do motivo pelo qual o GNU/Linux é amplamente adotado em um leque de cenários diferentes, onde este segundo a Red Hat Enterprise é o sistema operacional dominante em servidores, devido a sua estabilidade, segurança e baixo custo, onde grandes empresas como Google e Amazon dependem do sistema para alimentar seus servidores e serviços na nuvem, onde o mesmo acaba sendo fortemente usado em dispositivos embarcados, como roteadores, dispositivos médicos e eletrônicos de consumo como celulares e televisões.

### **2.2.7 PRIVACIDADE E SEGURANÇA COM GNU/LINUX**

Em termos gerais, para compreender a importância da privacidade no contexto em questão, é necessário primeiro entender como ela se caracteriza e atua. Ela não é apenas um conceito previamente definido como a não violação do íntimo, mas sim um conceito altamente sugestivo e enraizado em questões culturais e filosóficas de uma sociedade. Além disso, a privacidade está diretamente ligada à economia e à política nas quais o indivíduo está inserido.

Alguns dos principais pioneiros no campo da legislação sobre privacidade da informação foram Warren e Brandeis. Em 1890, eles publicaram um ensaio intitulado 'O Direito à Privacidade', no qual apresentaram pontos de vista sobre o avanço da fotografia e a ampla circulação de jornais. Na perspectiva deles, esses avanços permitiam aos jornalistas cada vez mais intrometer-se em assuntos privados, “O direito à privacidade é um direito humano fundamental, protegido por lei em muitas partes do mundo. É o direito de controlar as informações sobre si mesmo e de proteger sua vida privada de intromissões indevidas” (WARREN; BRANDEIS, 1890, p. 193)

A CMP (Plataforma de Gestão de Conhecimento) argumenta e assume a responsabilidade de garantir que todas as decisões do usuário sejam respeitadas, e que a privacidade não deve ser considerada como um limite máximo para manter as empresas e organizações afastadas

“A CMP é uma plataforma de gestão de conhecimento que ajuda as organizações a organizar, armazenar e compartilhar informações de forma eficaz. Ela fornece uma variedade de recursos e ferramentas que podem ser usados para melhorar a colaboração, a produtividade e a tomada de decisão.” (Documentação CMP, 2023)

Verifica-se que, ao contrário das grandes empresas que primariamente visam o lucro e, com isso, a mineração e coleta de dados sobre o cliente ocorrem de forma constante, organizações e empresas trabalham de maneira ativa e direta, rastreando todos os movimentos possíveis do usuário. Isso ocorre como uma solução para aprimorar os serviços de recomendações de propagandas, criando um perfil que categoriza seus gostos, rotina, pesquisas e preferências. Isso chega ao ponto de observar padrões para determinar riscos, em uma grande tentativa de entender o comportamento humano de maneira abrangente.

Diante desse cenário, torna-se fundamental que o usuário tenha conhecimento para fazer sua escolha em relação à ferramenta ou software que utilizará, pois essa escolha vai além da usabilidade e comodidade; ela impacta diretamente em sua privacidade. Na maioria das vezes, nos deparamos com a realidade em que o usuário carece de discernimento técnico para avaliar e

selecionar a solução mais apropriada e acaba erroneamente acreditando que detém o controle sobre seus dados pessoais e sua privacidade.

Entende-se que possuímos a ilusão de autonomia digital quando se trata de privacidade, uma falsa sensação de controle que, essencialmente, é ignorância e uma falácia da privacidade. Isso acontece quando há a ilusão da autonomia digital, o que impede que os indivíduos reconheçam a real extensão da exposição de seus dados pessoais. O paradigma da consciência informada emerge como um farol para orientar o usuário no meio das complexidades que enfrentamos no âmbito digital.

Em sua essência, a ideia é que, à medida que os usuários naveguem, eles devem possuir um entendimento abrangente sobre as tecnologias utilizadas e os dados que compartilham, juntamente com seus potenciais riscos, rejeitando a noção de que a ignorância é uma virtude. A conscientização e o conhecimento dos fatos envolvidos são necessários e fundamentais.

