



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE GRADUAÇÃO
DEPARTAMENTO DE ENGENHARIAS E TECNOLOGIAS
CURSO DE BACHARELADO INTERDISCIPLINAR EM TECNOLOGIA DA
INFORMAÇÃO

JOÃO PEDRO MEDEIROS NOGUEIRA MOURA

**IMPACTO DO USO DE SISTEMAS OPERACIONAIS NO APRENDIZADO DE
PROGRAMAÇÃO DOS DISCENTES NA UFERSA CAMPUS DE PAU DOS FERROS**

PAU DOS FERROS - RN

2026

JOÃO PEDRO MEDEIROS NOGUEIRA MOURA

**IMPACTO DO USO DE SISTEMAS OPERACIONAIS NO APRENDIZADO DE
PROGRAMAÇÃO DOS DISCENTES NA UFERSA CAMPUS DE PAU DOS FERROS**

Monografia apresentada à Universidade Federal Rural do Semi-Árido como requisito para obtenção do título de Bacharel Interdisciplinar em Tecnologia da Informação.

Orientador: Reudismam Rolim de Sousa, Prof. Dr.

PAU DOS FERROS - RN

2026

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei n° 9.279/1996 e Direitos Autorais: Lei n° 9.610/1998. O conteúdo desta obra tomar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos

M929i Moura, João Pedro Medeiros Nogueira.
Impacto do uso de sistemas operacionais no
aprendizado de programação dos discentes na UFRSA
campus de Pau dos Ferros / João Pedro Medeiros
Nogueira Moura. - 2026.
80 f. : il.

Orientador: Reudismam Rolim de Sousa.
Monografia (graduação) - Universidade Federal
Rural do Semi-Árido, Curso de Tecnologia da
Informação, 2026.

1. Sistemas operacionais. 2. Sistema
operacional windows. 3. Distribuições linux. 4.
Aprendizagem. 5. Linguagens de programação. I.
Sousa, Reudismam Rolim de, orient. II. Título.

Ficha catalográfica elaborada por sistema gerador automático em conformidade
com AACR2 e os dados fornecidos pelo autor(a).

Biblioteca Campus Pau dos Ferros
Bibliotecária: Erlanda Maria Lopes da Silva
CRB: 15/966

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Institut de Ciéncias Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Biblioteca da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

JOÃO PEDRO MEDEIROS NOGUEIRA MOURA

**IMPACTO DO USO DE SISTEMAS OPERACIONAIS NO APRENDIZADO DE
PROGRAMAÇÃO DOS DISCENTES NA UFERSA CAMPUS DE PAU DOS FERROS**

Monografia apresentada à Universidade Federal Rural do Semi-Árido como requisito para obtenção do título de Bacharel Interdisciplinar em Tecnologia da Informação.

Defendida em: 03/07/2026.

BANCA EXAMINADORA

Reudismam Rolim de Sousa, Prof. Dr. - (UFERSA)
Presidente

Hortência Pessoa Rêgo Gomes, Profa. Ma. - (UFERSA)
Membro Examinador

Dyego Magno Oliveira Souza, Bel. - (UFERSA)
Membro Examinador

DEDICATÓRIA

Dedico este trabalho ao professor Reudismam Rolim de Sousa, que muito me ajudou no desenvolvimento da pesquisa e esteve ao meu lado até o momento da conclusão (Presente).

AGRADECIMENTOS

Agradeço, imensamente, aos familiares, professores e amigos por todo o apoio e ajuda e, também, aos discentes envolvidos na pesquisa, que contribuíram com seu tempo para a realização deste trabalho.

RESUMO

Este trabalho busca compreender os efeitos da utilização de diferentes sistemas operacionais (SOs) no ambiente acadêmico, tendo como principal foco a análise destes no aprendizado de programação dos discentes nos cursos de Tecnologia da Informação, Engenharia de *Software*, Engenharia da computação e Ciência e Tecnologia. Os SOs oferecem um leque de recursos, funcionalidades e filosofias de *design* que podem moldar substancialmente a jornada de ensino-aprendizagem dos discentes. Para Tanenbaum e Bos (2016), o Windows oferece uma GUI (*Graphical User Interface*) intuitiva e amigável, o que pode diminuir as barreiras de entrada para iniciantes em programação, fazendo com que os discentes foquem mais rapidamente na lógica de programação sem se deter em complexidades de configuração do sistema. Entretanto, o Linux (e suas diversas distribuições) apresenta uma natureza intrínseca mais modular, aberta e altamente personalizável, características derivadas de seu modelo de desenvolvimento baseado em *software* livre e código aberto. Para o desenvolvimento desta pesquisa, utilizou-se um questionário, no qual foram avaliadas características como: idade, curso, tempo de estudo de programação, linguagens de programação utilizadas, diversas avaliações a respeito do SO, como incompatibilidade, facilidade de uso, além de aspectos retirados e adaptados da ISO/IEC 25000. Dentre os resultados obtidos na pesquisa, tem-se que os discentes que utilizavam sistemas baseados em Linux obtiveram uma maior porcentagem de notas acima da média em relação aos grupos de discentes que utilizavam Windows, *Dual Boot* e *Trial Boot*. Além disso, os discentes consideraram os SOs baseados em Linux como melhores para o aprendizado de programação. Conclui-se que a utilização de um determinado sistema para programação vem da ponderação entre as mais diversas características, como: a facilidade de uso, a flexibilidade, a personalização, entre outros aspectos pessoais ou acadêmicos. Nesse contexto, a escolha do SO se torna, portanto, uma decisão estratégica, em que o discente deve ponderar-se a respeito das ferramentas, da compatibilidade e da filosofia de trabalho que melhor se adequa às suas necessidades e objetivos.

Palavras-chave: sistemas operacionais; sistema operacional windows; distribuições linux; aprendizagem; linguagens de programação.

ABSTRACT

This work seeks to understand the effects of using different operating systems (OSs) in the academic environment, focusing primarily on their impact on students' programming learning in Information Technology, Software Engineering, Computer Engineering, and Science and Technology courses. Operating systems offer a range of resources, functionalities, and design philosophies that can substantially shape students' learning journey. According to Tanenbaum and Bos (2016), Windows offers an intuitive and user-friendly GUI (Graphical User Interface), which can lower entry barriers for programming beginners, allowing students to focus more quickly on programming logic without getting bogged down in system configuration complexities. However, Linux (and their various distributions) have a more modular, open, and highly customizable intrinsic nature, characteristics derived from their development model based on free and open-source software. For the development of this research, a questionnaire was used, in which characteristics such as age, course of study, programming study time, programming languages used, various evaluations regarding the operating system, such as incompatibility and ease of use, as well as aspects taken and adapted from ISO/IEC 25000 were evaluated. Among the research findings, students using Linux-based systems achieved a higher percentage of above-average grades compared to groups using Windows, dual-boot, and trial-boot configurations. Furthermore, the students considered Linux-based operating systems to be superior for learning programming. In conclusion, it is clear that the use of a particular system comes from the consideration of various characteristics, such as ease of use, flexibility, customization, among other personal or academic aspects. In this context, the choice of operating system therefore becomes a strategic decision, in which the student must consider the tools, compatibility, and work philosophy that best suits their needs and objectives.

Keywords: operating systems; windows operating system; linux distributions; learning; programming languages.

LISTA DE FIGURAS

Figura 1 - Faixa etária dos discentes.....	30
Figura 2 - Cursos dos discentes.....	31
Figura 3 - Período dos discentes.....	32
Figura 4 - Tempo de estudo de programação.....	33
Figura 5 - Média final dos discentes nas componentes curriculares analisadas.....	34
Figura 6 - SOs utilizados em cada componente curricular analisado.....	36
Figura 7 - SOs utilizados nos componentes curriculares analisados.....	37
Figura 8 - Distribuições Linux e a média final dos discentes.....	39
Figura 9 - Windows e a média final dos discentes.....	41
Figura 10 - Dual Boot (Windows, Distribuições Linux ou macOS) e a média final dos discentes.....	43
Figura 11 - Trial Boot (Windows e distribuições Linux) e média final dos discentes.....	45
Figura 12 - SOs e a média final dos discentes.....	46
Figura 13 - Avaliação do SO com base nas métricas ISO/IEC.....	48
Figura 14 - Avaliação do SO com base nas métricas ISO/IEC (total).....	49
Figura 15 - Importância do SO no desempenho nos componentes curriculares de programação.....	50
Figura 16 - Satisfação para programar com o SO utilizado.....	52
Figura 17 - Facilidade para programar com o SO utilizado.....	53
Figura 18 - Atendimento do SO utilizado à evolução das necessidades dos discentes.....	54
Figura 19 - Confiança no SO para explorar novos conceitos de programação.....	55
Figura 20 - Os materiais didáticos utilizados são compatíveis com o SO.....	56
Figura 21 - Frequência de problemas inesperados do SO.....	57
Figura 22 - Confiança no SO em relação à proteção de trabalho contra perda ou erro.....	58
Figura 23 - Resultado das perguntas objetivas.....	59
Figura 24 - Principal SO utilizado.....	64
Figura 26 - Tempo de uso do principal ambiente de desenvolvimento para os componentes curriculares analisados em relação ao SO marcado como principal.....	65
Figura 27 - SO mais adequado para o aprendizado de programação.....	66
Figura 28 - Frequência de alternância entre SOs para programar.....	67
Figura 29 - Linguagens de programação utilizadas.....	68
Figura 30 - Relação para menos de um ano de estudo de programação.....	69
Figura 31 - Relação entre um a dois anos de estudo de programação.....	70
Figura 32 - Relação entre três a quatro anos de estudo de programação.....	70
Figura 33 - Relação entre cinco a seis anos de estudo de programação.....	71
Figura 34 - Relação entre sete a oito anos de estudo de programação.....	71
Figura 35 - Relação para mais de oito anos de estudo de programação.....	72
Figura 36 - Relação entre todos os intervalos de estudo de programação (Compilado das Figuras 30 a 35).....	72

Figura 37 - Sentimento causados pelo SO.....	73
--	----

LISTA DE QUADROS

Quadro 1 - Estrutura do questionário aplicado aos discentes.....	20
Quadro 2 - Estrutura do questionário aplicado aos discentes.....	26

LISTA DE ABREVIATURAS E SIGLAS

BICT	Bacharelado Interdisciplinar em Ciência e Tecnologia
BITI	Bacharelado Interdisciplinar em Tecnologia da Informação
CPU	<i>Central Processing Unit</i> / Unidade Central de Processamento
EC	Engenharia da Computação
ES	Engenharia de <i>Software</i>
GUI	<i>Graphical user interface</i> / Interface Gráfica de Usuário
IEC	<i>International Electrotechnical Commission</i> / <i>Comissão Eletrotécnica Internacional</i>
ISO	<i>International Organization for Standardization</i> / <i>Organização Internacional de Normalização</i>
SO	Sistema operacional
TI	Tecnologia da Informação
UNIX	<i>Uniplexed Information and Computing System</i> / Sistema de Computação e Informação Uniplexado
UX	<i>User Experience</i> / Experiência de Usuário
WSL	<i>Windows Subsystem for Linux</i> / Subsistema Windows para Linux

SUMÁRIO

1 INTRODUÇÃO.....	14
1.1 Declaração Sobre Uso de Inteligência Artificial.....	16
2 REFERENCIAL TEÓRICO.....	17
2.1 Sistemas Operacionais.....	17
2.2 Lógica de programação.....	18
2.3 ISO/IEC.....	18
3 METODOLOGIA.....	20
3.1 Avaliação do Sistema Operacional.....	27
4 RESULTADOS.....	31
4.1 Perfil dos Discentes.....	31
4.2 Média Final dos Discentes.....	35
4.3 SOs Utilizados por Componente Curricular.....	37
4.4 SOs e Média Final.....	39
4.4.1 Distribuições Linux e a Média Final dos discentes.....	40
4.4.2 Windows e a Média Final dos discentes.....	42
4.4.3 Dual Boot e a Média Final dos discentes.....	44
4.4.4 Trial Boot e a Média Final dos discentes.....	46
4.4.5 SOs e a Média Final dos discentes.....	48
4.5 Avaliação Geral dos SOs com base em ISO/IEC.....	49
4.6 Avaliação dos SOs em Relação à Programação.....	51
4.7 Perguntas Objetivas.....	60
4.7.1 Análise da Primeira Questão.....	61
4.7.2 Análise da Segunda Questão.....	62
4.7.3 Análise da Terceira Questão.....	63
4.7.4 Análise da Quarta Questão.....	64
4.7.5 Análise da Quinta Questão.....	64
4.8 Perguntas sobre Sistemas Operacionais.....	65
4.9 Linguagens de Programação.....	70
4.10 SOs e Tempo de Estudo das Linguagens de Programação.....	71
4.11 Perguntas Subjetivas.....	75
4.11.1 Sentimentos Causados ao Usar um SO.....	75
4.11.2 Dificuldades e Facilidades Encontradas no SO.....	76
4.11.3 Motivações e Desmotivações Advindas do SO.....	77
4.11.4 Adequação do SO ao Aprendizado.....	77
5 CONSIDERAÇÕES FINAIS.....	79
REFERÊNCIAS.....	80

1 INTRODUÇÃO

A habilidade de resolução de problemas por meio da programação é complexa e desafiadora (Bennedsen; Caspersen, 2007; Jenkins, 2002), exigindo do discente não apenas a compreensão de conceitos abstratos e complexos, como estruturas de dados e algoritmos (Lister *et al.*, 2006; Cañas; Bajo; Gonzalvo, 1994), mas também a prática desses conceitos (Castro, 2025) e o domínio de ferramentas de *software* (Silva *et al.*, 2024). Por sua vez, na Computação, os Sistemas Operacionais (SOs) são indispensáveis (Tanenbaum; Bos, 2016), fornecendo o alicerce e a estrutura essencial para os desenvolvedores. No contexto do ensino de programação, os SOs desempenham um ambiente ideal para programar, operar, codificar, compilar, depurar e testar suas soluções (Silberschatz; Galvin; Gagne, 2018; Maziero, 2019).

Os SOs oferecem um leque de recursos, funcionalidades e filosofias de *design* que podem moldar substancialmente a jornada de ensino-aprendizagem (Kenski, 2024). O SO Windows, por exemplo, disponibiliza uma GUI (*Graphical User Interface*) intuitiva e amigável (Tanenbaum; Bos, 2016), o que pode diminuir as barreiras de entrada para iniciantes em programação, permitindo-lhes focar mais rapidamente na lógica de programação sem se deter em complexidades de configuração do sistema (Chan; Zhang, 2007; Rauterberg, 1992). Em contrapartida, o SO Linux (e suas diversas distribuições¹) apresenta uma natureza inerentemente mais modular, aberta e altamente personalizável, características derivadas de seu modelo de desenvolvimento baseado em código livre e aberto (Silberschatz; Galvin; Gagne, 2018). Embora possa ter uma curva de aprendizado inicial mais íngreme, especialmente no que diz respeito ao uso da linha de comando, esse SO pode oferecer uma visão mais profunda do funcionamento interno dos sistemas de *software*, sendo amplamente utilizado em contextos educacionais para o ensino de conceitos sobre SOs (Tanenbaum; Bos, 2016).

Bryant e O'Hallaron (2016) discutem que a compreensão dos SOs (incluindo sua interação entre o *hardware* e o compilador) é fundamental para o desenvolvimento de *software*. Além disso, o ensino acerca do SO pode servir também como um indicador de aprendizagem (Kolgatin; Holubnychy; Kolgatina, 2021). Essa relação reforça a necessidade de integrar, desde os ciclos iniciais da formação em TI, discussões sobre como as características dos SOs influenciam no

¹ Distribuição Linux: Conjunção entre o *kernel* Linux, as ferramentas utilitárias (GNU) e opcionalmente uma GUI. (ex.: Ubuntu, Linux Mint, etc.).

aprendizado e no desenvolvimento de *software*.

Dado que o aprendizado de programação é essencial para a formação em TI (Sousa; Leite, 2020), torna-se necessário escrever, depurar, testar e manter um código eficiente, uma competência central indispensável (McConnell, 2004). No contexto de Computação, os SOs constituem uma parte essencial e intrínseca do desenvolvimento de *software*, representando a base na qual os programas são executados (Tanenbaum; Bos, 2016).

Segundo Guzdial (2015), a infraestrutura tecnológica pode atuar como facilitadora ou como fonte de frustração e atraso na aprendizagem de programação, reforçando o uso de ferramentas no SO de apoio à programação.

Diante desse contexto, o presente trabalho visa compreender os impactos dos SOs utilizados pelos discentes durante o processo de aprendizagem de programação. Para isso, foi realizado um estudo de caso dos componentes curriculares do curso de Bacharelado Interdisciplinar em Tecnologia da Informação (BITI) da UFERSA Campus Pau dos Ferros. Esses componentes também podem estar associados a outros cursos, como os bacharelados Interdisciplinar em Ciência e Tecnologia (BICT), Engenharia de *Software* (ES) e Engenharia da Computação (EC), que compartilham componentes no citado Campus.

O objetivo geral deste trabalho é analisar as implicações, desafios e aprendizados, tais como, seu desempenho acadêmico, obstáculos técnicos e práticos decorrentes do uso de diferentes tipos de SOs (como Windows, distribuições Linux, macOS, etc.) isoladamente ou concomitantemente (*Dual Boot*², *Trial Boot*³), na aprendizagem de programação dos discentes da UFERSA Campus Pau dos Ferros.

Por fim, a estrutura do projeto de pesquisa foi dividida em cinco capítulos, dos quais no capítulo presente foi realizada uma contextualização da temática, apresentando justificativas, problematização e objetivos do trabalho. O capítulo dois tem seu enfoque voltado ao referencial teórico, abrangendo o conceito de SO e de suas diferentes variações, a lógica de programação e as ISO/IECs. O capítulo três contempla a metodologia utilizada no trabalho. O capítulo quatro descreve os resultados obtidos na pesquisa. O capítulo cinco apresenta as conclusões finais sobre a pesquisa.

² *Dual Boot*: Utilizar dois SOs no mesmo ambiente computacional.

³ *Trial Boot*: Utilizar três SOs no mesmo ambiente computacional.

1.1 Declaração Sobre Uso de Inteligência Artificial

Durante a elaboração desta pesquisa, foram utilizadas ferramentas de Inteligência Artificial Generativa (IAG) como recurso de apoio. Em específico, o modelo Qwen, desenvolvido pelo Alibaba, e Gemma, desenvolvido pela Google, esse último foi utilizado localmente e sem qualquer conexão à internet.

O primeiro (Qwen) foi empregado para auxiliar na revisão textual, no esclarecimento de conceitos relacionados ao tema estudado e no refinamento das perguntas aplicadas no questionário. Já o segundo (Gemma) foi empregado em resumos sobre as seções de comentários obtidos no questionário da pesquisa. As contribuições obtidas por meio dessas ferramentas foram analisadas, revisadas e adequadas pelo autor, que assume total responsabilidade pelo conteúdo apresentado, garantindo sua originalidade, consistência e conformidade com os princípios éticos da pesquisa científica, em atendimento às diretrizes estabelecidas pela Portaria CNPq nº 2.664/2026.

2 REFERENCIAL TEÓRICO

2.1 Sistemas Operacionais

Um Sistema Operacional (SO) é o *software* crucial que atua no gerenciamento dos recursos de *hardware* e *software* de um computador, muitas vezes fornecendo uma GUI ou uma CLI (interface de linha de comando), sendo a base para que os programas de aplicação possam ser utilizados. Assim, um SO atua como uma ponte entre o *hardware* físico da máquina (CPU, memória, dispositivos de entrada e saída) e os *softwares* que o usuário deseja executar. Suas principais funções incluem gerenciamento de processos (escalonamento e execução de programas), gerenciamento de memória (alocação e desalocação), gerenciamento de arquivos (organização e acesso a dados) e gerenciamento de dispositivos de *hardware* (Silberschatz; Galvin; Gagne, 2018). Os SOs podem ser apresentados em tipos variados. Como exemplos a seguir:

O Windows (Microsoft) caracteriza-se por ser um sistema de propósito geral que prioriza o uso da GUI para facilitar a interação com o usuário. Segundo Silberschatz, Galvin e Gagne (2018), o Windows utiliza uma arquitetura baseada em um micronúcleo modificado e é amplamente adotado em ambientes corporativos e domésticos pela sua vasta compatibilidade de *software*.

O macOS (Apple) é baseado no núcleo XNU (X is Not Unix) que deriva do BSD (Berkeley Software Distribution) e Mach, sendo descrito como um sistema que combina a robustez do UNIX (*Uniplexed Information and Computing System*) com uma interface proprietária altamente intuitiva. Silberschatz, Galvin e Gagne (2018) ressaltam sua integração vertical com o *hardware* da Apple e sua arquitetura de camadas, que garante segurança e desempenho.

O Linux é um núcleo (*kernel*) de SO livre e de código aberto, compatível com os padrões UNIX. Tanenbaum e Bos (2016) destacam sua natureza modular e o fato de ser um sistema multitarefa e multiusuário, em que as ferramentas do projeto GNU (GNU's Not Unix) complementam o sistema para fornecer um ambiente operacional completo e altamente estável.

O FreeBSD é um SO focado em desempenho e estabilidade em serviços de rede e servidores. Derivado do 4.4BSD-Lite da Universidade da Califórnia, Berkeley, ele mantém a

tradição do desenvolvimento colaborativo e aberto (Ziaee, 2025). Tanenbaum e Bos (2016) o definem como um exemplo clássico de sistema derivado do UNIX que influenciou diversos outros sistemas modernos.

Outros sistemas abrangem diferentes categorias, incluindo sistemas móveis, como Android e iOS, além de sistemas baseados em UNIX, como Solaris e OpenBSD (Silberschatz; Galvin; Gagne, 2018).