Por essa razão, em soluções de Software Livre, onde temos acesso ao código-fonte e podemos observar de maneira detalhada as verdadeiras intenções do software em relação ao cliente, podemos entender como ocorre o tráfego de dados, os protocolos de segurança programados e os riscos de espionagem do usuário, mas como descrito por Howard e LeBlanc,

“O código aberto pode ser tão seguro quanto o código proprietário, mas apenas se for desenvolvido e mantido de forma segura. Os desenvolvedores de código aberto devem seguir as melhores práticas de segurança e os usuários devem tomar medidas para proteger seus sistemas” (HOWARD; LEBLANC, 2019, p. 2)

Muitas tecnologias que seguem essa filosofia são amplamente projetadas com o objetivo de desidentificar informações sobre o cliente, tornando as informações pessoais anônimas. Isso torna praticamente impossível relacionar históricos de pesquisa ou compras a indivíduos e, assim, violar sua privacidade de alguma maneira. Esses processos, em sua maioria, utilizam técnicas sofisticadas de ocultação e Coasergin.

A ocultação é uma técnica de segurança de dados que visa tornar informações confidenciais inacessíveis a terceiros não autorizados. Ela

pode ser implementada de várias maneiras, incluindo criptografia, controle de acesso e mascaramento de dados. (Tanenbaum e Steen, 2017, p. 345)

Desde o início do seu desenvolvimento, a segurança sempre foi um dos principais pilares do GNU/Linux, devido à sua arquitetura e à sua adoção como um Software Livre, com auditoria constante por parte de profissionais e especialistas na área

A mineração de dados e a coleta de informações também ocorrem em softwares proprietários, nos quais a auditoria por terceiros não pode ser realizada devido à propriedade intelectual privada e exclusiva. Nesse cenário, os princípios de privacidade atingiram um novo patamar com a disseminação da tecnologia e a coleta massiva de dados por governos, empresas e organizações. Essas entidades utilizam as informações coletadas para fins estabelecidos por elas mesmas, seja por questões de segurança ou aprimoramento das tecnologias, através da análise dos padrões de cada usuário, criando perfis de comportamento para direcionar propagandas

Assim, ao adotarmos uma perspectiva mais aprofundada sobre o tema, o software proprietário essencialmente limita o controle do usuário sobre a ferramenta. Tal restrição impede a possibilidade de estudar ou compreender a natureza do software, levando potencialmente à presença de ferramentas que comprometem diretamente os direitos à privacidade e segurança do usuário. Em certos casos, essa limitação se estende ao ponto de envolver o aluguel do dispositivo para executar tarefas de mineração de dados.

Com isso, software acaba tendo controle sobre vários aspectos da sua máquina, como a execução de programas e o acesso a dados sem o conhecimento prévio do usuário. Um exemplo disso é a Microsoft e seu controle sobre a execução de programas e o acesso a dados, impondo restrições aos usuários e a determinados programas que poderiam ser usados para proteger contra tais ações.

Parte principal do texto, que contém a exposição ordenada e pormenorizada do assunto. Divide-se em seções e subseções, que variam em função da abordagem do tema e do método, como dito por Santos, “Dever ético dos hackers compartilhar sua perícia escrevendo softwares de código-fonte aberto

facilitando acesso a informações e a recursos de computação sempre que possível” (Santos, 2002, p. 3).

O sistema operacional se destaca nessa área, adotando uma abordagem sólida e concisa em relação às permissões, as quais são extremamente restritas. Ele separa o sistema em diferentes níveis de privilégios de uso e acesso, determinando quem pode escrever, ler e executar arquivos. Essa separação ocorre em três níveis: proprietário, grupos e outros. Para executar comandos com privilégios de administrador, o usuário comum precisa conhecer os comandos e a senha associada ao dispositivo, como no caso do uso do comando “sudo”.

“O Linux é um sistema operacional seguro por design. O kernel Linux é projetado para ser robusto e resistente a ataques. O código-fonte do kernel Linux é amplamente revisado por uma grande comunidade de desenvolvedores, o que ajuda a identificar e corrigir vulnerabilidades de segurança.” (BINNIE, 2022, p. 1)

Com isso, o sistema acaba se tornando menos atrativo para hackers, que têm como principal objetivo infectar o maior número de usuários possível para que seus ataques sejam eficazes. O Linux, com todas as suas configurações padrão de controle de acesso, separação de privilégios, atualizações regulares, firewall, criptografia e autenticação forte, acaba impondo vários obstáculos a tais práticas maliciosas.

Por ser um sistema projetado e pensado na segurança do cliente, o GNU/Linux conta com protocolos de defesa e segurança integrados no seu Kernel de forma ativa, sendo notáveis o Linux Kernel Lockdown e o SELinux.

Observa-se que para iniciarmos ainda mais no nosso debate sobre segurança é necessário entender o que é secure boot (SB), onde este se caracteriza como um recurso na UEFI que fornece uma verificação concisa sobre o estado na cadeia de inicialização do sistema seja um código confiável, sendo projetado para proteger todo o sistema contra códigos maliciosos, onde cada programa inicializado possui sua própria assinatura, e o sistema trabalha verificando essas assinaturas criptografadas.

“Secure Boot é uma tecnologia que ajuda a proteger os sistemas contra malware e ataques de inicialização. Ela funciona verificando a integridade do firmware de inicialização e do kernel do sistema operacional antes de permitir que o sistema seja inicializado” (The Linux Foundation, 2022)

Proposto por Matthew Garrett, um grande defensor do Secure Boot (SB), o Linux Kernel Lockdown foi integrado na versão do Kernel 5.4 após anos de debate na comunidade sobre sua inclusão no Kernel. O próprio Linus Torvalds não concordava com a relação da ferramenta com o Secure Boot, já que nem todos os usuários têm máquinas com o recurso UEFI ativado. Ativamente, a ferramenta trabalha tentando proteger o sistema contra modificações não autorizadas, bloqueando o acesso a determinados componentes do Kernel, incluindo o root.

Quando o bloqueio está em vigor, vários recursos são desativados ou têm seu acesso restrito, o que inclui o bloqueio do acesso a dados de segurança. Somente os módulos validamente assinados podem ser carregados, juntamente com binários assinados, proporcionando uma camada significativa de segurança ao sistema. Isso impede que terceiros, por meio de arquivos ou programas maliciosos, infectem silenciosamente o sistema do usuário.

Desenvolvido inicialmente pela NSA como uma série de patches para complementar o Kernel Linux, o SELinux utiliza modos de segurança do tipo LSM e funciona como um controlador de acesso a aplicações, arquivos e processos no sistema. Ele emprega um conjunto de regras e políticas de segurança para determinar o que é permitido e o que não é quando uma entidade específica, como uma aplicação ou processo, solicita acesso a um determinado recurso ou objeto, onde, “SELINUX é um sistema de segurança de domínios de segurança para sistemas operacionais baseados em Linux. Ele fornece uma maneira de controlar o acesso a recursos do sistema e aplicativos por processos e usuários” (The Linux Foundation, 2022).

A partir dessa perspectiva, o SELinux entra em ação verificando o cache de vetores de acesso, onde as devidas permissões para as entidades presentes no sistema são armazenadas. Com isso, o SELinux decide se permite ou nega o acesso a essa entidade. Como um protocolo adicional de segurança, se o SELinux não conseguir tomar uma decisão com base nas permissões armazenadas em cache, ele envia uma solicitação para um servidor de segurança. Esse servidor realiza uma verificação completa do contexto de segurança da aplicação e aplica um

novo contexto ao banco de dados de políticas do SELinux, revelando o nível de permissão.

Portanto, temos duas ferramentas de extrema importância e eficiência no combate a ataques externos, que protegem o usuário contra softwares e processos maliciosos, sem que o usuário precise ter conhecimento sobre toda a arquitetura e segurança do sistema. Isso ocorre de maneira automática, sem o conhecimento do cliente, proporcionando comodidade e segurança, que são os principais benefícios dessas ferramentas.

Ao observar esses pontos, podemos notar o impacto direto que uma aplicação, neste caso, um sistema operacional que é um software livre, tem nos usuários, proporcionando um senso de comunidade e pertencimento. As decisões são orientadas para o benefício da maioria, e, assim, a assistência de diversas partes é fornecida de maneira constante. Isso resulta em evolução e segurança a uma velocidade exponencial, agregando valor para aqueles que usam ou desejam usar uma determinada ferramenta. Os benefícios para a segurança, privacidade e objetivos são moldados de maneira democrática e concisa.