2.2 Lógica de programação

A lógica de programação consiste em técnicas e conceitos que estruturam a forma como um programador organiza uma sequência de instruções para resolver um problema específico de maneira eficiente e correta (Silva, 2016), sendo uma base fundamental para a criação de programas de computador.

Esse conceito é essencial para o desenvolvimento de algoritmos (passos finitos e bem definidos para atingir um objetivo), utilizando conceitos como variáveis, estruturas de controle (condicionais como *if/else*, laços de repetição como *for* e *while*), modularização (funções e procedimentos) e operadores lógicos e aritméticos. O domínio da lógica é independente da linguagem de programação, sendo o primeiro passo para aprender a codificar (Manzano, 2019).

2.3 ISO/IEC

A International Organization for Standardization (ISO) e a *International Electrotechnical Commission* (IEC) são organizações internacionais de normalização utilizadas mundialmente. Ambas são responsáveis pelo desenvolvimento e publicação de padrões internacionais, que visam garantir a qualidade, a segurança, a eficiência e a interoperabilidade de produtos, serviços e sistemas em escala global.

No contexto específico de TI, telecomunicações e Engenharia de *Software*, as normas são elaboradas pelo ISO/IEC JTC 1 (*Joint Technical Committee 1*). O prefixo conjunto "ISO/IEC" em uma norma indica que o documento foi aprovado por ambos os órgãos, garantindo que o padrão atenda aos requisitos gerais de gestão e processos industriais (ISO) quanto às rigorosas especificidades técnicas de *hardware*, *software* e eletrônica (IEC). A partir deste comitê se deriva a família de normas ISO/IEC 25000 (SQuaRE), que estabelece os modelos de referência para a

avaliação da qualidade de sistemas e *software* utilizados neste trabalho. (International Organization For Standardization; International Electrotechnical Commission, 2014).

3 METODOLOGIA

Para a condução desta pesquisa, foi desenvolvido um questionário para coletar dados sobre os discentes, tais como: os SOs que os discentes utilizam, as suas médias nos componentes curriculares relacionados à programação, entre outras informações relevantes ao estudo.

O questionário aplicado foi enviado por *e-mail* pela chefia do departamento do campus de Pau dos Ferros para os discentes dos cursos de BITI, EC, ES e BICT. A partir disso, as respostas foram coletadas.

O citado questionário abrangeu quatro cursos da UFERSA do Campus de Pau dos Ferros. Os componentes curriculares considerados neste estudo de caso foram comuns ao Projeto Pedagógico de Curso (PPC) do curso de BITI. Uma vez que este compartilha componentes com os cursos de EC, ES e BICT, pois os impactos dessa utilização também são refletidos nos componentes curriculares desses.

Para fins de verificação, antes da disponibilização para o público-alvo dessa pesquisa, houve teste antecipado que contou com a participação de três discentes, que proveram *feedback* sobre o questionário, permitindo corrigir quaisquer problemas antecipadamente, além de servir como base de cálculo para o tempo médio de resposta. O formulário final contou com 34 perguntas, sendo distribuídos da seguinte forma:

Para as perguntas iniciais busca-se descobrir, de forma anônima, qual o perfil do discente. Entre os dados coletados, tem-se a faixa de idade, o curso, o período, os componentes curriculares e SOs utilizados pelos discentes.

Em seguida, tem-se a experiência prévia, aqui busca-se conhecer há quanto tempo o discente utiliza o seu SO, caso costuma trocar de SOs com alguma frequência, do tempo que estuda programação e as linguagens de programação já utilizadas. Avalia-se também a qualidade do SO dos discentes, analisando-o de forma granular em relação a diversas ópticas de qualidade, com base em ISOs/IECs, tocando em pontos como segurança, confiabilidade, privacidade, experiência de usuário (UX) e adaptabilidade.

Além disso, contam-se com perguntas relacionadas a problemas gerais e de compatibilidade. Aborda-se aqui a frequência com que o discente já enfrentou alguma dificuldade de compatibilidade no seu SO, o grau de comportamentos inesperados, a utilização de algum artifício para contornar esses problemas e como estes afetaram ou não seu aprendizado de

programação. O questionário trata também das experiências subjetivas acerca do(s) SO(s), como: o discente acredita que seu sistema pode afetar o seu aprendizado de programação; algum aspecto dele já facilitou ou dificultou seu aprendizado; o discente queria mudar algo no sistema para adequá-lo aos estudos de programação.

Ademais, mede-se o emocional do discente em relação ao seu SO, com respeito ao quanto ele se sente confiante para explorar novos conceitos de programação, a satisfação em relação ao seu SO, além de situações em que o SO motiva ou não o discente a continuar usando aquele sistema para programar e qual seria o SO mais adequado para programar e a razão de sua escolha.

Por fim, o discente avalia se os materiais didáticos utilizados em sala de aula são compatíveis com seu SO, quais os SOs utilizados em sala de aula, se gostaria que fossem disponibilizados outros tipos de SO e se, na última questão, considera que o SO influencia no entendimento de como o código interage com o *hardware* e *software*.

Quadro 1 - Estrutura do questionário aplicado aos discentes

(continua)

Seções	Tipo	Questão Apresentada
Perfil do discente	Múltipla Escolha ⁴	1. Qual é sua faixa de idade?
	Múltipla Escolha (BITI, BICT, EC, ES)	2. Qual é o seu curso?
	Múltipla Escolha (do 1º período até o 10º período em diante)	3. Em qual período do curso você está atualmente?

⁴Opções disponibilizadas: Até 18 (dezoito) anos; entre 21 (vinte um) e 22 (vinte e dois) anos; entre 23 (vinte e três) e 24 (vinte e quatro) anos; entre 25 (vinte e cinco) e 26 (vinte e seis) anos; entre 27 (vinte e sete) e 28 (vinte e oito) anos; mais de 28 (vinte e oito anos)).

Quadro 1 - Estrutura do questionário aplicado aos discentes

(continuação)

Componentes curriculares e SO	Múltipla Escolha ⁵	4. Informe sua média final e SO(s) usado(s) nas seguintes componentes curriculares.
Experiência Prévia	Múltipla Escolha (Menos de um ano; entre um e dois anos; entre três e quatro anos; entre cinco e seis anos; entre sete e oito anos; mais de oito anos)	5. Há quanto tempo você utiliza o(s) SO(s) marcado(s) nas questões anteriores como ambiente principal de desenvolvimento?
	Múltipla Escolha (Quase sempre; Frequentemente; Às vezes; Raramente)	6. Caso utilize mais de um SO, com que frequência você precisa alternar entre eles para realizar suas atividades de programação?
	Múltipla Escolha (menos de um ano; entre um e dois anos; entre três e quatro anos; entre cinco e seis anos; entre sete e oito anos; mais de oito anos)	7. Há quanto tempo você estuda programação?
	Múltipla Escolha (C; Java, JavaScript; Python; Outra opção)	8. Quais linguagens de programação você já utilizou?

⁵Opções disponibilizadas para as médias: Abaixo da média final, exatamente na média final, acima da média final e componente curricular ainda não cursado. Tem-se que a média final é igual a sete e menor nota em que um discente pode concluir um componente curricular é igual à 5.

Componentes curriculares analisados: Algoritmos, Análise e Projetos De Sistemas Orientados a Objetos, Arquitetura de *Software*, Banco de Dados, Computação Gráfica, Estrutura de Dados I, Estrutura de Dados II, Gerência de Configuração e Mudanças, Laboratório de Algoritmos, Laboratório de Estrutura de Dados II, Laboratório de Estruturas de Dados I, Manutenção de *Software*, Métodos Formais de Engenharia De *Software*, Processo de *Software*, Programação Concorrente e Distribuída, Programação Orientada a Objetos, Projeto Detalhado de *Software*, Segurança e Dependabilidade, Sistemas Digitais, Sistemas Distribuídos, Sistemas Inteligentes e Teste de *Software*.

Opções disponibilizadas para os SOs: Windows, macOS, Ubuntu, Linux Mint, Pop OS, Fedora, além de uma opção em aberto em que o participante poderia escolher outro SO não listado.

Quadro 1 - Estrutura do questionário aplicado aos discentes

(continuação)

Avaliação do Sistema Operacional	Múltipla Escolha (Windows; macOS; Ubuntu; Linux Mint; Pop OS; Fedora; Outra opção)	9. Qual é seu SO principal?
	Escala 1-10 (um a dez)	10. O quão importante você considera o SO para o seu desempenho nos componentes curriculares de programação?
	Escala: Péssimo, Ruim, Regular, Bom, Ótimo	11. Em relação ao seu SO avalie os seguintes aspectos: Capacidade de interação (usabilidade): Apropriabilidade, Reconhecível, Aprendibilidade, Operabilidade, Proteção contra Erro do Usuário, Engajamento do Usuário, Inclusividade
	Escala 1-10 (um a dez)	12. Quão fácil é realizar suas atividades de programação no seu SO?
	Escala: Péssimo, Ruim, Regular, Bom, Ótimo	13. Em relação ao seu SO avalie os seguintes aspectos: Confiabilidade Percebida: Maturidade, Disponibilidade, Tolerância a Falhas, Recuperabilidade, Conformidade com Expectativas

Quadro 1 - Estrutura do questionário aplicado aos discentes

(continuação)

	Escala 1-10 (um a dez)	14. Com que frequência você precisa interromper seu trabalho por problemas inesperados do SO?
	Escala: Péssimo, Ruim, Regular, Bom, Ótimo	15. Em relação ao seu SO avalie os seguintes aspectos sobre UX: Emocional, Estética Percebida, Significado Pessoal, Engajamento Sustentado, Satisfação Holística
	Subjetiva	16. Descreva uma emoção que você sente ao usar este SO para programar:
	Escala: Péssimo, Ruim, Regular, Bom, Ótimo	17. Em relação ao seu SO avalie os seguintes aspectos: Confiança no Sistema: Segurança Percebida, Transparência Operacional, Controle Percebido, Previsibilidade, Responsabilidade
	Escala 1-10 (Confia Totalmente; Não Confia)	18. O quanto você confia que este SO protegerá seu trabalho contra perda ou erro?

Quadro 1 - Estrutura do questionário aplicado aos discentes

(continuação)

	Escala: Péssimo, Ruim, Regular, Bom, Ótimo	19. Em relação ao seu SO avalie os seguintes aspectos: Adaptabilidade Contextual: Adaptabilidade, Coexistência, Interoperabilidade, Apropriabilidade Funcional, Portabilidade de Conhecimento
	Escala 1-10 (um a dez) (Nunca; Sempre)	20. Seu SO atende à evolução das suas necessidades como programador/discente?
Problemas Técnicos e de Compatibilidade	Escala 1-10 (um a dez) (Nunca; Sempre)	21. Com que frequência você encontra erros ou comportamentos inesperados relacionados especificamente ao SO?
	Múltipla Escolha (Sim; Não)	22. Você já deixou de usar uma ferramenta, biblioteca ou framework por incompatibilidade com seu SO?
	Múltipla Escolha (Sim; Não)	23. Você utiliza máquinas virtuais (Docker, <i>containers</i>) para contornar limitações do SO?
Experiências Qualitativas	Múltipla Escolha (Sim, Não)	24. Você acredita que o SO pode afetar o aprendizado de programação?

Quadro 1 - Estrutura do questionário aplicado aos discentes

(continuação)

	Subjetiva	26. Se você pudesse mudar algo no SO principal que utiliza para torná-lo mais adequado ao aprendizado de programação, o que seria?
Motivação, Confiança e Satisfação	Escala 1-10 (um a dez) (Não afeta; Afeta totalmente)	27. O SO que você utiliza afeta a sua confiança para explorar novos conceitos de programação?
	Escala 1-10 (um a dez) (Nada satisfeito; Totalmente satisfeito)	28. Quão satisfeito você está com o SO que utiliza para programar?
	Subjetiva	29. O que mais te motiva ou desmotiva a continuar usando seu SO atual para programar?
Preferências e Recomendações	Múltipla Escolha (Windows; macOS; Ubuntu; Linux Mint; Pop OS; Fedora; Outra opção)	30. Qual SO você considera ser o mais adequado para o aprendizado de programação?
Contexto Institucional e Pedagógico	Escala 1-10 (Nunca; Sempre)	31. Os materiais didáticos que você utiliza são compatíveis com o SO que você tem acesso?
	Múltipla Escolha (Windows; macOS; Ubuntu; Linux Mint; Pop OS; Fedora; Outra opção)	32. Qual(is) o(s) SO(s) utilizado(s) nas suas aulas práticas?

Quadro 1 - Estrutura do questionário aplicado aos discentes

(conclusão)

	Múltipla Escolha (Sim; Não)	33. Você gostaria que fosse oferecido suporte a diferentes SOs?
Interação <i>Hardware-Software</i>	Múltipla Escolha (Sim; Não)	34. Você acredita que o SO influencia sua capacidade de entender como o código interage com o <i>hardware</i> ?

Fonte: Autoria Própria (2026).

3.1 Avaliação do Sistema Operacional

O questionário também contou com perguntas associadas aos diferentes normativos (ISO/IEC) associados à qualidade de produtos digitais (Quadro 2).

Quadro 2 - Estrutura do questionário aplicado aos discentes

(continua)

Avaliação do Sistema Operacional	Definição (Adaptada para o Contexto da Pesquisa)	ISO/IEC (Referência)	Fontes / Normas Complementares
Construto 1: Capacidade de Interação (Interação Humano-Computador)			
Capacidade de Interação	Grau em que o SO pode ser operado por discentes de TI para alcançar objetivos acadêmicos com eficácia, eficiência e satisfação.	ISO/IEC 25010:2023, Seção 3.4	ISO/IEC 25022:2016 (Qualidade em Uso)
Apropriabilidade Reconhecível	O usuário identifica rapidamente se o SO atende às suas necessidades acadêmicas e de desenvolvimento.	ISO/IEC 25010:2023, Seção 3.4.1	ISO 9241-11:2018 (Ergonomia)
Aprendibilidade	Tempo e esforço necessários para dominar tarefas básicas e avançadas no ambiente (ex.: terminal, atalhos, configuração).	ISO/IEC 25010:2023, Seção 3.4.2	ISO/IEC 25022:2016, Seção 5.2 (Eficiência)

Quadro 2 - Estrutura do questionário aplicado aos discentes

(continuação)

Operabilidade	Facilidade de controle, navegação e execução de comandos no dia a dia acadêmico (ex.: gestão de arquivos, depuração).	ISO/IEC 25010:2023, Seção 3.4.3	ISO 9241-11:2018 (Eficácia e Eficiência)
Proteção contra Erro do Usuário	Mecanismos que previnem ou mitigam ações incorretas (ex.: confirmações de exclusão, desfazer, <i>sandboxing</i>).	ISO/IEC 25010:2023, Seção 34.4	ISO/IEC 25022:2016, Seção 5.4 (Liberdade de Risco)
Engajamento do Usuário	Grau em que a interface motiva a exploração e o uso prolongado do sistema para fins acadêmicos.	ISO/IEC 25010:2023, Seção 3.4.5	ISO 9241-210:2010 (Design Centrado no Usuário)
Inclusividade	Acessibilidade para usuários com diferentes níveis de habilidade técnica, preferências e necessidades.	ISO/IEC 25010:2023, Seção 3.4.6	ISO 9241-171:2008 (Diretrizes de Acessibilidade)
Construto 2: Confiabilidade Percebida (Estabilidade do Sistema)			
Confiabilidade Percebida	Grau em que o usuário percebe que o SO executa funções sob condições acadêmicas normais, sem falhas inesperadas.	ISO/IEC 25010:2023, Seção 3.5	ISO/IEC 25022:2016 (Qualidade em Uso)
Maturidade	Frequência percebida de travamentos, bugs ou comportamentos inesperados durante tarefas críticas (compilação, simulações).	ISO/IEC 25010:2023, Seção 3.5.1	ISO/IEC 25022:2016, Seção 5.1 (Eficácia)
Disponibilidade	Percepção de que o SO está "pronto para uso" quando necessário, sem nenhuma inatividade inesperada.	ISO/IEC 25010:2023, Seção 3.5.2	ISO/IEC 25022:2016, Seção 5.2 (Eficiência)
Tolerância a Falhas	Capacidade percebida de recuperação após erros de aplicação, configuração ou falhas de <i>hardware</i> .	ISO/IEC 25010:2023, Seção 3.5.3	ISO/IEC 25022:2016, Seção 5.4 (Liberdade de Risco)

Quadro 2 - Estrutura do questionário aplicado aos discentes

(continuação)

Recuperabilidade	Facilidade para restaurar o estado de trabalho após interrupções (ex.: reinicialização, falha na aplicação).	ISO/IEC 25010:2023, Seção 3.5.4	ISO 9241-11:2018 (Eficácia)
Conformidade com Expectativas	Comportamento do sistema alinhado ao modelo mental do usuário técnico (ex.: previsibilidade de comandos, consistência).	ISO/IEC 25022:2016, Seção 3.5 (Adaptado)	ISO 9241-11:2018 (Satisfação)
Construto 3: Experiência do Usuário (UX)			
Experiência do Usuário (UX)	Percepções e respostas resultantes do uso do SO, abrangendo aspectos emocionais, estéticos e de valor no contexto acadêmico.	ISO 9241-210:2010, Cláusula 3.15	ISO/IEC 25022:2016 (Qualidade em Uso)
Emocional	Sentimentos evocados pelo uso do SO: confiança, frustração, prazer, autonomia, controle.	ISO 9241-210:2010, Cláusula 3.15	ISO/IEC 25022:2016, Seção 5.3 (Satisfação)
Estética Percebida	Clareza visual, organização, consistência e harmonia da interface gráfica e terminal.	ISO/IEC 25010:2023, Seção 3.4.5	ISO 9241-210:2010 (Design Centrado no Usuário)
Significado Pessoal	Alinhamento do SO com valores e identidade profissional dos discentes (ex.: <i>software</i> livre, privacidade, filosofia de código-aberto).	ISO 9241-210:2010, Cláusula 3.15	ISO/IEC 25022:2016, Seção 5.3 (Satisfação)
Engajamento Sustentado	Disposição para explorar funcionalidades além do mínimo necessário (ex.: personalização, <i>scripts</i> , automação).	ISO/IEC 25010:2023, Seção 4.4.5	ISO 9241-210:2010 (Iteração e Envolvimento)
Satisfação Holística	Avaliação global que integra aspectos funcionais, emocionais e contextuais do uso do SO.	ISO/IEC 25022:2016, Seção 5.3	ISO 9241-11:2018 (Satisfação)
Construto 4: Confiança no Sistema (Segurança e Controle)			

Quadro 2 - Estrutura do questionário aplicado aos discentes

(conclusão)

Confiança no Sistema	Grau em que o discente acredita que o sistema protegerá seus dados, respeitará sua intenção e permitirá controle adequado.	ISO/IEC 25010:2023, Seção 3.6	ISO/IEC 25022:2016 (Liberdade de Risco)
Segurança Percebida	Crença na proteção de dados contra acesso não autorizado, perda acidental ou vazamento.	ISO/IEC 25010:2023, Seção 3.6.1 e 3.6.2	ISO/IEC 25022:2016, Seção 5.4 (Liberdade de Risco)
Transparência Operacional	Compreensão do discente sobre ações do sistema, logs, permissões e motivação de comportamentos automáticos.	ISO/IEC 25010:2023, Seção 3.6.4	ISO 9241-11:2018 (Eficácia e Satisfação)
Controle Percebido	Sensação de autonomia para personalizar, auditar e reverter configurações (ex.: sudo, permissões, kernel).	ISO/IEC 25010:2023, Seção 3.6.3 e 3.6.5	ISO 9241-210:2010 (Envolvimento do Usuário)
Previsibilidade	Comportamento do sistema consistente com expectativas técnicas do usuário (ex.: resposta a comandos, estabilidade).	ISO/IEC 25010:2023, Seção 3.5R.1	ISO/IEC 25022:2016, Seção 5.4 (Liberdade de Risco)
Responsabilidade	Clareza sobre quem (ou o que) é responsável por falhas, decisões do sistema ou ações executadas (auditoria, logs).	ISO/IEC 25010:2023, Seção 3.6.1	ISO/IEC 27001:2022 (Gestão de Segurança da Informação)

Fonte: Autoria Própria (2026).

4 RESULTADOS

O questionário foi aplicado no período de 28 de abril até 13 de maio de 2026 e contou com 38 (trinta e oito) respostas, nas quais três dos quatro cursos (BITI, ES e EC) foram contemplados nas respostas. No processo de análise dos dados, as escalas originais (1-10) foram padronizadas para a escala Likert (1-5). Além disso, os resultados foram organizados para melhor apresentação dos resultados obtidos.