### **3 CONCLUSÃO**

De acordo com os princípios teóricos apresentados anteriormente, tende-se a verificar a relevância da utilização do software livre em diversos aspectos da tecnologia contemporânea. No entanto, isso não diminui a importância que o software proprietário exerce, uma vez que também desempenha um papel significativo no cenário tecnológico. Este estudo buscou esclarecer os conceitos fundamentais, a filosofia subjacente e as práticas relacionadas ao software livre. Além disso, ao explorar o sistema de software livre mais proeminente, o GNU/Linux, conseguimos exemplificar questões críticas relacionadas à segurança e privacidade no mundo digital.

Observando a história do software livre, percebemos a ênfase na liberdade, que concede aos usuários o direito de usar, modificar e compartilhar o software. Esse compromisso com a liberdade é comparável ao icônico monólito na obra 2001: Uma Odisseia no Espaço, que desempenha um papel crucial na evolução humana.

Da mesma forma, o software livre atua como um elemento fundamental na evolução tecnológica, orientando uma comunidade diversificada com objetivos comuns.

E através do GNU/Linux, que por si só é um exemplo proeminente e uma referência quando se trata de Software Livre, observamos o seu impacto significativo e abrangente na indústria tecnológica. Seu crescimento constante e a adoção generalizada em vários setores tecnológicos demonstram o quão poderosa pode ser a evolução e manutenção de um sistema que recebe contribuições ativas de uma gigantesca comunidade.

Nesse contexto, pode vir a evidenciar que a disseminação do software livre não é apenas uma questão ideológica que afeta um grupo limitado de pessoas, mas sim uma influência global que muitas vezes passa despercebida por trás das cifras de vendas. Como indivíduos e como parte de uma comunidade, contribuimos para algo maior e deixamos nossa marca em um legado duradouro.

Em um mundo cada vez mais interconectado, a segurança e a privacidade tornaram-se preocupações centrais. Nesse sentido, o software livre desempenha um papel fundamental, ajudando a dissipar as incertezas que envolvem nossa presença no mundo digital. A proteção da privacidade online é uma questão de compreensão e conhecimento. O software livre oferece uma visão transparente desse ecossistema complexo..

O software livre oferece uma janela de transparência, onde, no entanto, essa jornada em direção à conscientização informada não é unilateral; requer a prontidão dos usuários para se envolverem ativamente na defesa de sua própria privacidade e dados pessoais. A escolha da ferramenta de software, embora importante, é apenas o primeiro passo na busca por uma compreensão mais profunda e uma defesa eficaz da privacidade em nosso mundo digital atual

Nós podemos concluir que os benefícios provenientes da utilização de aplicações de software livre podem se manifestar de diversas formas, desde o incentivo à educação e compartilhamento de informações, por meio dos quais podemos construir um alicerce de conhecimento, até a proteção da privacidade do indivíduo e a salvaguarda de sua vida digital, envolvendo seus dados e segurança. Além disso, esses benefícios vão além das questões técnicas, abrangendo também



o âmbito sociológico e cultural, promovendo debates e acesso democrático à tecnologia.

## REFERÊNCIAS

BARUH, Lemi; SECINTI, Ekin; CEMALCILAR, Zeynep. Preocupações de Privacidade Online e Gerenciamento de Privacidade: Uma Revisão Meta-Analítica. Em: \*Revista de Comunicação\*, 2017.

BESSEN, James. Política Governamental em Relação ao Software Livre e de Código Aberto. Brookings Institution Press, Instituto Empresarial Americano, 2002. p. 12-33.

Desenvolvimento do Kernel Linux em Números. Ano 2012. Disponível em: <<https://www.pingdom.com/blog/desenvolvimento-do-kernel-linux-em-numeros/>>. Acesso em: 10 set. 2023.

FORTES, Débora. Linux: o que você pode ganhar ou perder com a revolução do pingim. Revista Info-Exame, São Paulo, ano 17, n. 200, p. 48-74, nov. 2002.

KERNIGHAN, Brian; RITCHIE, Dennis. A Linguagem de Programação C. Pearson, 2ª edição, 22 de fevereiro de 1978. Publicado em: 22 de março de 1988.