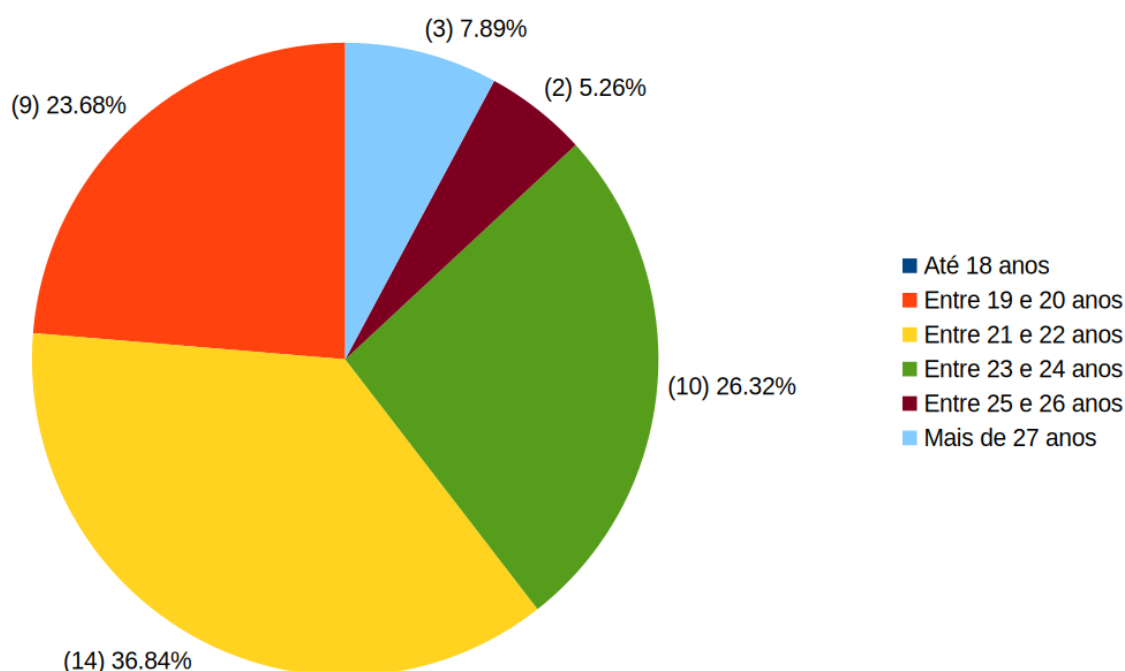
Durante esse capítulo, há a apresentação de gráficos, advindos da pesquisa, que são compostos por duas informações, a primeira delas está entre aspas e contém o número de discentes que escolheram alguma opção entre as questões de múltipla escolha e em seguida há a respectiva porcentagem desse número (ex.: (10) 10,00% ou 10,00% (10)). O mesmo também se aplica a textos corridos.

Ademais, dentre as perguntas nas quais continham uma seção de comentários, quando essas foram analisadas, percebeu-se que algumas dessas respostas não correspondiam ao que foi questionado, tais como, escrever por extenso que não havia nada a comentar (ex.: “Sem comentários”, “Não tem”), colocar um ponto como resposta (ex.: “.”) ou ainda iniciar a escrita e não concluí-la (ex.: “Porque”). No caso dessas ocorrências, a decisão tomada foi de retirá-las do escopo daquela questão e seguir para a análise apenas com os dados normalizados.

4.1 Perfil dos Discentes

No tocante ao perfil dos discentes, referente à faixa etária dos discentes, a distribuição das respostas é apresentada na Figura 1.

Figura 1 - Faixa etária dos discentes

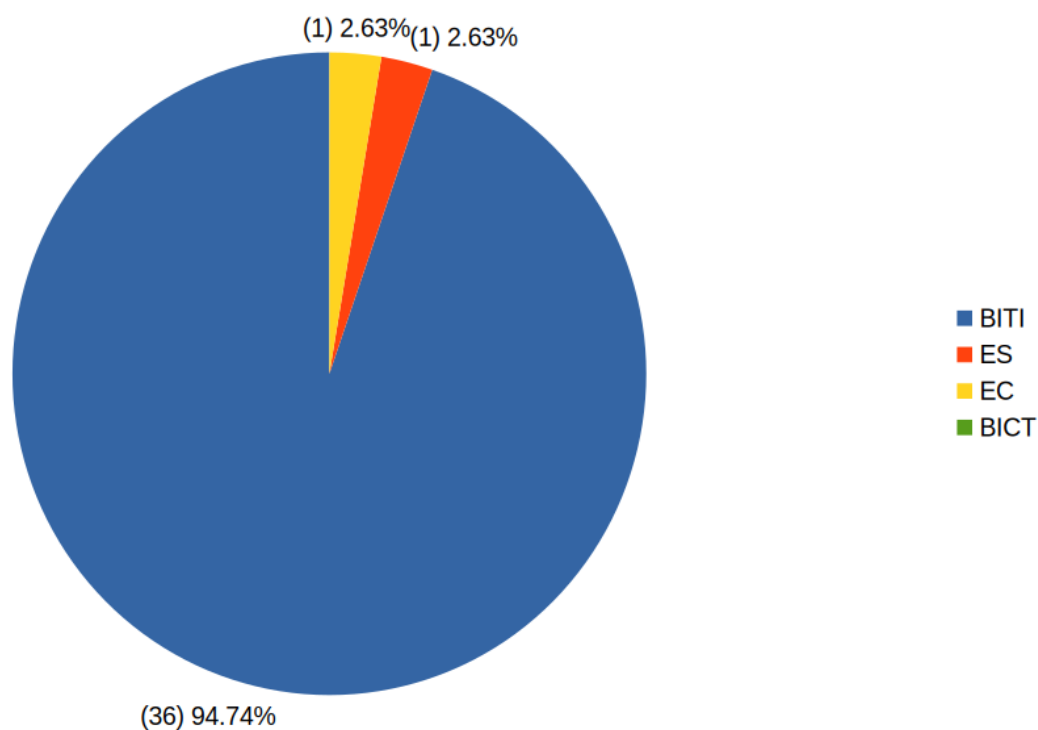


Fonte: Autoria Própria (2026).

Dentre os discentes analisados, a maioria se encontra na faixa etária entre 21 e 22 anos (36,84%), seguido por 23 e 24 anos (26,32%), 19 e 20 anos (23,68%), mais de 27 anos (7,89%), 25 e 26 anos (5,26%).

Na Figura 2, pode-se observar a distribuição desses discentes em relação aos cursos matriculados.

Figura 2 - Cursos dos discentes

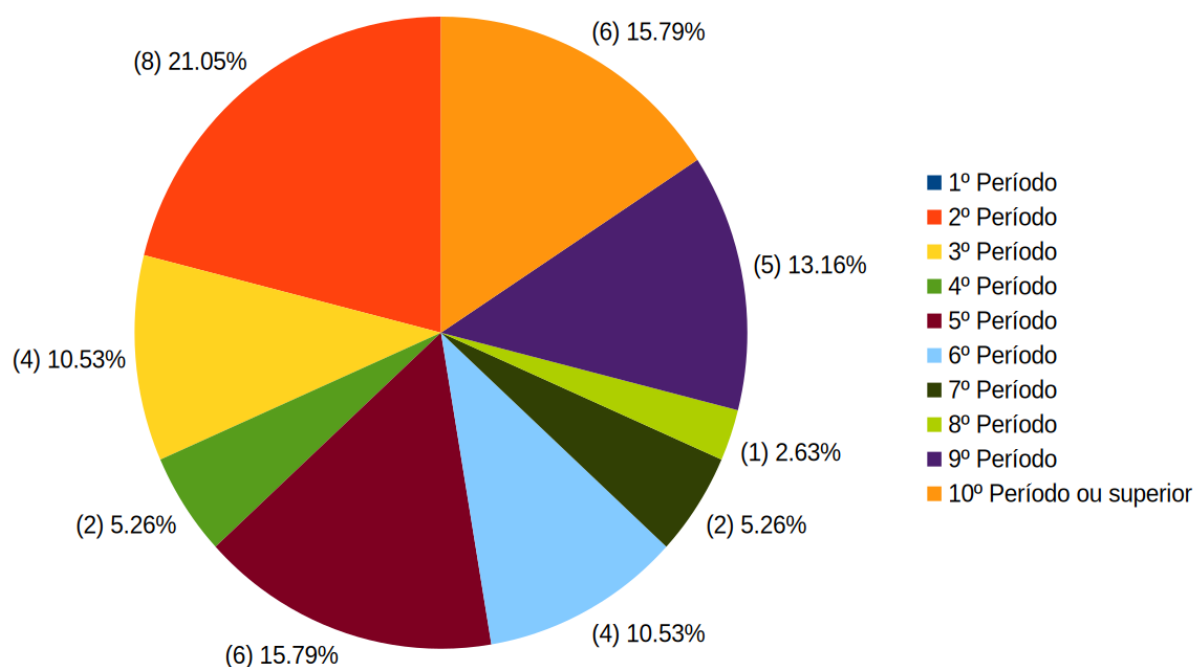


Fonte: Autoria Própria (2026)

Nesse gráfico acima, tem-se a relação entre os cursos pesquisados. Dentre os discentes analisados, grande parte é do curso de BITI (94,74%), seguido por ES e EC com a mesma porcentagem (2,63%).

Na próxima pergunta, é questionado o período dos discentes nos seus cursos, cuja distribuição é apresentada na Figura 3.

Figura 3 - Período dos discentes

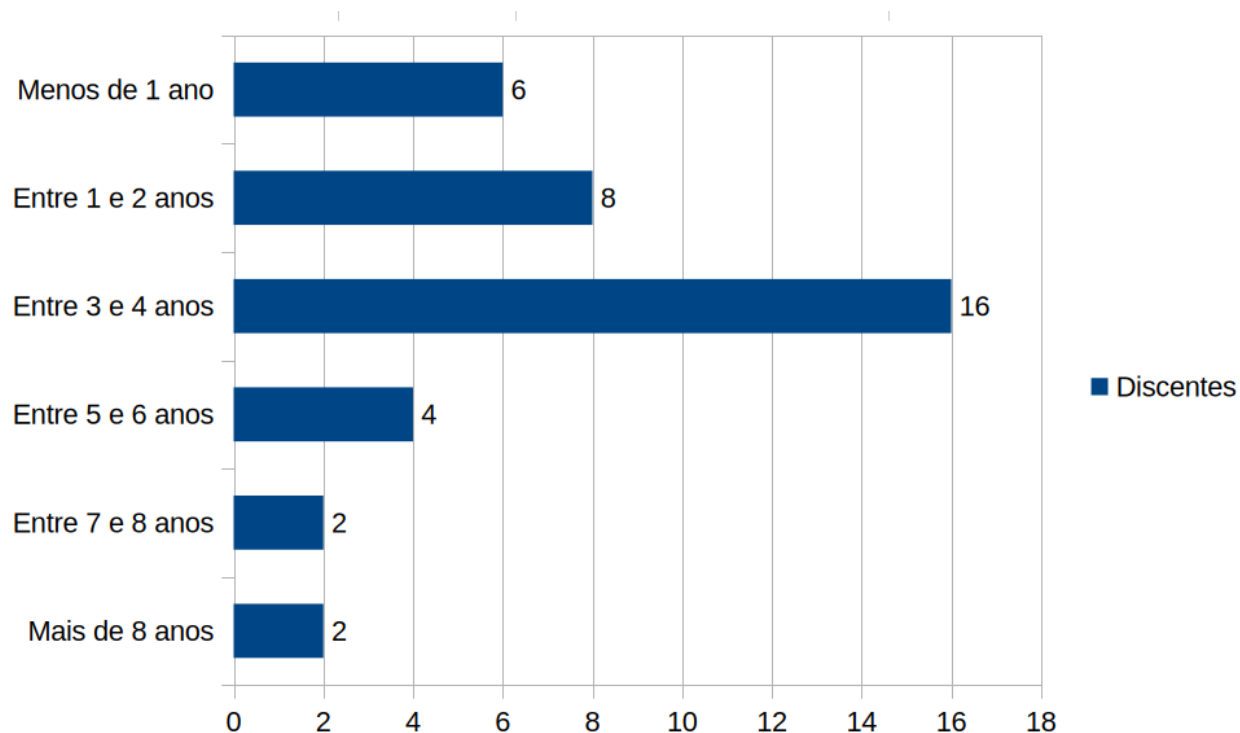


Fonte: Autoria Própria (2026)

Dentre as respostas obtidas, a maioria delas está no segundo período, com uma porcentagem de 21,05%, seguido pelo 5º período e 10º período ou mais, ambos com 15,79%, 9º período 13,16%, 3º e 6º períodos com 10,53%, 4º e 7º períodos com 5,26%, 8º período com 2,63% e, por fim, nenhum dos discentes afirmou estar no primeiro período.

Referente ao tempo de estudo de programação, a distribuição das respostas é apresentada na Figura 4.

Figura 4 - Tempo de estudo de programação



Fonte: Autoria Própria (2026)

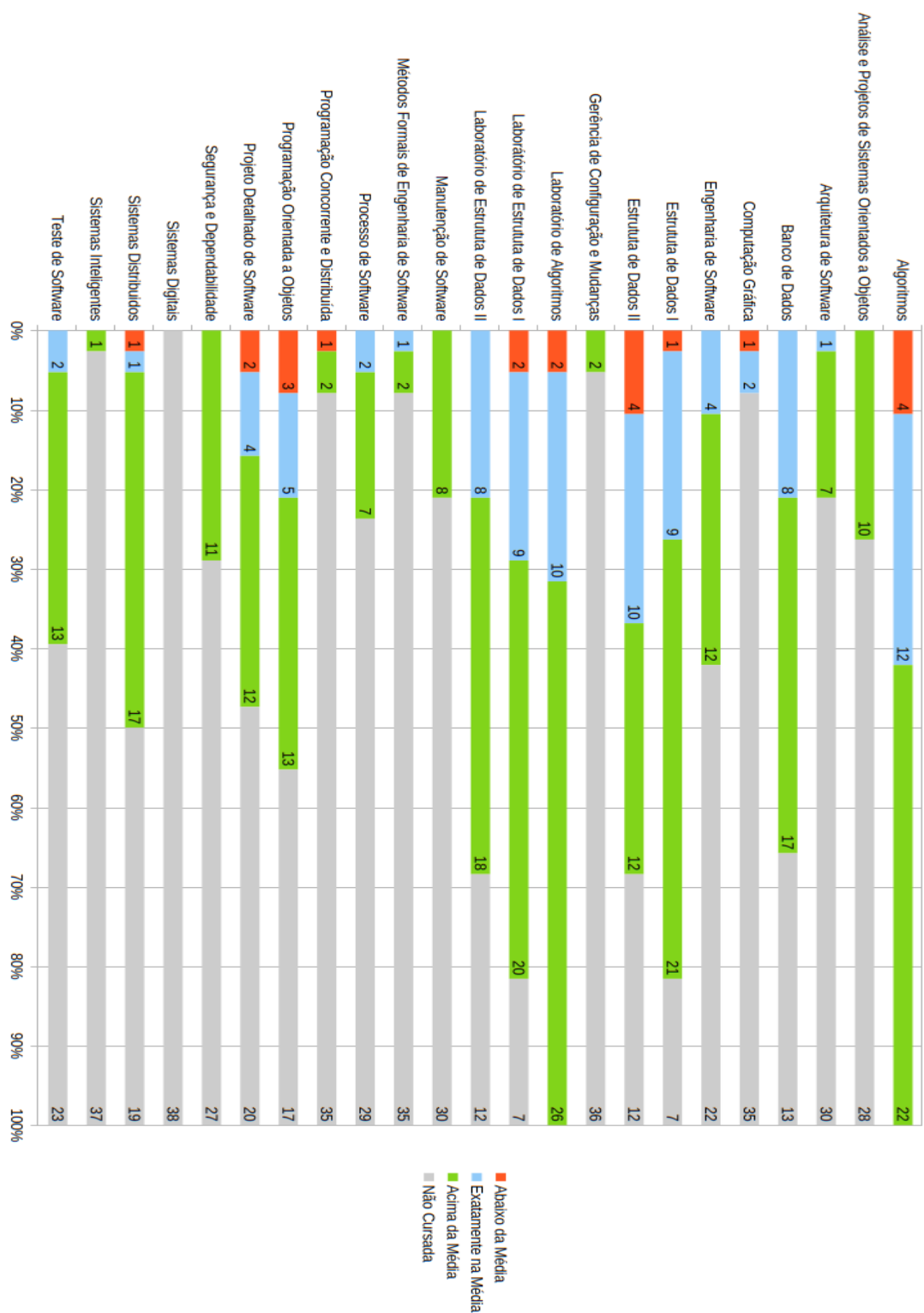
Com base no gráfico acima, tem-se que maioria, 42,11% (16) dos discentes estuda programação entre três a quatro anos, seguido por 21,05% (8) entre um a dois anos, 15,79% (6) com menos de um ano, 10,53% (4) entre cinco a seis anos, 5,26% (2) estão entre sete e oito anos e, por fim, 5,26% (2) estão a mais do que oito anos.

Em suma, o público geral que a pesquisa mais atingiu foram discentes de BITI (94,74%), entre 21 e 22 anos (36,84%), entre 3 a 4 anos de estudo de programação (42,11%), que, no tempo de execução desta pesquisa, estavam no 2º período (21,05%).

4.2 Média Final dos Discentes

O gráfico da Figura 5 apresenta a média final dos discentes nos componentes curriculares relacionados à programação de acordo com a grade do curso de BITI. Ao invés de utilizar valores brutos, os discentes foram questionados se sua média final foi abaixo, exatamente ou acima da média final. Caso não tenham participado do referido componente curricular, o discente poderia informar em sua resposta. Para os gráficos seguintes, são apresentados os dados da Figura 5.

Figura 5 - Média final dos discentes nas componentes curriculares analisadas



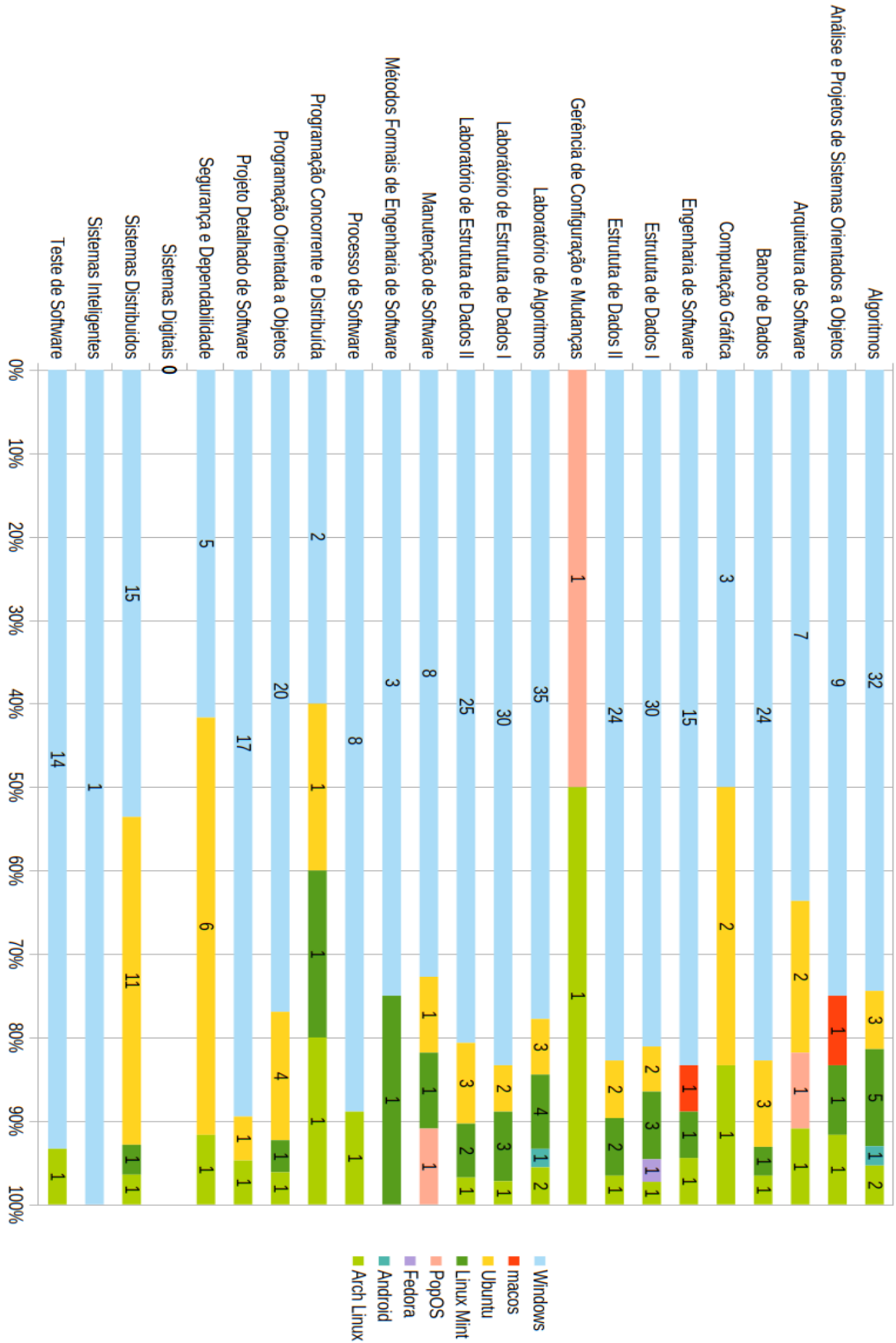
Fonte: Autoria Própria (2026)

Com base nos dados do gráfico, a maioria dos discentes analisados ficou acima da média final (69,62%), seguido por exatamente na média final (24,40%) e abaixo da média final (5,98%). Os dados dos discentes que não cursaram ou concluíram as componentes curriculares não foram incluídos no gráfico. No geral, pode-se concluir que os discentes estão apresentando bom desempenho em relação às notas, com uma taxa de 94,02% de aprovação direta a partir da 3ª unidade.

4.3 SOs Utilizados por Componente Curricular

Para cada pergunta do questionário a respeito da média final do discente em um componente curricular, caso este não escolha a opção de ainda não tê-lo cursado, então será perguntado ao participante qual SO tinha utilizado no mesmo, por isso, existem menos respostas nesta seção. Na Figura 6, tem-se a correlação entre as componentes curriculares e os SO utilizados nela.

Figura 6 - SOs utilizadas em cada componente curricular analisado



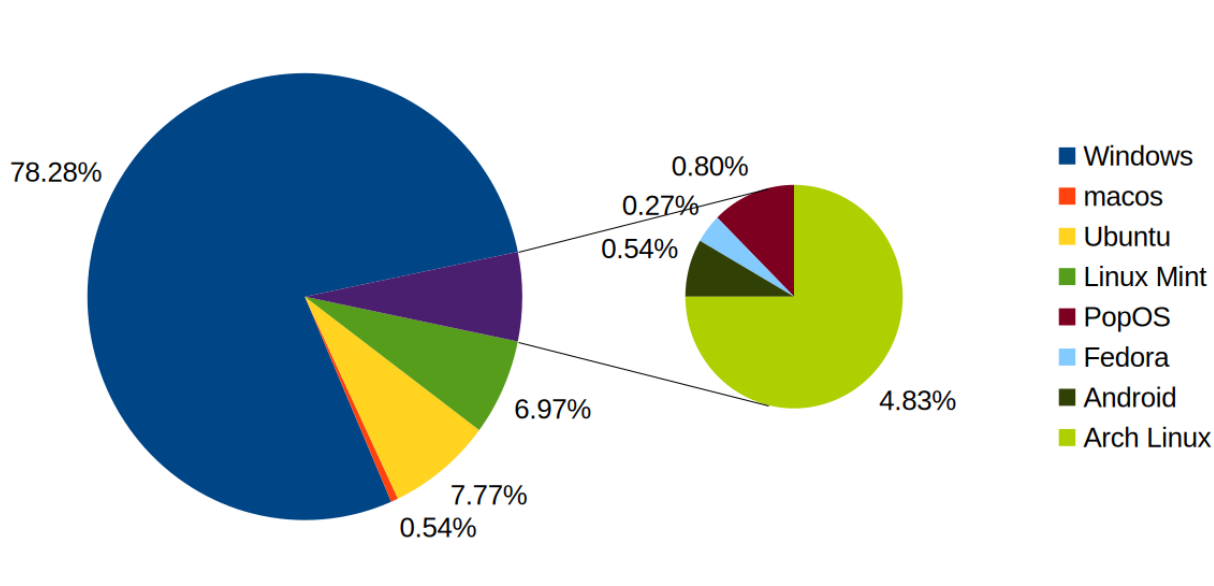
Fonte: Autoria Própria (2026)

O gráfico apresenta uma grande tendência no uso do Windows em quase todas as componentes curriculares analisadas, apenas uma pequena fração dos resultados diz ter utilizado alguma distribuição Linux, especialmente o Ubuntu, e com fração menor ainda tem-se o macOS.

Em contraste à dominância do Windows, alguns componentes curriculares se destacam, como: Sistemas Distribuídos, Segurança e Dependabilidade, Algoritmos e Laboratório de Algoritmos.

Em seguida, na Figura 7, tem-se a correlação em porcentagem dos dados coletados.

Figura 7 - SOs utilizados nos componentes curriculares analisados⁶



Fonte: Autoria Própria (2026)

Observa-se que o Windows teve uma taxa de uso de 78,28%, seguido por Ubuntu (7,77%), Linux Mint (6,97%), Arch Linux (4,83%), Pop OS (0,80%), Android (0,42%), macOS(0,42%), Fedora (0,21%). Ao distribuir os dados entre três grandes grupos, excluindo Android, tem-se: Windows (78,50%), Distribuições Linux (21,09%) e macOS (0,42%).

4.4 SOs e Média Final

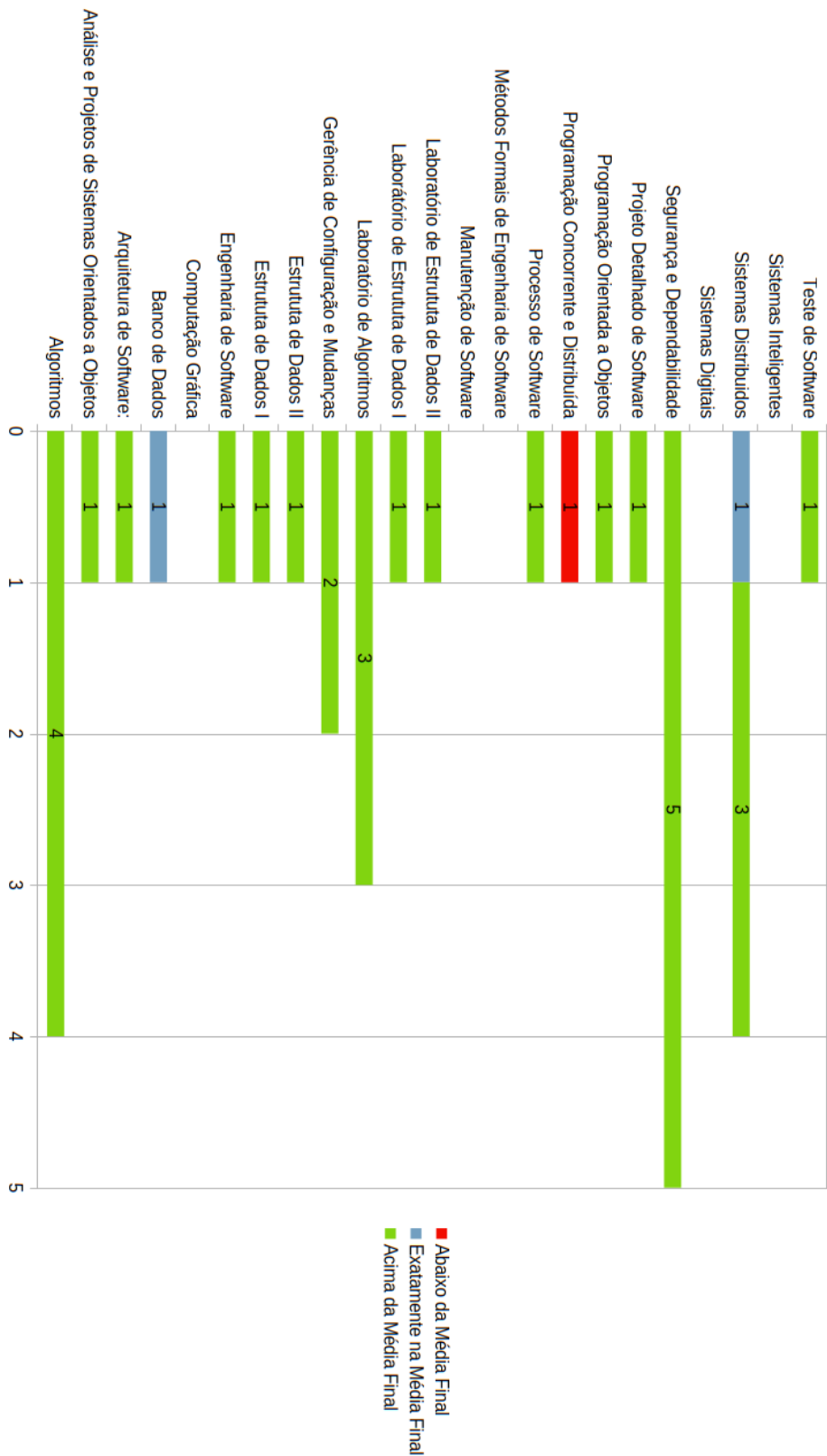
⁶ Distribuições Linux: Ubuntu, Linux Mint, Pop OS, Fedora, Arch Linux.

Os gráficos seguintes apresentam as relações entre a média final dos discentes nas componentes curriculares e o(s) SO(s) utilizado(s) nelas.

4.4.1 Distribuições Linux e a Média Final dos discentes

No gráfico apresentado na Figura 8, é mostrada a relação entre os discentes que dizem utilizar distribuições Linux e a sua média final nos componentes curriculares analisados.

Figura 8 - Distribuições Linux e a média final dos discentes



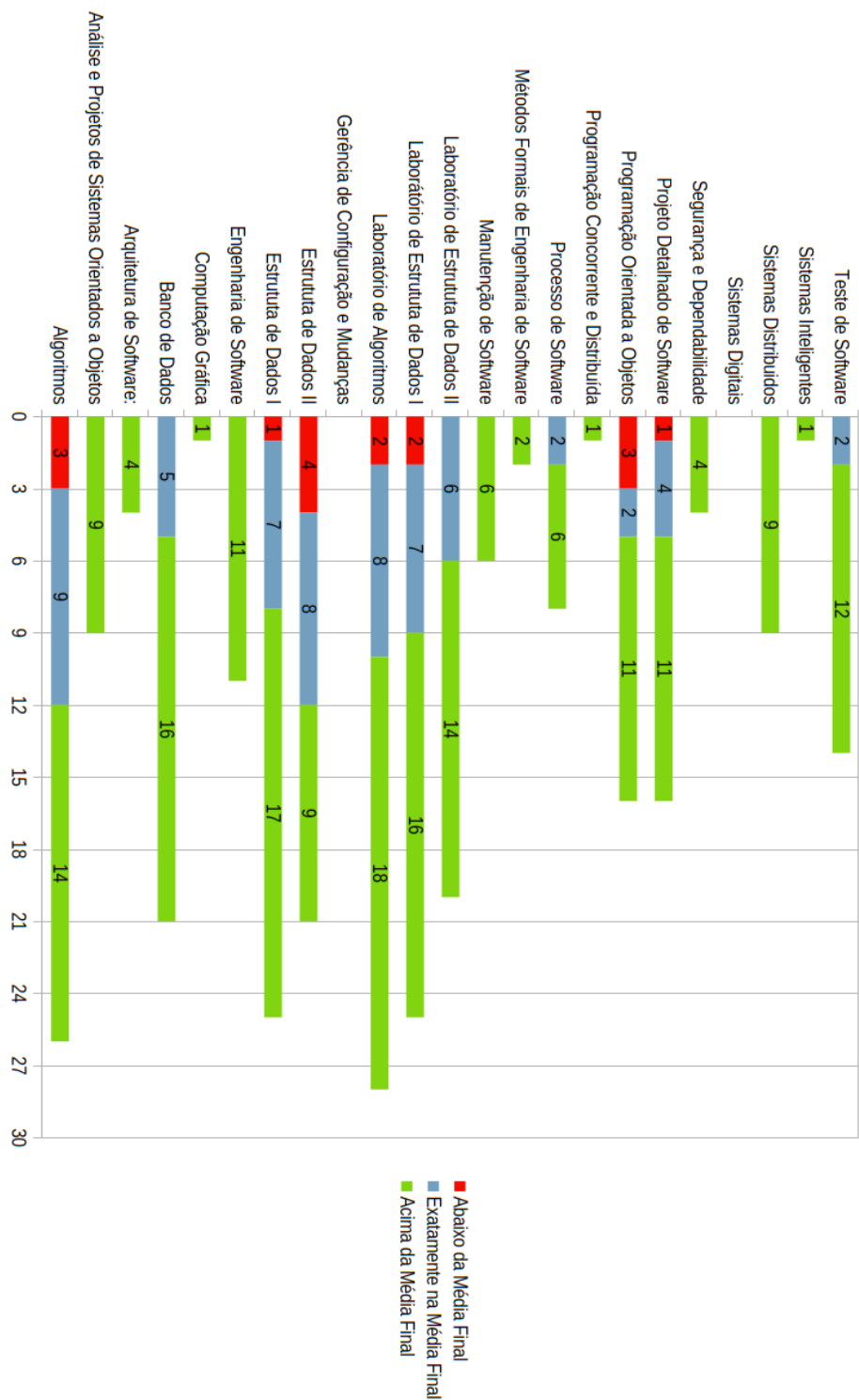
Fonte: Autoria Própria (2026)

Pode-se observar que dentre os discentes que dizem utilizar alguma distribuição Linux, 90,32%, estão acima da média final, 6,45% encontrando-se exatamente na média final e apenas 3,23% estão abaixo da média final. No geral, pode-se concluir que os discentes estão apresentando bom desempenho em relação às notas, com uma taxa de 96,77% de aprovação direta a partir da 3ª unidade.

4.4.2 Windows e a Média Final dos discentes

Por sua vez, no gráfico da Figura 9, é apresentada a relação entre os discentes que dizem utilizar Windows e a sua média final nos componentes curriculares analisados.

Figura 9 - Windows e a média final dos discentes



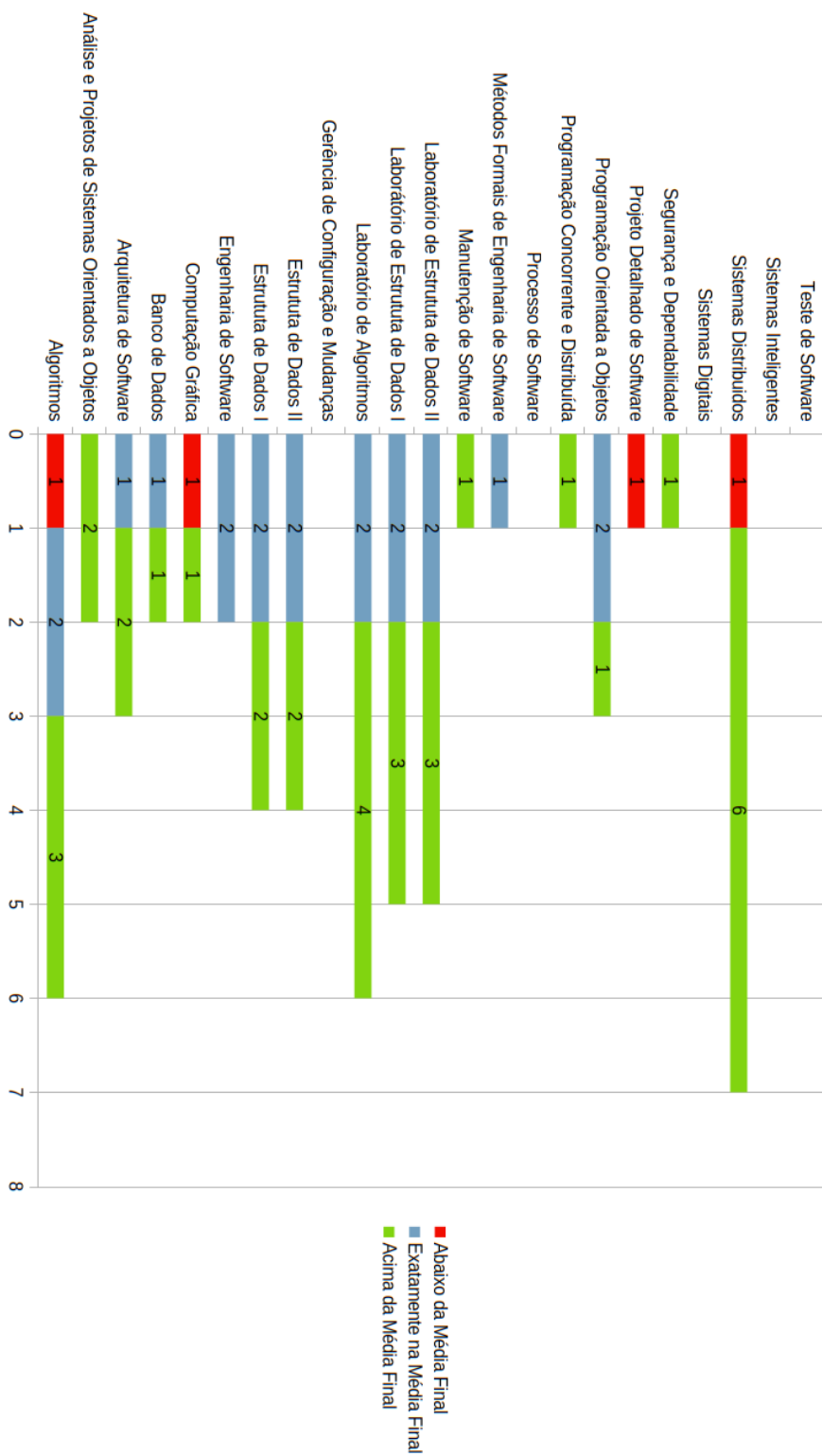
Fonte: Autoria Própria (2026)

Pode-se verificar que a maioria dos discentes que diz utilizar Windows, 71,64%, está acima da média final, 22,39% se encontram exatamente na média final e 5,97% ficaram abaixo da média final. No geral, pode-se concluir que os discentes estão apresentando bom desempenho em relação às notas, com uma taxa de 94,03% de aprovação direta a partir da 3ª unidade.

4.4.3 *Dual Boot* e a Média Final dos discentes

O gráfico da Figura 10 apresenta a relação entre os discentes que dizem utilizar *Dual Boot* (Windows, Distribuições Linux ou macOS) e a sua média final nos componentes curriculares analisados.

Figura 10 - *Dual Boot* (Windows, Distribuições Linux ou macOS) e a média final dos discentes



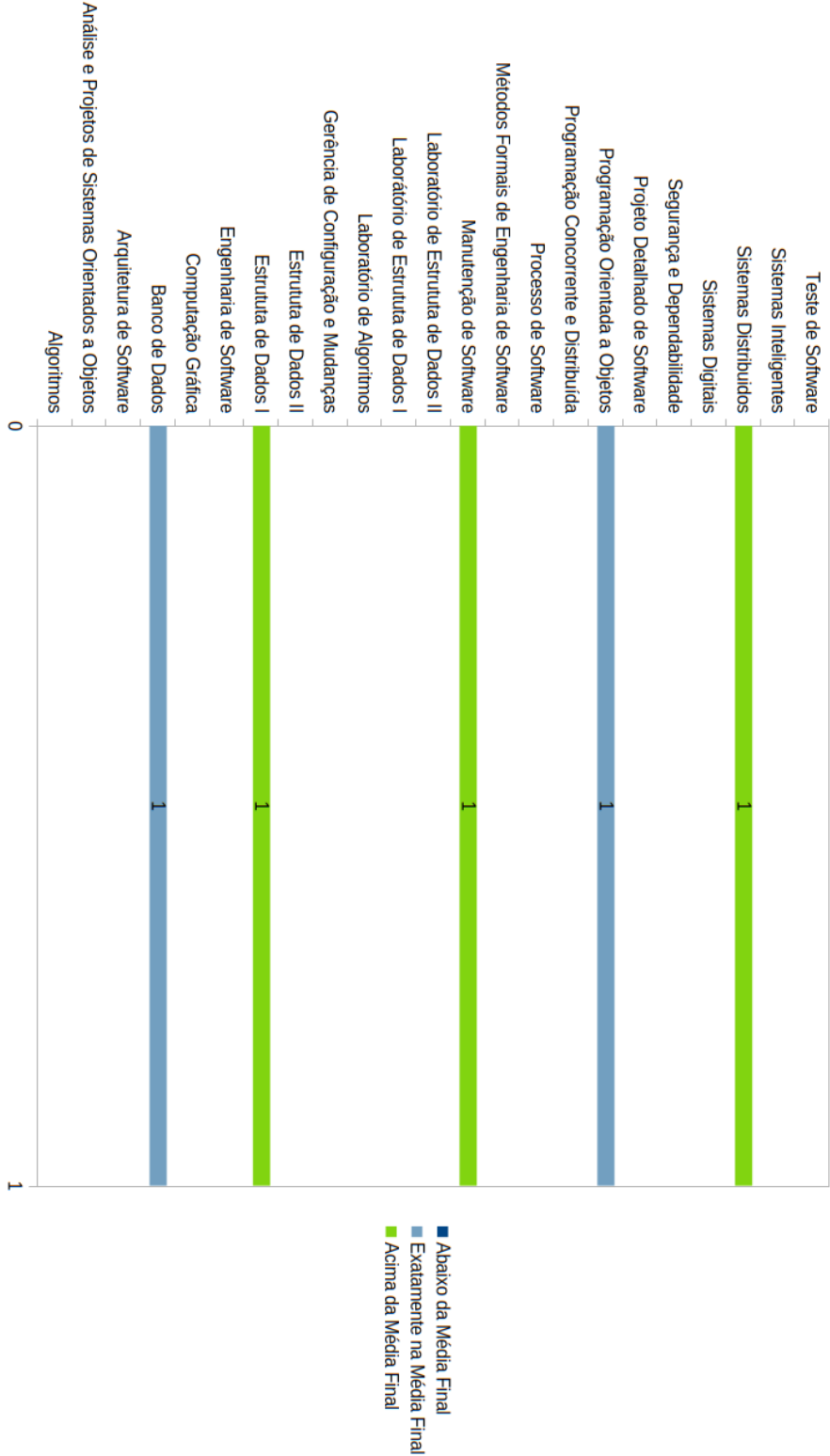
Fonte: Autoria Própria (2026)

Pode-se verificar que a maioria dos discentes que dizem utilizar *Dual Boot* (Windows, junto a alguma distribuição Linux ou macOS), teve os seguintes resultados: 58,93% estão acima da média final, 33,93% se encontram exatamente na média final, e apenas 7,14% estão abaixo da média final. No geral, pode-se concluir que os discentes estão apresentando bom desempenho em relação às notas, com uma taxa de 92,86% de aprovação direta a partir da 3ª unidade.

4.4.4 *Trial Boot* e a Média Final dos discentes

No tocante à relação entre os discentes que dizem utilizar *Trial Boot* (Windows e Distribuições Linux) e a sua média final nos componentes curriculares analisados, os dados são apresentados na Figura 11.