Neves Garcia, Mauro; Braga dos Santos, Silvana Mara; Silva Pereira, Raquel da; Bedineli Rossi George. SOFTWARE LIVRE EM RELAÇÃO AO SOFTWARE PROPRIETÁRIO: ASPECTOS FAVORÁVEIS E DESFAVORÁVEIS PERCEBIDOS POR ESPECIALISTAS. Gestão & Regionalidade, 2010, 26(78), 106-120. Disponível em: <<https://www.redalyc.org/articulo.oa?id=133417428009>>. Acesso em: 29 de set. 2023.

O Ambiente de Desktop do Linux, Matthew Helmke, 2ª edição, 2022, p. 1.

O Kernel Linux: Uma Referência Completa, Robert Love, 4ª edição, 2022, p. 2.

O SISTEMA OPERACIONAL GNU. O que é copyleft. Disponível em: <<https://www.gnu.org/licenses/copyleft.pt-br.html#:~:text=Copyleft%20%C3%A9%20um%20m%C3%A9todo%20geral,do%20programa%20tamb%C3%A9m%20sejam%20livres>>. Acesso em: 20 set. 2023.

O SISTEMA OPERACIONAL GNU. O que é Software Livre? Disponível em: <<https://www.gnu.org/philosophy/free-sw.pt-br.html>>. Acesso em: 20 set. 2023.

OVADIA, Steven. Linux para Acadêmicos, Parte II: As Vantagens do Software Livre e de Código Aberto. Publicado em: 21 fev. 2014.

Perry, G. Miller, D. Guia Absoluto do Iniciante em Programação C (3ª edição). Que Publishing, 2013.

Red Hat. O que é um servidor Linux? Recuperado em 21 de outubro de 2023, de <https://www.redhat.com/pt-br/topics/linux/linux-server>

RUSLING, David A. O Kernel Linux. New Riders Pub, 1999.

SANTOS, Francisco Coelho. Peripécias de agosto: alguns episódios da “cena hacker”. Faculdade de Filosofia e Ciências Humanas, Universidade Federal de Minas Gerais, Belo Horizonte, 2002. Mimeografado.

Segurança no Linux: Um Guia Prático, Chris Binnie, 2ª edição, 2022, p. 1.

Silveira, Sérgio Amadeu. Inclusão Digital, Software Livre e Globalização Contra-hegemônica. Em: Software Livre e Inclusão Digital. São Paulo: Conrad Editora do Brasil, 2003.

Silveira, Sérgio Amadeu. Software Livre: a Luta pela Liberdade do Conhecimento. São Paulo: Editora Fundação Perseu Abramo, 2004.

Singh, Amandeep, R.K Bansal, Ph.D, e Neetu Jha. "Open Source Software vs Proprietary Software." *International Journal of Computer Applications*, vol. 114, no. 18, março de 2015

STALLMAN, Richard M.; LESSING, Lawrence. Software Livre, Sociedade Livre: Ensaios Selecionados de Richard M. Stallman. Editora: Fundação Software Livre, 1 de junho de 2002.

Tanenbaum, Andrew S., e Maarten van Steen. "Ocultação de dados." *Redes de computadores: uma abordagem introdutória*. 7. ed. Pearson Education, 2017.

TORVALDS, Linus; DIAMOND, David. *Só por Prazer: Linux, os bastidores da sua criação*. Rio de Janeiro: Campus, 2001.

Tiboni, Antonio Carlos. *Software Livre como Política de Governo*. 2014. Disponível em: <<http://hdl.handle.net/10183/127438>>. Acesso em: 1 de set. 2023.

UBUNTU DOCUMENTATION. Site de Documentação do Ubuntu. 2023. Disponível em: <https://help.ubuntu.com/>. Acesso em: 30 ago. 2023.

Vasilomanolakis, E., Daubert, J., Luthra, M., Gazis, V., Wiesmaier, A., & Kikiras, P. Sobre a Segurança e Privacidade de Arquiteturas e Sistemas da Internet das Coisas. Em: 2015 Oficina Internacional sobre Internet das Coisas Segura (SIoT). DOI: 10.1109/siot.2015.9.

Warren, Samuel D., e Louis D. Brandeis. "The Right to Privacy." *Harvard Law Review* 4(5):193-220, 1890.

WHEELER, David A. Por que Software de Código Aberto / Software Livre (CA/SL)? Olhe para os Números! Publicado em: 7 nov. 2004.