Figura 11 - *Trial Boot* (Windows e distribuições Linux) e média final dos discentes



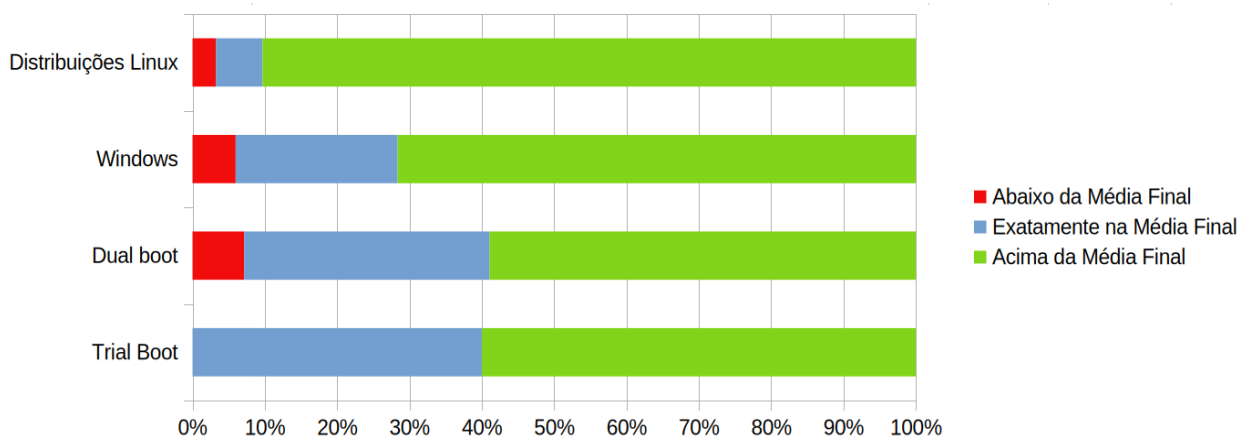
Fonte: Autoria Própria (2026)

Pode-se verificar que a maioria dos discentes que dizem utilizar *Trial Boot* (Windows, junto a distribuições Linux), teve os seguintes resultados: 60,00% estão acima da média final, 40,00% se encontram exatamente na média final; com base nos dados obtidos, ninguém ficou abaixo da média final, ou seja, todos foram aprovados.

4.4.5 SOs e a Média Final dos discentes

Por sua vez, o gráfico abaixo apresentado na Figura 12 relaciona todos os SOs utilizados junto à média final dos discentes.

Figura 12 - SOs e a média final dos discentes



Fonte: Autoria Própria (2026)

As porcentagens em relação aos SOs analisados são as seguintes: para as distribuições Linux, tem-se que 3,23% estão abaixo da média final, 6,45% estão exatamente na média final e 90,32% estão acima da média final. Para o Windows, tem-se que 5,97% estão abaixo da média final, 22,39% estão exatamente na média final e 71,64% estão acima da média final. Para *Dual Boot*, tem-se que 7,14% estão abaixo da média final, 33,93% estão exatamente na média final e 58,93% estão acima da média final. Para *Trial Boot*, tem-se que nenhum está abaixo da média final, 40% estão exatamente na média final e 60% estão acima da média final.

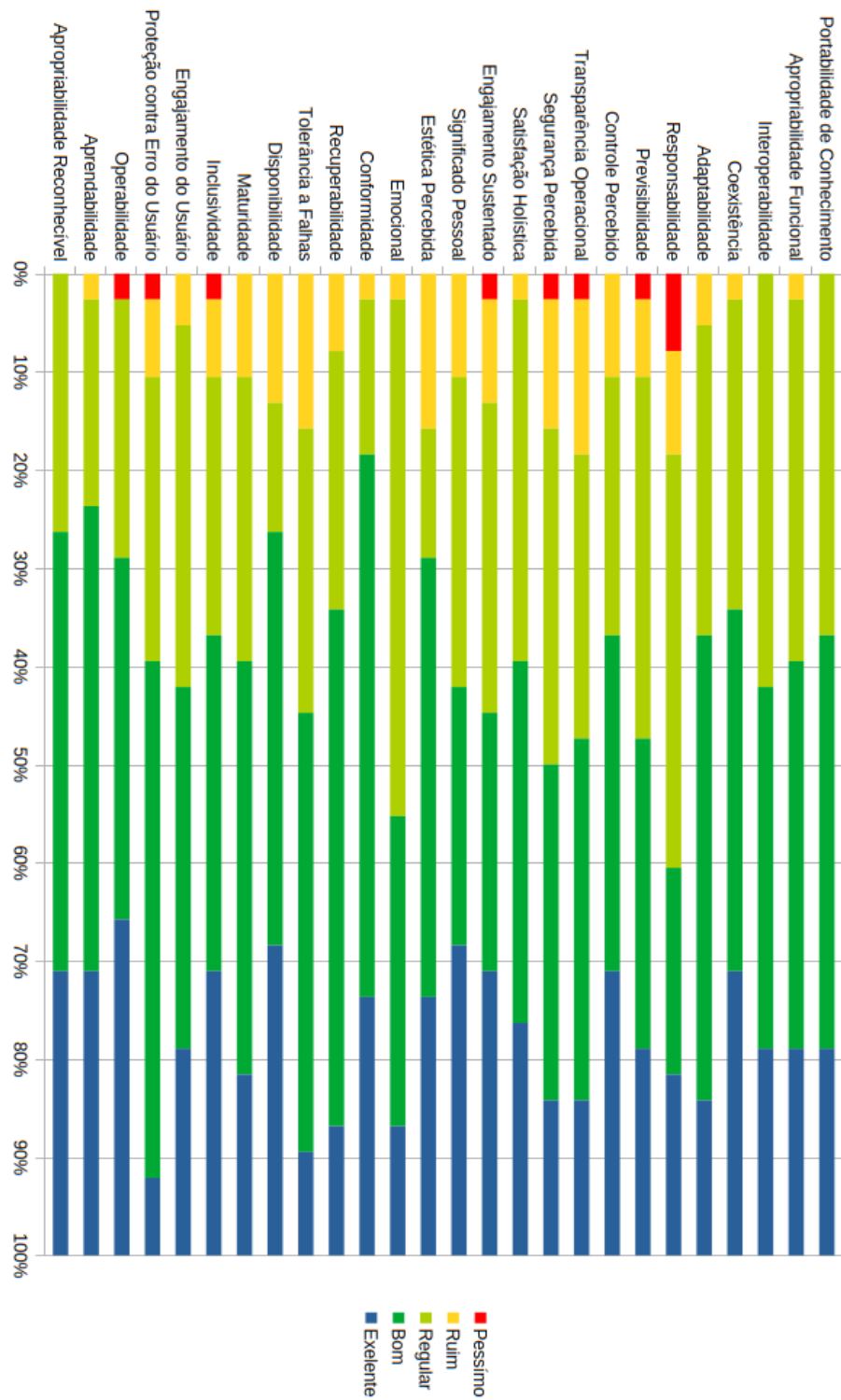
Em conclusão, com base nos resultados obtidos, tem-se que os discentes que utilizam distribuições Linux tiveram uma maior porcentagem de notas acima da média final do que todos os outros grupos (90,32%), seguido por Windows (71,64%), *Trial Boot* (60%) (Windows,

Distribuições Linux ou macOS) e *Dual Boot* (58,93%) (Windows, Distribuições Linux). Em relação aos discentes que utilizam *Trial Boot*, tem-se que esse grupo não obteve nenhuma taxa abaixo da média final.

4.5 Avaliação Geral dos SOs com base em ISO/IEC

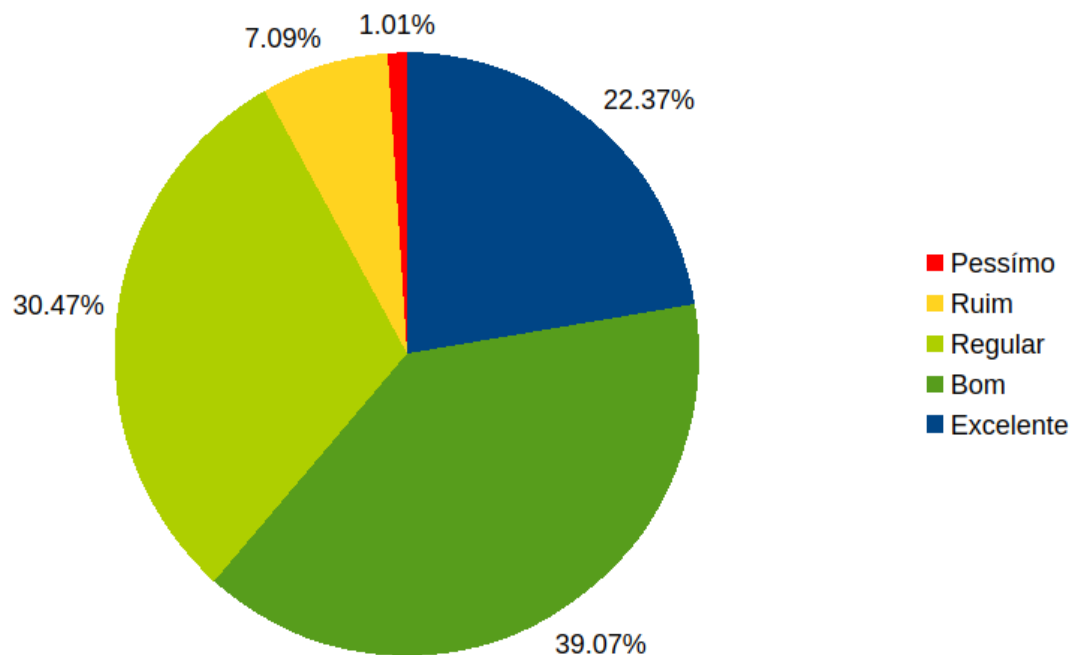
Na Figura 13 é apresentada a avaliação geral dos SOs com base nos aspectos de qualidade definidos nas ISOs/IECs. A Figura 14 relaciona esses valores em porcentagem.

Figura 13 - Avaliação do SO com base nas métricas ISO/IEC



Fonte: Autoria Própria (2026)

Figura 14 - Avaliação do SO com base nas métricas ISO/IEC (total)



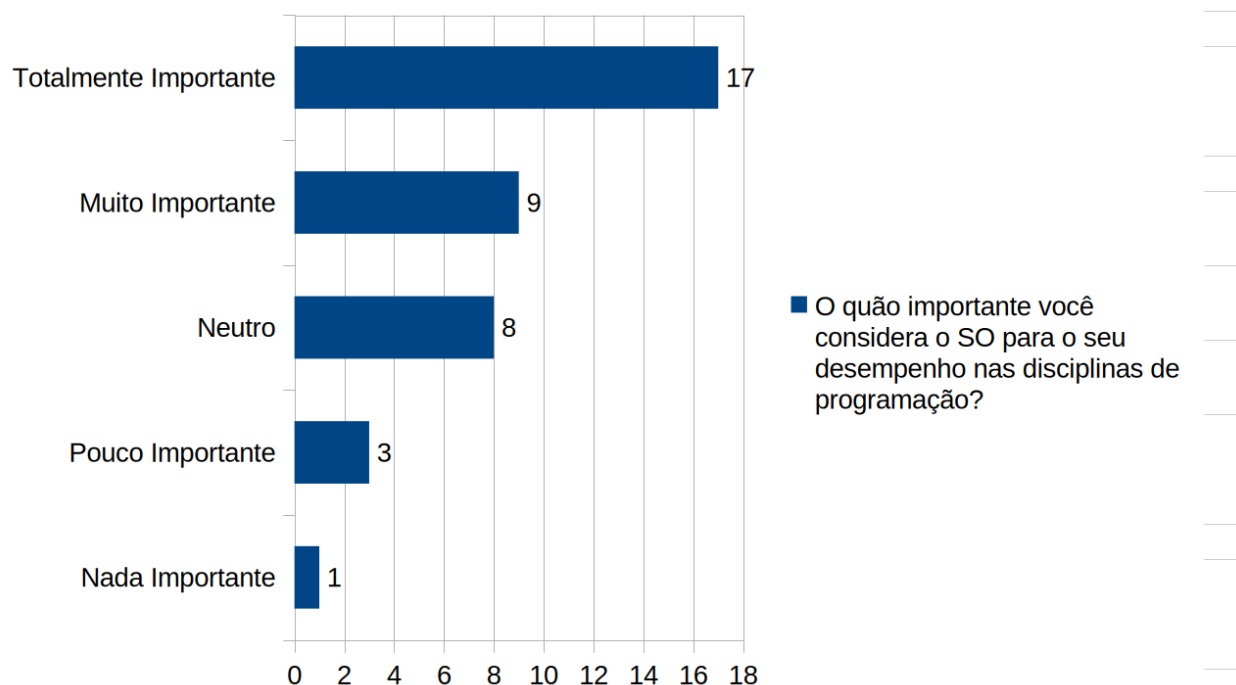
Fonte: Autoria Própria (2026)

Pode-se observar que grande parte dos discentes considera as características do seu SO como Bom (39,07%), seguido por Regular (30,47%), Excelente (22,37%), Ruim (7,09%) e Pessimo (1,01%). Ao compilar esses resultados em dois grandes grupos, tem-se: Excelente e Bom com 61,44%, enquanto Ruim e Pessimo ficaram com 8,10%.

4.6 Avaliação dos SOs em Relação à Programação

Na Figura 15, encontram-se compiladas as perguntas anteriormente em escala 1-10 (um a dez), transformadas em escala Likert 1-5 (um a cinco). O questionário original ofertou opções de um a dez, que foram convertidas considerando a cada duas opções, uma opção na escala Likert de 1-5. Por exemplo, os valores 1-2 da escala de 1-10 representam o valor 1 (discordo totalmente) na escala de Likert 1-5. Esse procedimento foi adotado para melhor visualizar os dados. Nos gráficos seguintes, esse mesmo procedimento foi adotado.

Figura 15 - Importância do SO no desempenho nos componentes curriculares de programação



Fonte: Autoria Própria (2026)

No gráfico, pode-se perceber que a maioria dos discentes considera o SO como um fator muito importante para o desempenho nos componentes curriculares relacionados à programação. Os dados em porcentagem são: Totalmente Importante 44,74% (17), Muito Importante 23,68% (9), Neutro 21,05% (8), Pouco Importante 7,89% (3), Nada Importante 2,63% (1).

Para a sessão de comentários, 24 (vinte e quatro) discentes explicaram suas escolhas, e 23 (vinte e três) seguiram para análise, conforme mencionado no início deste capítulo.

Com base nas respostas dos discentes, revela-se uma preferência pelo sistemas Linux, com o Windows ainda sendo o mais utilizado, porém com ressalvas significativas.

A maioria dos discentes demonstra um forte interesse em ambientes Linux, frequentemente associado a um aprendizado mais profundo e eficiente em lógica de programação. Essa preferência justifica-se pela percepção de que o Linux estimula a resolução de problemas e o aprofundamento em conceitos, com o terminal e a facilidade de instalação de pacotes sendo particularmente valorizados.

Os discentes enfatizam a importância de sistemas leves, enxutos e versáteis, apontando também sistemas baseados em UNIX, como melhores para programação,, em específico para a linguagem C, enquanto Linux Mint e Pop OS são citados por sua facilidade de uso.

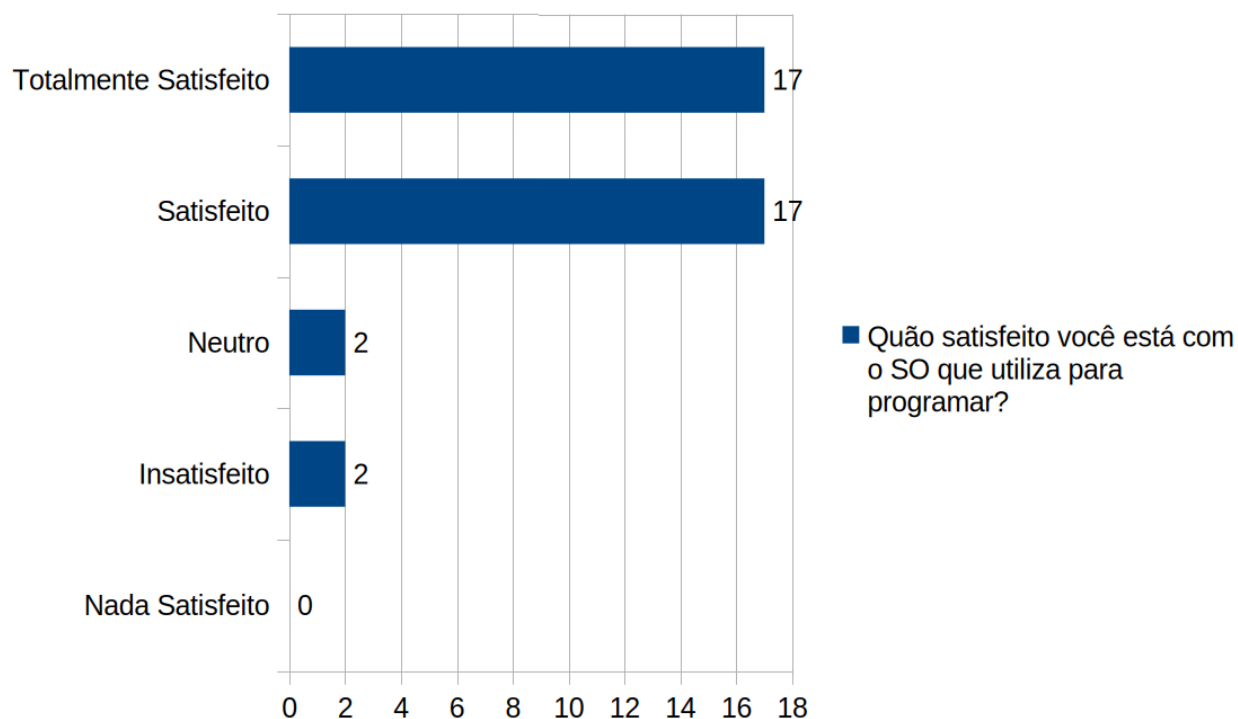
Contudo, o Windows permanece como a escolha principal para muitos, impulsionada pela familiaridade, facilidade de uso e disponibilidade de ferramentas de desenvolvimento.

Apesar das críticas sobre limitações e menor flexibilidade, a percepção é de que o Windows é mais intuitivo e prático, especialmente para discentes que não possuem experiência prévia com Linux. A importância atribuída ao SO para o desempenho nos componentes curriculares varia, com alguns discentes julgando-o como crucial para entender a lógica por trás da leitura de códigos, enquanto outros minimizam ou consideram indiferente sua influência nesse contexto.

Em suma, a escolha do SO é fortemente influenciada pela experiência e pelo conhecimento prévio do discente. A familiaridade com um sistema, seja ele Linux ou Windows, desempenha um papel fundamental, assim como a percepção da adequação do sistema para as necessidades específicas do discente e do curso.

Referente à satisfação para programar em relação ao SO utilizado, a distribuição das respostas dos discentes é apresentada na Figura 16.

Figura 16 - Satisfação para programar com o SO utilizado

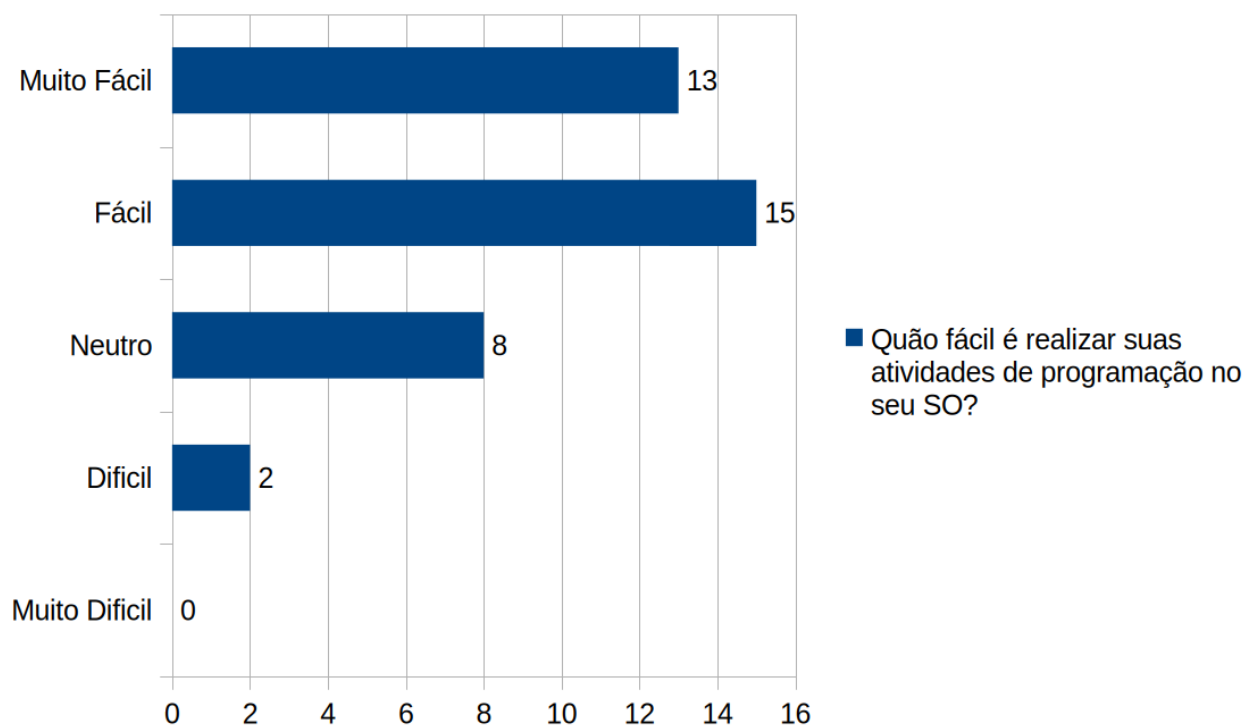


Fonte: Autoria Própria (2026)

A maioria dos discentes se considera satisfeita ou totalmente satisfeita com o SO que utilizam para programar. Os dados obtidos foram: Totalmente Satisfeito 44,74% (17), Satisfeito 44,74% (17), Neutro 5,26% (2), Insatisfeito 5,26% (2), Nada Satisfeito 0% (0).

Por sua vez, no gráfico apresentado da Figura 17, pode ser vista a distribuição das respostas para a facilidade de programar ofertada pelo SO.

Figura 17 - Facilidade para programar com o SO utilizado

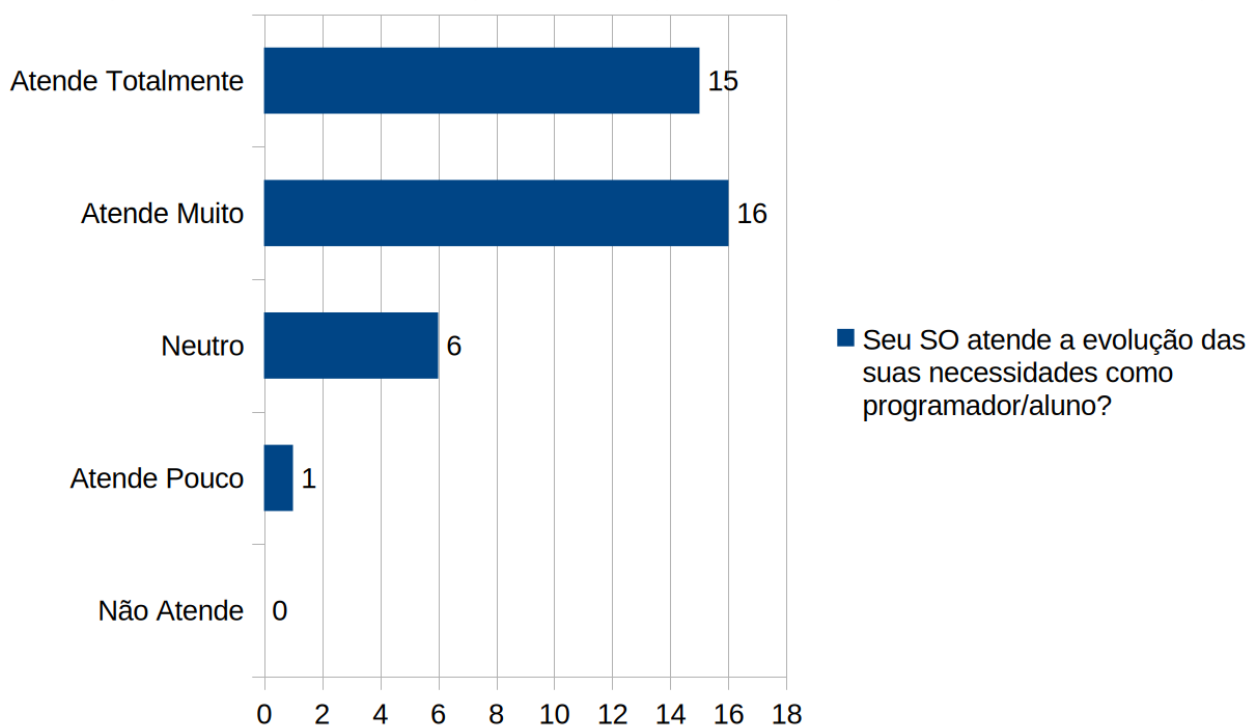


Fonte: Autoria Própria (2026)

A maioria dos discentes considera fácil, seguido por muito fácil ou totalmente satisfeita com o SO que utilizam para programar. Os dados obtidos foram: Muito Fácil 34,21% (15), Fácil 39,47% (13), Neutro 21,05% (8), Difícil 5,26% (2) e Muito Difícil 0% (0).

Por sua vez, a Figura 18 apresenta a relação do quanto o SO operacional utilizado pelo discente atende à evolução de suas necessidades.

Figura 18 - Atendimento do SO utilizado à evolução das necessidades dos discentes.



Fonte: Autoria Própria (2026)

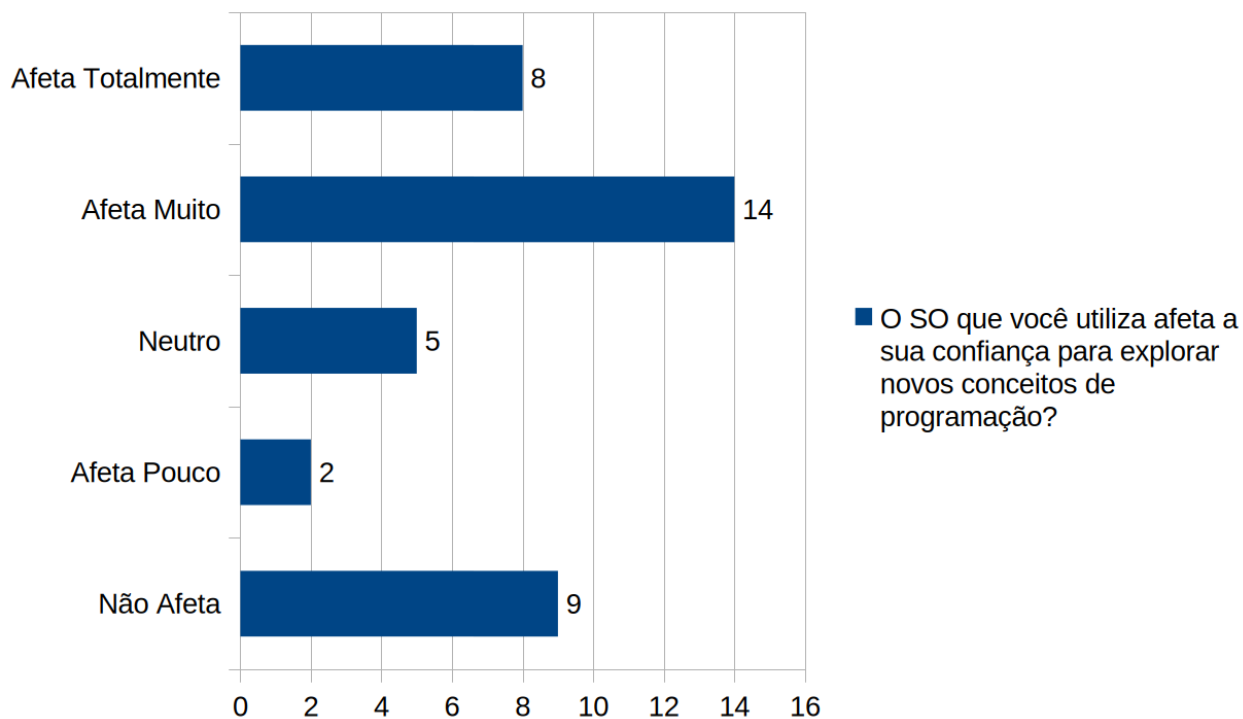
No geral, a maioria dos discentes considera que seu SO de escolha atende muito bem suas necessidades de programação. Os dados obtidos foram: Atende Totalmente 39,47% (15), Atende Muito 42,11% (16), Neutro 15,79% (6), Atende Pouco 2,63% (2), Não Atende 0% (0).

Entre os três comentários obtidos, apenas dois foram relevantes para a análise, de acordo como mencionado no início deste capítulo.

Os discentes informaram que seu SO supre muito bem as suas necessidades e que isso também é uma fonte de motivação. Além disso, a compatibilidade de programas do Windows e a facilidade de uso de terminais Linux são fatores apontados pelos discentes.

Na sequência, a Figura 19 apresenta a confiança dos discentes no SO para explorar novos conceitos de programação.

Figura 19 - Confiança no SO para explorar novos conceitos de programação

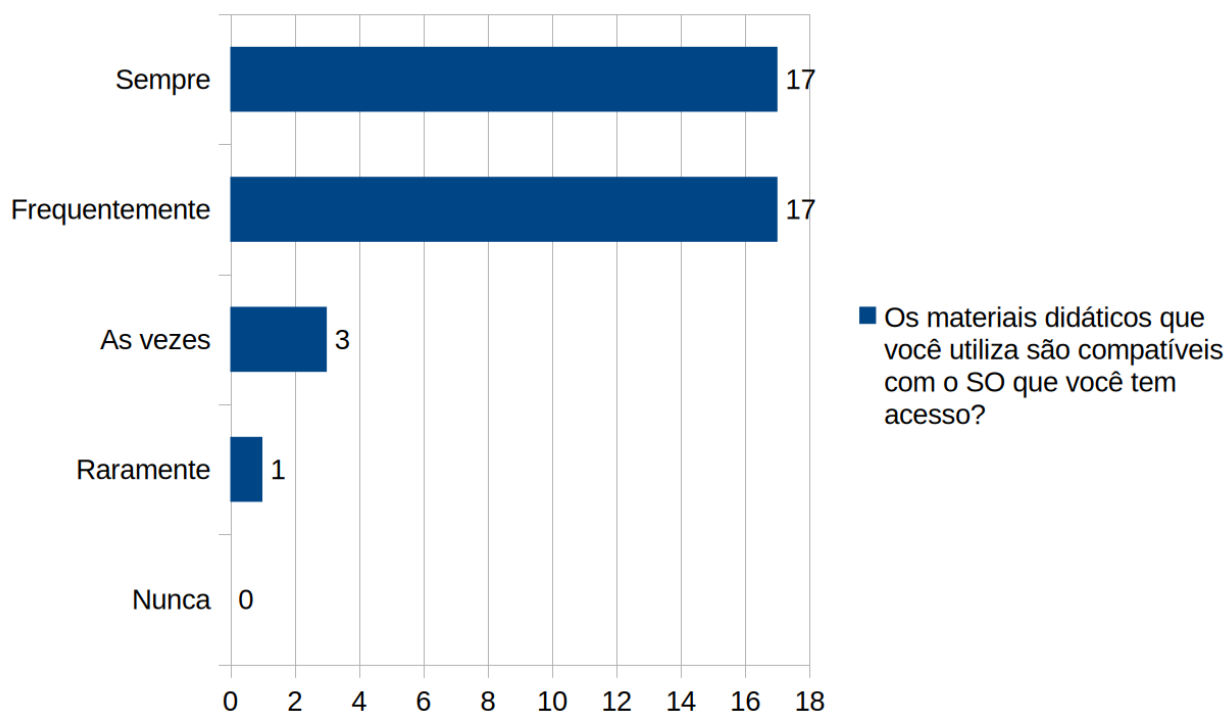


Fonte: Autoria Própria (2026)

No geral, a maioria dos discentes considera que seu SO de escolha afeta muito sua confiança ao explorar novos conceitos de programação. Os dados obtidos foram: Afeta Totalmente 21,05% (8), Afeta Muito 36,84% (14), Neutro 13,16% (5), Afeta Pouco 5,26% (2), Não Afeta 23,68% (9).

Por sua vez, o gráfico da Figura 20 busca compreender se os materiais didáticos utilizados são compatíveis com o SO utilizado pelos discentes.

Figura 20 - Os materiais didáticos utilizados são compatíveis com o SO



Fonte: Autoria Própria (2026)

No geral, a maioria dos discentes considera que os materiais didáticos utilizados são compatíveis com seu SO. Os dados obtidos foram: Sempre 44,74% (17), Frequentemente 44,74% (17), Às vezes 7,89% (3), Raramente 2,63% (1), Nunca 0% (0).

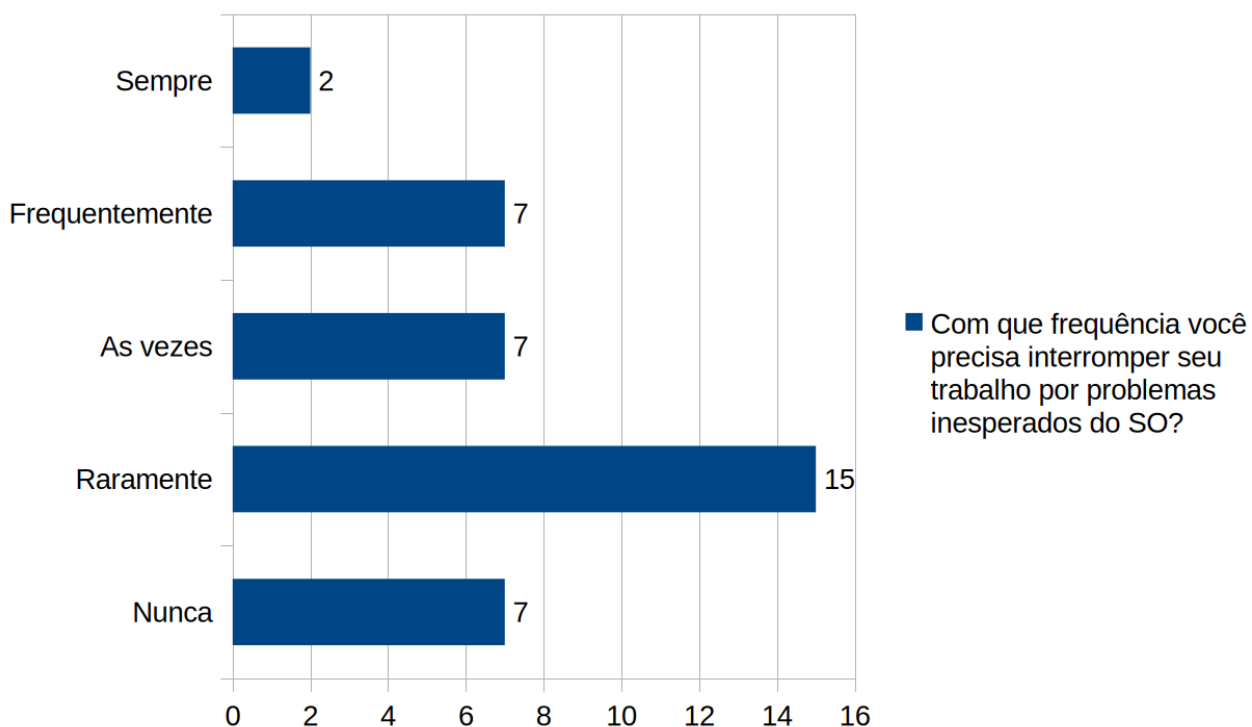
Dentre os quatro comentários obtidos, apenas um foi relevante para a pesquisa, conforme mencionado anteriormente.

Houve um relato a respeito da dificuldade de instalação da biblioteca `allegro5`⁷ para o sistema Windows, biblioteca utilizada para desenvolvimento de jogos em projetos de componentes curriculares, tais como Algoritmos e Estruturas de Dados I.

Já o gráfico da Figura 21 apresenta a distribuição das respostas referente à frequência de ocorrências de problemas inesperados do SO utilizado pelo discente.

⁷ <https://github.com/liballeg/allegro5>

Figura 21 - Frequência de problemas inesperados do SO

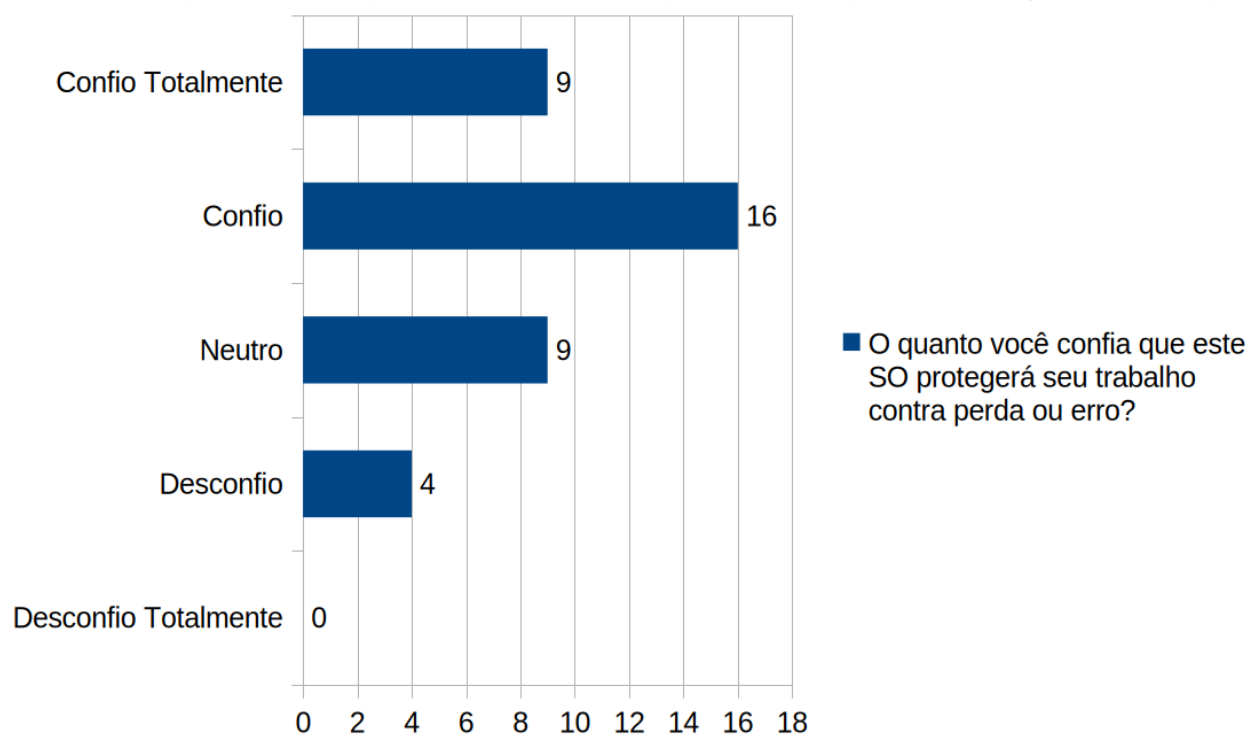


Fonte: Autoria Própria (2026)

No geral, a maioria dos discentes considera que os materiais didáticos utilizados são compatíveis com seu SO. Os dados obtidos foram: Sempre 5,26% (2), Frequentemente 18,42% (7), Às vezes 18,42% (7), Raramente 39,47% (15), Nunca 18,42% (7).

No tocante à confiança do discente de que o SO utilizado protege os seus trabalhos contra perda ou erros, os dados são apresentados na Figura 22.

Figura 22 - Confiança no SO em relação à proteção de trabalho contra perda ou erro



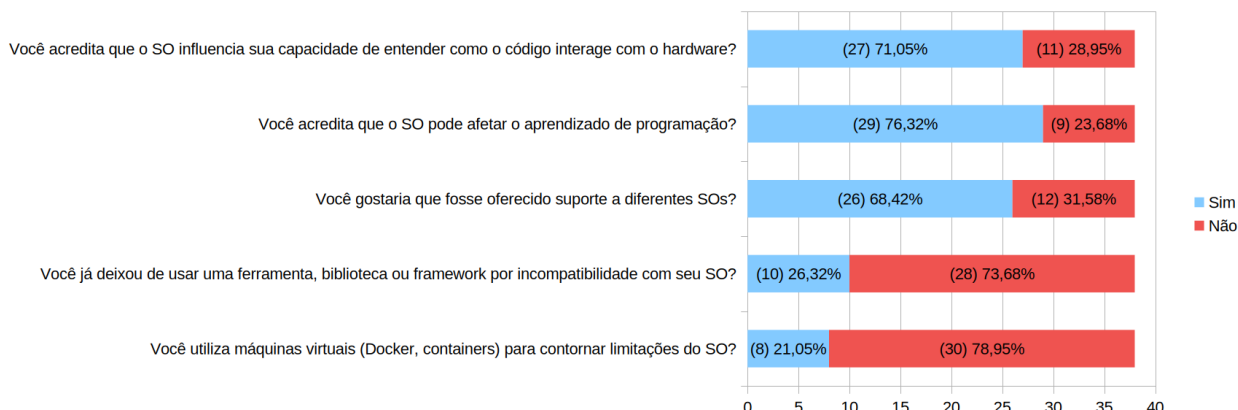
Fonte: Autoria Própria (2026)

No geral, a maioria dos discentes considera que seu SO irá proteger seu trabalho caso haja alguma perda ou erro no sistema. Os dados obtidos foram: Confio Totalmente 23,68% (9), Confio 42,11% (16), Neutro 23,68% (9), Desconfio 10,53% (4), Desconfio Totalmente 0% (0).

4.7 Perguntas Objetivas

Na Figura 23 são apresentadas respostas para perguntas com respostas binárias (sim e não), conta-se com o suporte de seções de comentários para que o participante esclareça sua escolha. O gráfico conta com a quantidade e da porcentagem dessas respostas, além disso, ao analisar cada questão, há uma subdivisão dessas porcentagens obtidas em relação ao principal SO utilizado pelo discente, com o intuito de esclarecer a demografia dos SOs nas respostas. No gráfico a seguir, é possível ver os resultados obtidos para cada uma delas:

Figura 23 - Resultado das perguntas objetivas



Fonte: Autoria Própria (2026)

4.7.1 Análise da Primeira Questão

Para a primeira pergunta, a maioria dos discentes acredita que o SO influencia na capacidade de entender como o código interage com o *hardware*, com uma taxa de 71,05%, enquanto 28,95% dizem que não. Dentre as respostas, tem-se que: entre os que afirmaram que influencia, 70,37% usam Windows e 29,63% utilizam distribuições Linux. Para os que declaram que não influencia, o Windows situa-se em 81,82% e as distribuições Linux encontram-se com 18,18%.

Para a sessão de comentários, 15 (quinze) discentes explicaram suas escolhas, e 14 (quatorze) delas seguiram para análise, conforme mencionado no início deste capítulo. A maioria das respostas indica que o SO exerce uma influência significativa na capacidade de um indivíduo compreender a interação entre código e *hardware*. Essa influência, nas palavras dos discentes, reside principalmente no papel do SO como um intermediário fundamental entre o *software* e os componentes físicos do computador.

Diversas respostas destacam que o SO facilita a compreensão ao fornecer acesso a recursos internos e à comunicação entre o *software* e o *hardware*. Sistemas baseados em Linux são frequentemente apontados como exemplos de SOs que promovem essa compreensão, devido à sua natureza mais transparente e à exposição de informações sobre o funcionamento do sistema. A capacidade de interagir diretamente com a máquina via comandos, como no Linux, é vista como uma abordagem mais intuitiva e informativa em comparação com a interação mais "automatizada" oferecida por sistemas como o Windows.

No entanto, uma parcela das respostas argumenta que a influência do SO na compreensão dessa interação é menos direta. Algumas pessoas acreditam que esse conhecimento é algo que deve ser buscado de forma agonística ao SO, enquanto outras defendem que essa comunicação é fundamentalmente a mesma, independentemente da plataforma. Sistemas de “baixo nível”, como o Arch Linux ou Kali Linux, exigem um conhecimento mais profundo sobre o *hardware* e o *kernel*, enquanto SOs mais focados no usuário, como o Windows, abstraem essas complexidades.

Em suma, a maioria das respostas converge para a ideia de que o SO desempenha um papel crucial na compreensão da interação entre código e *hardware*, com sistemas baseados em Linux sendo frequentemente citados como facilitadores dessa compreensão devido à sua transparência e acesso direto ao *hardware*.

4.7.2 Análise da Segunda Questão

Na segunda questão, a maioria dos discentes acredita que o SO utilizado pode afetar o aprendizado de programação com uma taxa de 76,32%, enquanto 22,68% acreditam que não. Ainda nessa questão, tem-se que: entre os que afirmaram que afeta, 65,52% usam Windows e 34,48% utilizam distribuições Linux. Para os que declaram que não afeta, o Windows encontra-se com uma taxa de 100% e as distribuições Linux não obtiveram respostas.

Entre as 38 (trinta e oito) respostas obtidas, somente 35 (trinta e cinco) seguiram relevantes ao intuito da pesquisa, conforme mencionado no início deste capítulo. A maioria das respostas obtidas pelos discentes sobre a pergunta converge para o seguinte ponto: o SO influencia significativamente esse processo. Fatores como o ambiente de trabalho, as ferramentas disponíveis e a forma como o discente interage com o código são moldados pelo SO escolhido.

Essa influência se manifesta de diversas maneiras, com sistemas baseados em Linux, por exemplo, sendo frequentemente citados como sistemas que incentivam uma compreensão mais profunda dos conceitos de programação, devido ao seu terminal e foco em *scripts*. Em contrapartida, a abstração do Windows, que esconde muitos dos processos de fundo em interfaces gráficas, pode dificultar o controle e a solução de problemas.

A dificuldade de configuração, a falta de compatibilidade com ferramentas de desenvolvimento e a necessidade de aprender a utilizar o terminal são apontados como fatores que podem gerar desmotivação e, conseqüentemente, dificultar o aprendizado. Por outro lado, a

interação com o SO, como aprender a manipular arquivos no terminal ou compilar e executar programas pode acelerar o aprendizado e ampliar o conhecimento do discente. A filosofia de trabalho de cada SO também desempenha um papel importante, com o Linux incentivando a exploração e o uso do terminal, enquanto o Windows oferece uma interface mais intuitiva.

Embora uma menor parcela das respostas argumente que o SO não afeta os conceitos básicos da programação, mas sim o ambiente e a experiência do discente, a maioria dos respondentes reconhece a importância da escolha do SO como um fator que pode influenciar o aprendizado de forma positiva ou negativa.

4.7.3 Análise da Terceira Questão

Para a terceira pergunta, a maioria dos discentes afirma que gostaria que fossem disponibilizados diferentes SOs para os computadores da universidade, com uma taxa de 68,42%, enquanto 31,58% dizem que não querem. Esses resultados podem ser divididos em dois grupos:

Entre os que afirmaram que querem suporte, 30,77% usam Windows e 69,23% utilizam distribuições Linux. Para os que declaram que não querem suporte, o Windows situa-se em 83,33% e as distribuições Linux encontram-se com 16,67%

Para a sessão de comentários, 15 (quinze) discentes explicaram a razão de suas escolhas, porém somente 14 (quatorze) seguiram para revisão, conforme mencionado no início deste capítulo.

A principal razão apontada é a necessidade de praticidade e adaptabilidade. Muitos discentes argumentam que a familiaridade com um SO específico é um fator crucial para o aprendizado e o suporte a diferentes plataformas permitiria que cada um escolhesse o ambiente com o qual se sente mais confortável e produtivo. Além disso, a diversidade de SOs oferece a oportunidade de compreender melhor o comportamento do *software* em diferentes contextos, o que é considerado valioso para o desenvolvimento e para o conhecimento da tecnologia em geral.

Uma corrente de respostas destaca o interesse em Linux, com menções específicas a distribuições como Linux Mint, Pop OS e Ubuntu, refletindo a popularidade dessa plataforma entre os discentes. A busca por mais competição no mercado também foi mencionada, com a observação de que a Microsoft está em processo de otimização do Windows 11, influenciada pelo SteamOS.

Outro ponto central é a questão da acessibilidade e inclusão. Alguns discentes acreditam que o suporte a diferentes SOs, incluindo macOS e o já existente Windows, tornaria o aprendizado de programação mais acessível e inclusivo, evitando limitar os usuários a um único ambiente. A capacidade de colaborar entre equipes que utilizam sistemas diferentes também é vista como um benefício.

Em resumo, a demanda por suporte a diferentes SOs reflete um desejo de maior flexibilidade, praticidade, conhecimento e inclusão no aprendizado de programação, com foco especial em Linux, mas também considerando outras plataformas relevantes para o mercado tecnológico.

4.7.4 Análise da Quarta Questão

Na quarta questão, a maioria dos discentes afirma que nunca deixou de utilizar ferramentas por causa do seu SO, com uma taxa de 73,68%, já 26,32% afirmaram que sim. Com base no SO declarado como principal, esses resultados podem ser divididos em dois grupos:

Entre os que afirmaram que deixaram, 87,5% usam Windows e 12,5% utilizam distribuições Linux. Para os que declaram que não deixaram, o Windows situa-se em 50% e as distribuições Linux encontram-se com 50%

Das seis respostas obtidas, cinco foram relevantes para a pesquisa, de acordo com o mencionado no início deste capítulo.

Para os discentes que responderam que sim, não há qualquer problema relatado a respeito de problemas de incompatibilidade de *softwares* de programação. No entanto, dentre os programas destacados, tem-se o pacote Office da Microsoft e o Anki (aplicativo de *flashcards*) marcados por tal inconsistência com os sistemas dos discentes. Mesmo assim, a forma com que solucionaram o problema foi encontrar alternativas de *software* semelhantes ou deixar de utilizar a ferramenta por completo. Para a primeira solução, o participante afirmou que conseguiu mudar para uma alternativa sem muita dificuldade. Para a última, a falta daquele *software* não teve impactos significativos expressivos.

4.7.5 Análise da Quinta Questão

Para a última pergunta, tem-se que a maioria dos discentes não utiliza Docker⁸ ou nenhuma forma de virtualização para contornar limitações do seu SO, com uma taxa de 78,95%, enquanto 21,05% dizem que utilizam. Com base no SO declarado como principal, esses resultados podem ser divididos em dois grupos: entre os que afirmaram que utilizam, 87,5% usam Windows e 12,5% utilizam distribuições Linux. Para os que declaram que não utilizam, o Windows situa-se em 70% e as distribuições Linux encontram-se com 30%.

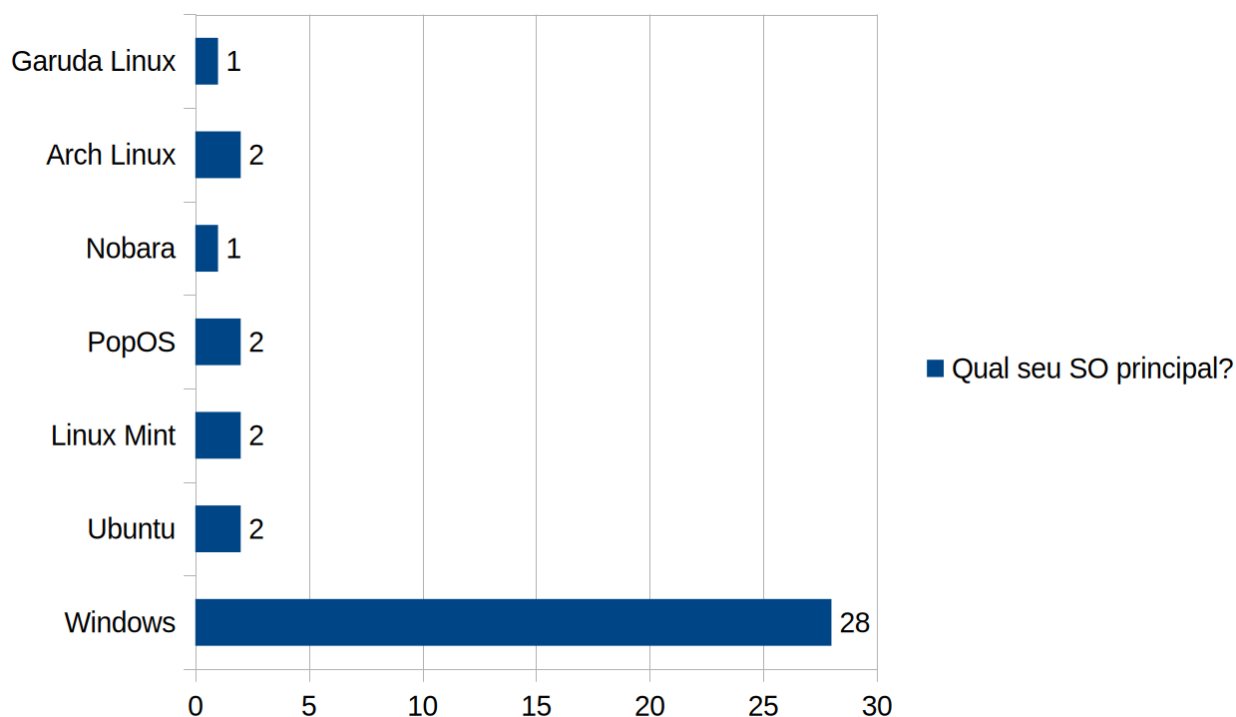
Entre os seis comentários obtidos, apenas cinco foram relevantes para análise, conforme mencionado no início deste capítulo. Os quatro que informaram que sim discutem que a utilização dessas ferramentas contribui para o seu aprendizado e habilidade geral com a programação, além disso destaca-se a facilidade de testar diferentes versões de linguagens e ferramentas sem haver conflito com o SO principal. Houve ainda uma resposta dizendo que ainda não havia utilizado essa alternativa, porém que iria testá-la.

4.8 Perguntas sobre Sistemas Operacionais

Na Figura 24 é apresentada a distribuição de respostas para o SO principal utilizado pelos discentes.

⁸ <https://www.docker.com/>

Figura 24 - Principal SO utilizado⁹



Fonte: Autoria Própria (2026)

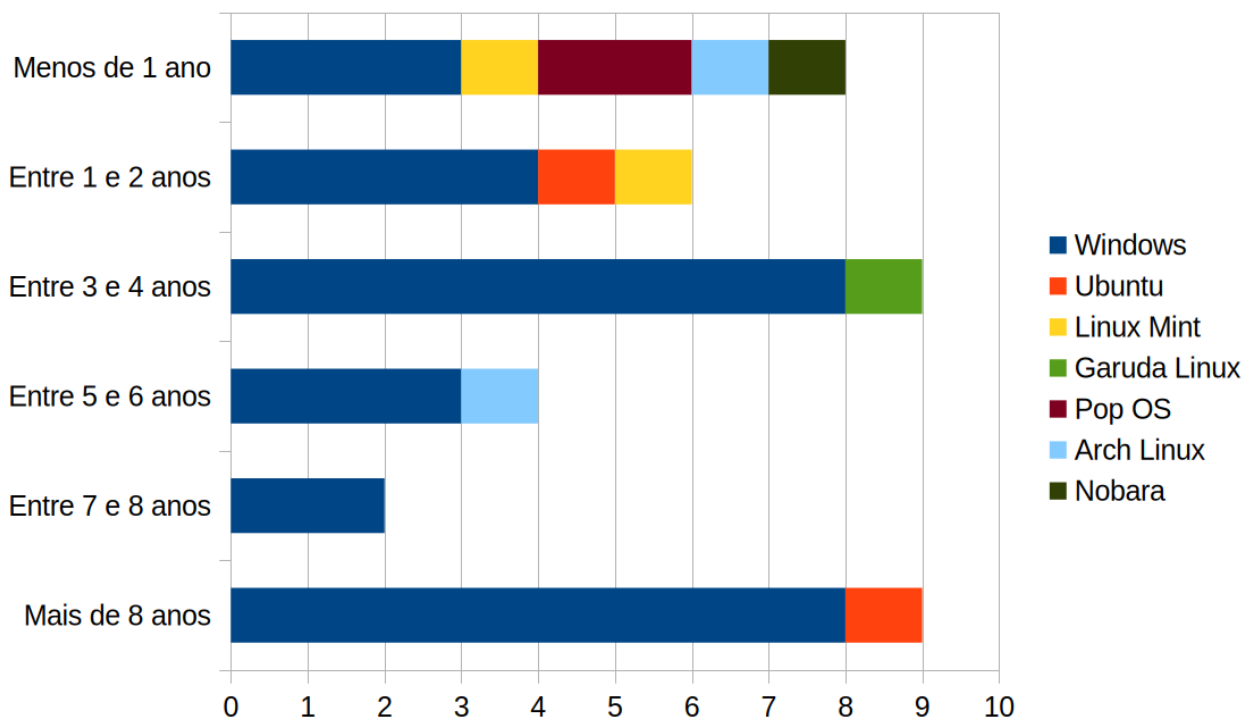
Pode-se verificar que a maioria dos discentes utiliza Windows com 73,68% (28), seguido por Ubuntu (2), Linux Mint (2), Pop OS (2), Arch Linux (2), todos os quatro correspondente a 5,26%, já Nobara (1) e Garuda Linux (1) somam 2,63%. Ao agrupar esses dados, com base no principal SO utilizado, tem-se que: Windows 73,68% e distribuições Linux 26,32%.

Resultados esses, extremamente parecidos quando comparados às porcentagens de SOs para os componentes curriculares analisados (excluindo macOS e Android), têm-se uma taxa de 78,83% para o Windows e 21,17% para distribuições Linux, a diferença tanto para Windows como para distribuições Linux é de 5,15%.

Por sua vez, a Figura 26 busca trazer uma análise sobre o tempo de uso de diferentes SOs pelos discentes nos componentes curriculares.

⁹ Distribuições Linux: Ubuntu, Linux Mint, Pop OS, Nobara, Arch Linux e Garuda Linux.

Figura 26 - Tempo de uso do principal ambiente de desenvolvimento para os componentes curriculares analisados em relação ao SO marcado como principal



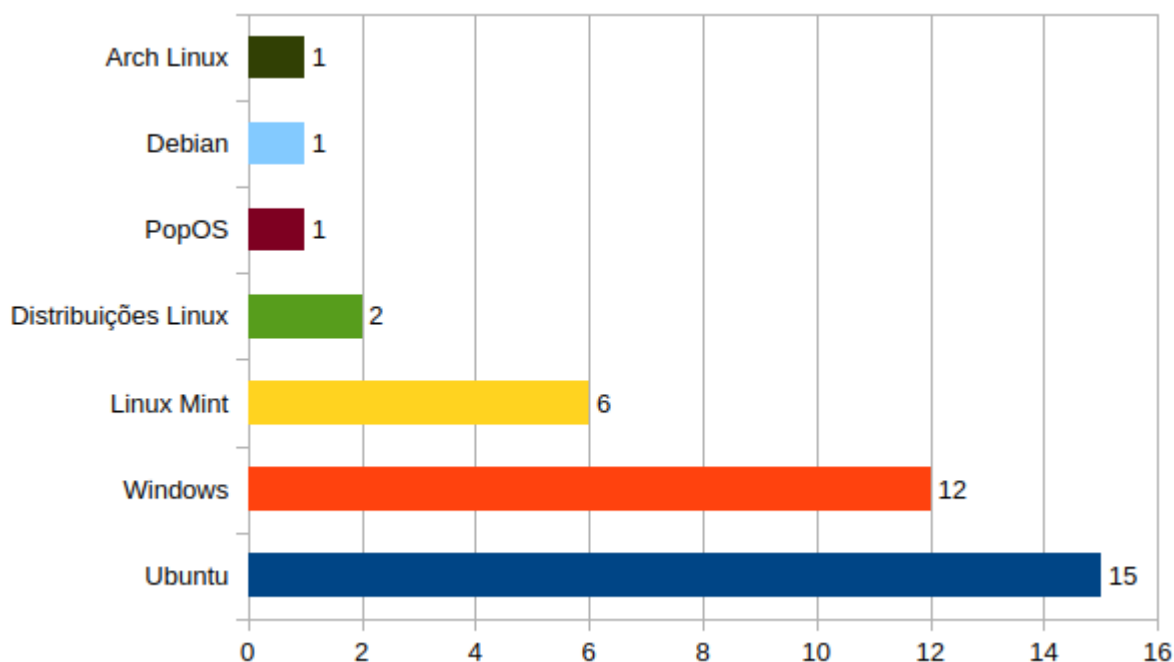
Fonte: Autoria Própria (2026)

Com base nas informações coletadas, percebe-se que os discentes que informaram utilizar um SO específico enquanto estavam cursando os primeiros componentes curriculares de programação continuaram com suas escolhas de SO até aquelas mais avançadas do curso. Ainda pode-se observar outra possível tendência: há uma maior adoção de sistemas baseados em Linux logo no início do curso, porém, conforme o passar do tempo, os discentes tendem a utilizar mais o Windows.

Entre os dez comentários obtidos nessa questão, nove seguiram para análise, como mencionado no início deste capítulo. Há uma variedade de experiências com SOs para desenvolvimento. O Windows é o SO preferido pela maioria, com usuários utilizando-o por um longo período, muitas vezes por sua praticidade no dia a dia. Já o uso do Linux ganha espaço, impulsionado pelo desejo de aprender novas linguagens de programação ou buscar um ambiente mais flexível e personalizável.

Quando perguntado sobre qual SO seria o mais adequado para o aprendizado de programação, chegou-se ao gráfico apresentado na Figura 27.

Figura 27 - SO mais adequado para o aprendizado de programação¹⁰



Fonte: Autoria Própria (2026)

A maioria dos discentes apontou para o Ubuntu com 39,47% (15) como o melhor SO para aprender a programar, seguido por Windows com 31,58% (12), Linux Mint com 15,79% (6), distribuições Linux no geral com 5,26% (2), Arch Linux (1), Pop OS (1) e Debian (1), estes com 2,63%. Apesar da grande diferença entre os sistemas, pode-se classificá-los em dois grandes grupos: Distribuições Linux com 68,42% e Windows com 31,58%.

Em seguida, a essa pergunta houve uma sessão de comentários para que os discentes explicassem suas escolhas, em que todos os 38 (trinta e oito) discentes responderam, porém após uma análise, apenas 36 (trinta e seis) delas foram levadas em consideração à pesquisa, conforme mencionado logo no início deste capítulo.

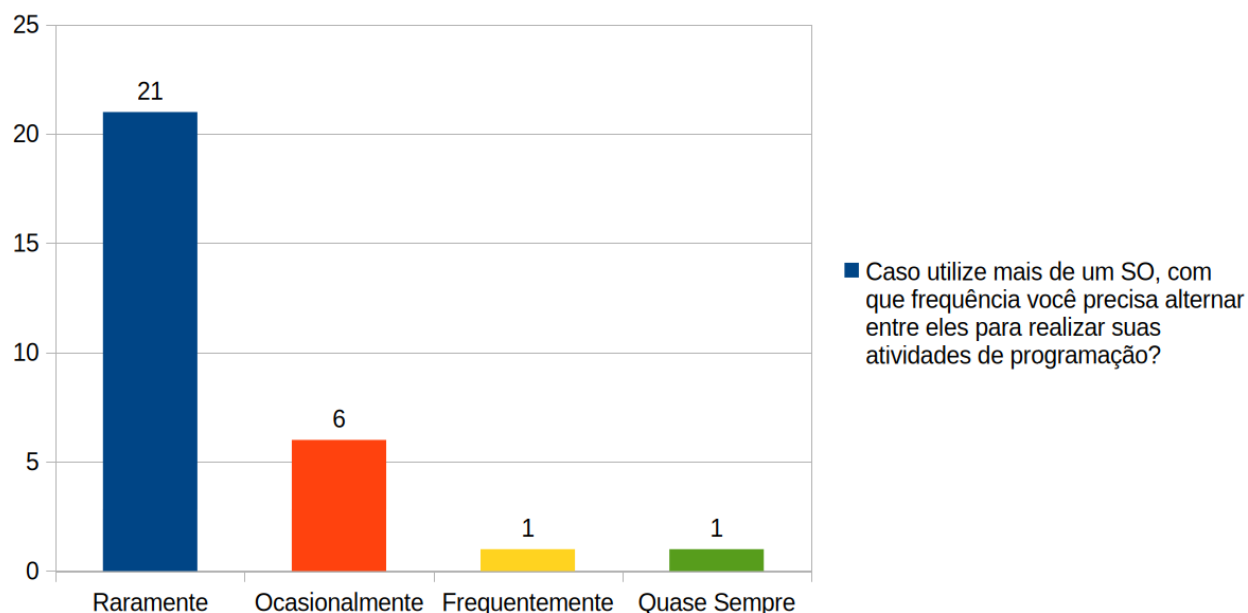
O Ubuntu surgiu como a opção mais citada e com maior respaldo. Aspectos como uma vasta documentação, uma comunidade ativa e a estabilidade do sistema foram os pontos citados como ideais para iniciantes, oferecendo um ambiente de desenvolvimento robusto e compatível com diversas ferramentas e linguagens.

¹⁰ Distribuições Linux: Ubuntu, Linux Mint, Pop OS, Nobara, Debian e Arch Linux.

O Linux Mint se destaca, especialmente para quem migra do Windows, devido à sua interface familiar, leveza e foco em otimizar o ambiente de programação. O Debian é recomendado por sua estabilidade. O Pop OS e o Arch Linux são mencionados como opções interessantes, porém mais complexas para iniciantes. Já o Windows é citado, principalmente pela familiaridade e compatibilidade com *softwares*, além de integrações com *engines* gráficas como Unreal Engine e Unity.

A próxima pergunta preocupa-se em saber a frequência da alternância entre SOs com o intuito de programar, cuja distribuição de respostas é apresentada na Figura 28.

Figura 28 - Frequência de alternância entre SOs para programar



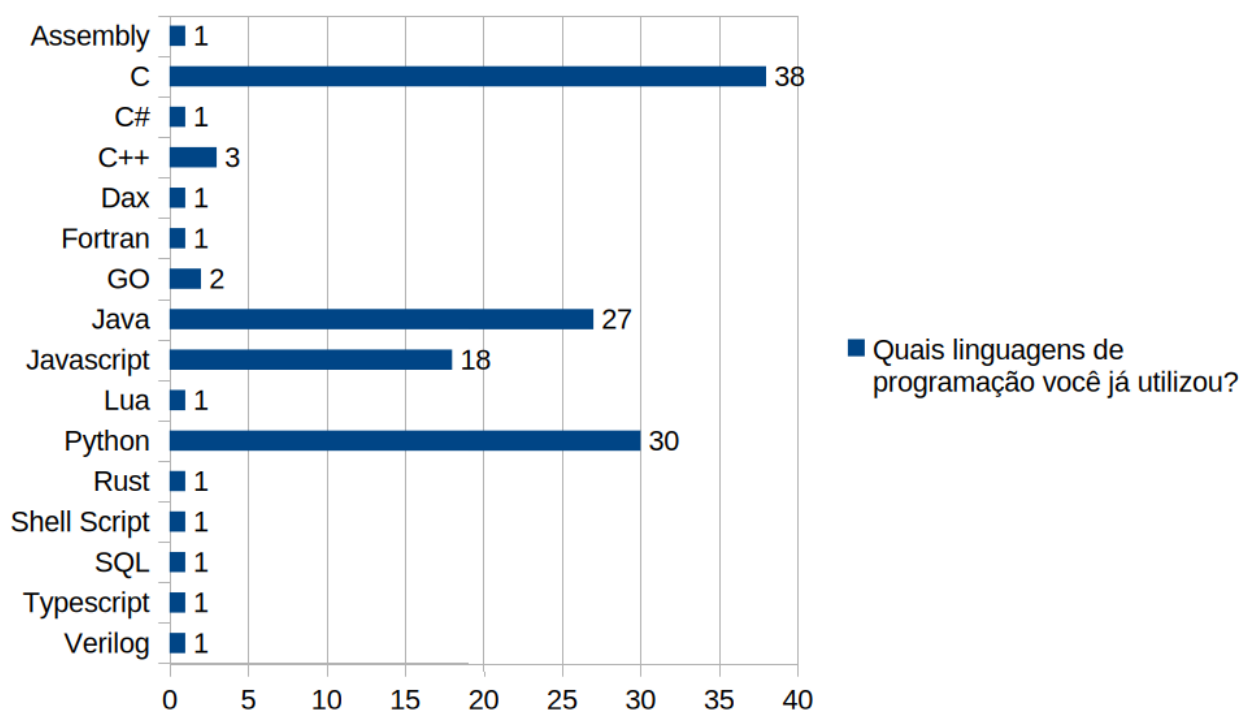
Fonte: Autoria Própria (2026)

Com base no gráfico, tem-se a seguinte relação: Raramente com 72,41% (21), Ocasionalmente com 20,69% (6), Frequentemente com 3,45% (1) e Quase Sempre com 3,45% (1). Ou seja, a maioria dos discentes não costuma alternar com frequência entre SOs apenas com o intuito de programar. Cabe ainda ressaltar que essa pergunta não foi marcada como obrigatória, já que o participante iria apenas respondê-la caso utilizasse *Dual Boot* ou *Trial Boot* ou ainda se utilizasse mais de um computador com diferentes sistemas para cada um (caso este que não está no escopo da pesquisa).

4.9 Linguagens de Programação

Em relação às linguagens de programação utilizadas, foram oferecidas as opções: C, Java, Python, JavaScript e “outra opção”, caso o participante necessitasse. Obteve-se o resultado apresentado na distribuição da Figura 29.

Figura 29 - Linguagens de programação utilizadas



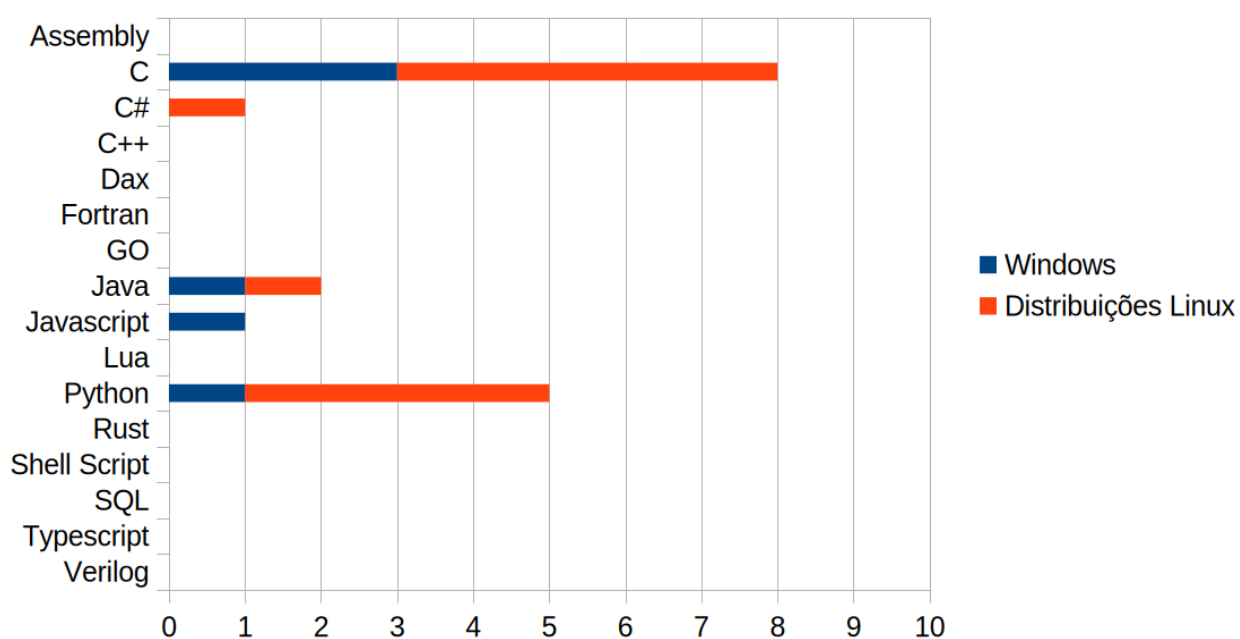
Fonte: Autoria Própria (2026)

No geral, os discentes utilizam, na maioria, as linguagens: C (38), Python (30), Java (27) e Javascript (18). Além disso, diversas outras linguagens foram citadas, como: C++ (3), Go (2), Assembly (1), Verilog (1), Lua (1), Rust (1), Shell Script (1), Typescript (1), C# (1), SQL (1), Fortran (1) e Dax (1). A alta utilização das linguagens C, Java, Python e JavaScript pode estar associado ao fato dos componentes curriculares utilizarem essas linguagens voltadas ao aprendizado de programação, já as outras menos utilizadas podem ser devido a alguma necessidade mais específica, tanto sobre o aspecto acadêmico (Assembly, C++) quanto do SO utilizado (Shell Script).

4.10 SOs e Tempo de Estudo das Linguagens de Programação

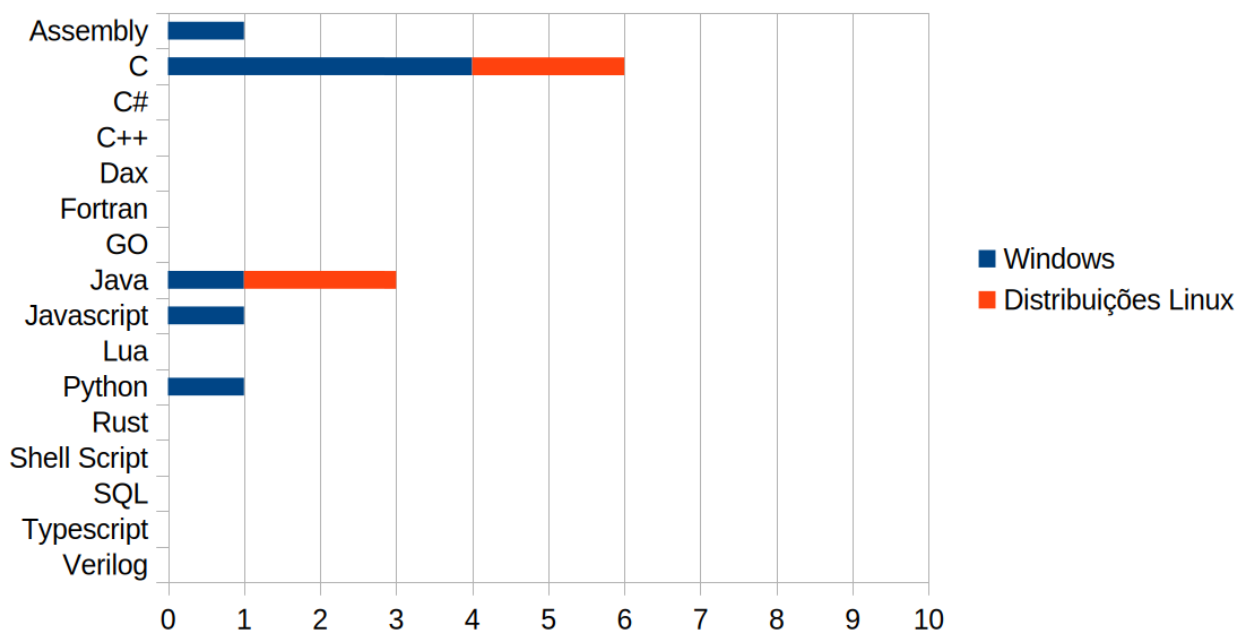
Outro ponto investigado foi uma relação entre os sistemas operacionais utilizados pelos discentes e o tempo em que os discentes estão interagindo com linguagens de programação. A distribuição das respostas pode ser visualizada nos gráficos apresentados nas Figuras 30 a 35. Por sua vez, a Figura 36 sintetiza esses dados.

Figura 30 - Relação para menos de um ano de estudo de programação



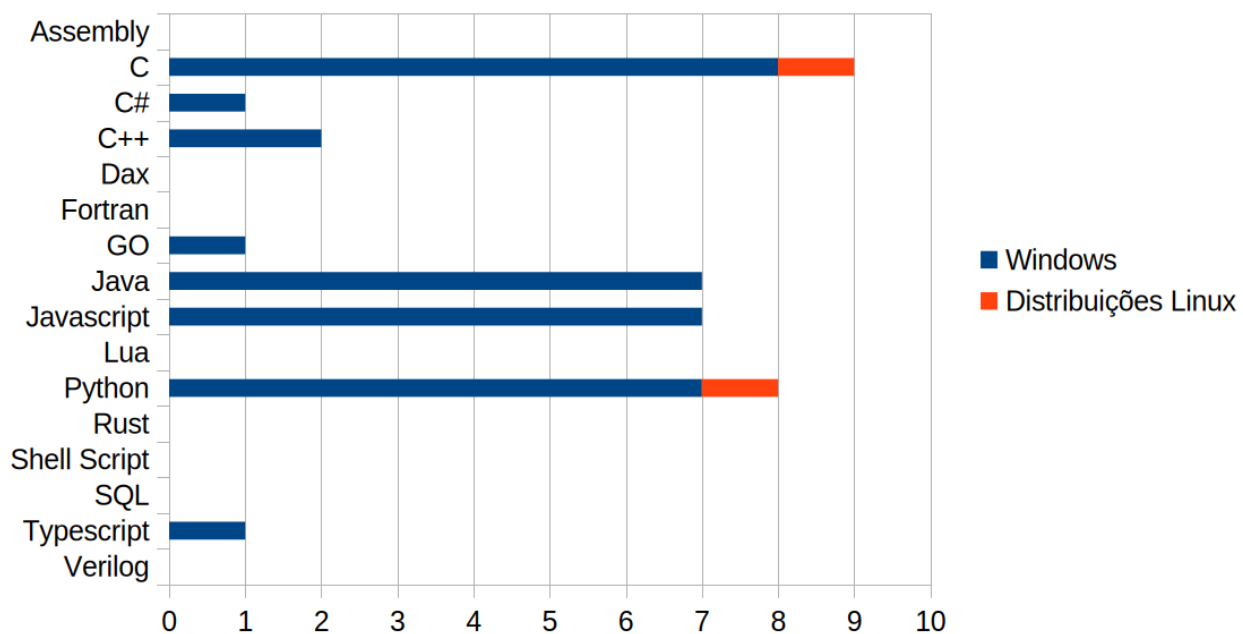
Fonte: Autoria Própria (2026)

Figura 31 - Relação entre um a dois anos de estudo de programação



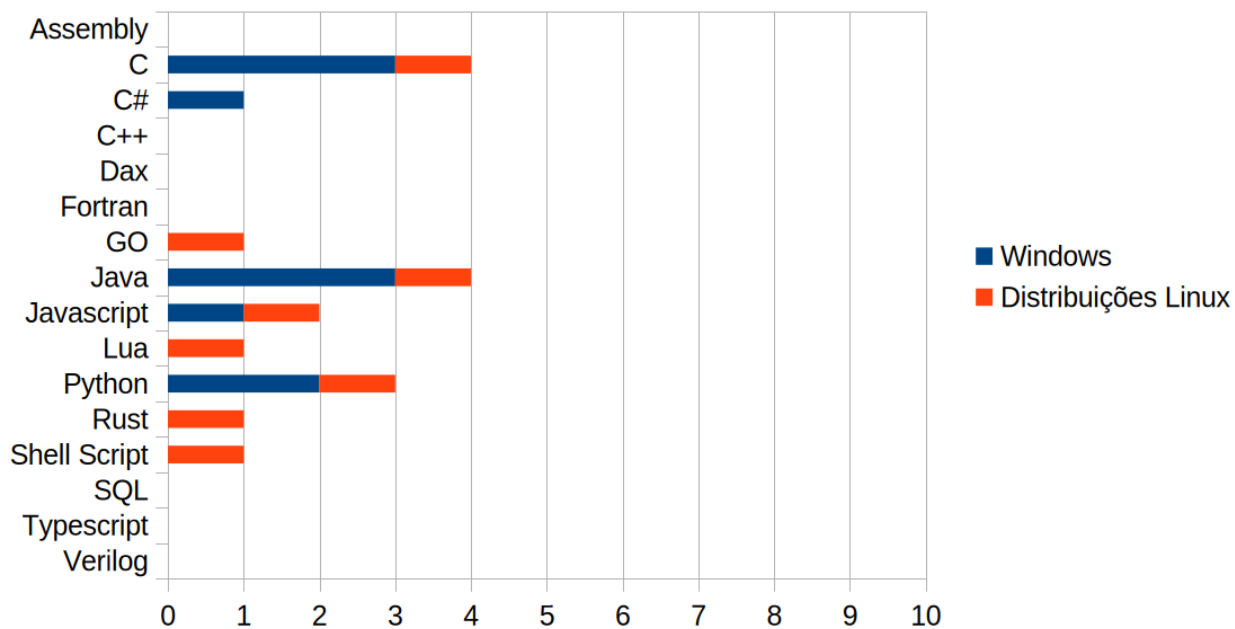
Fonte: Autoria Própria (2026)

Figura 32 - Relação entre três a quatro anos de estudo de programação



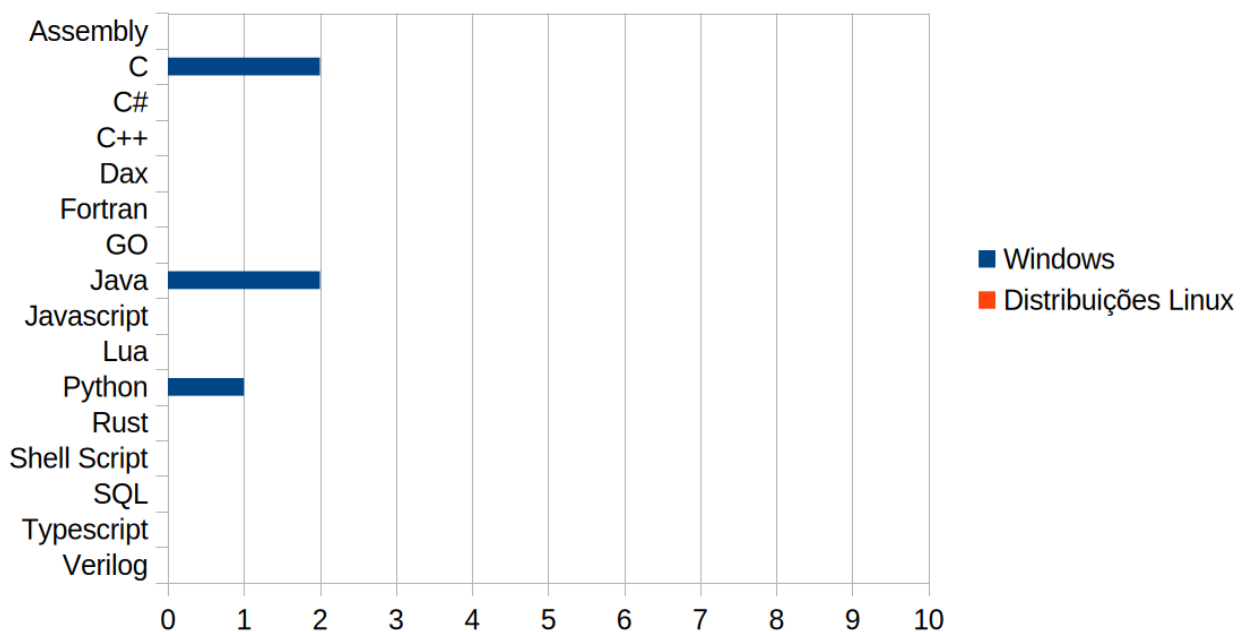
Fonte: Autoria Própria (2026)

Figura 33 - Relação entre cinco a seis anos de estudo de programação



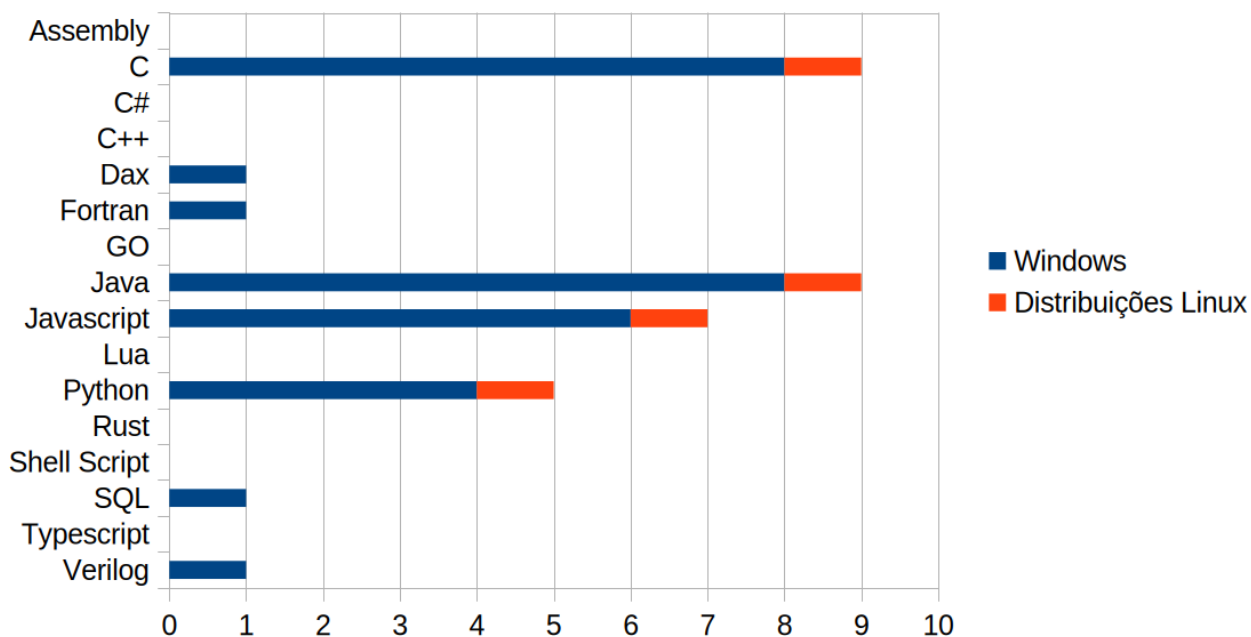
Fonte: Autoria Própria (2026)

Figura 34 - Relação entre sete a oito anos de estudo de programação



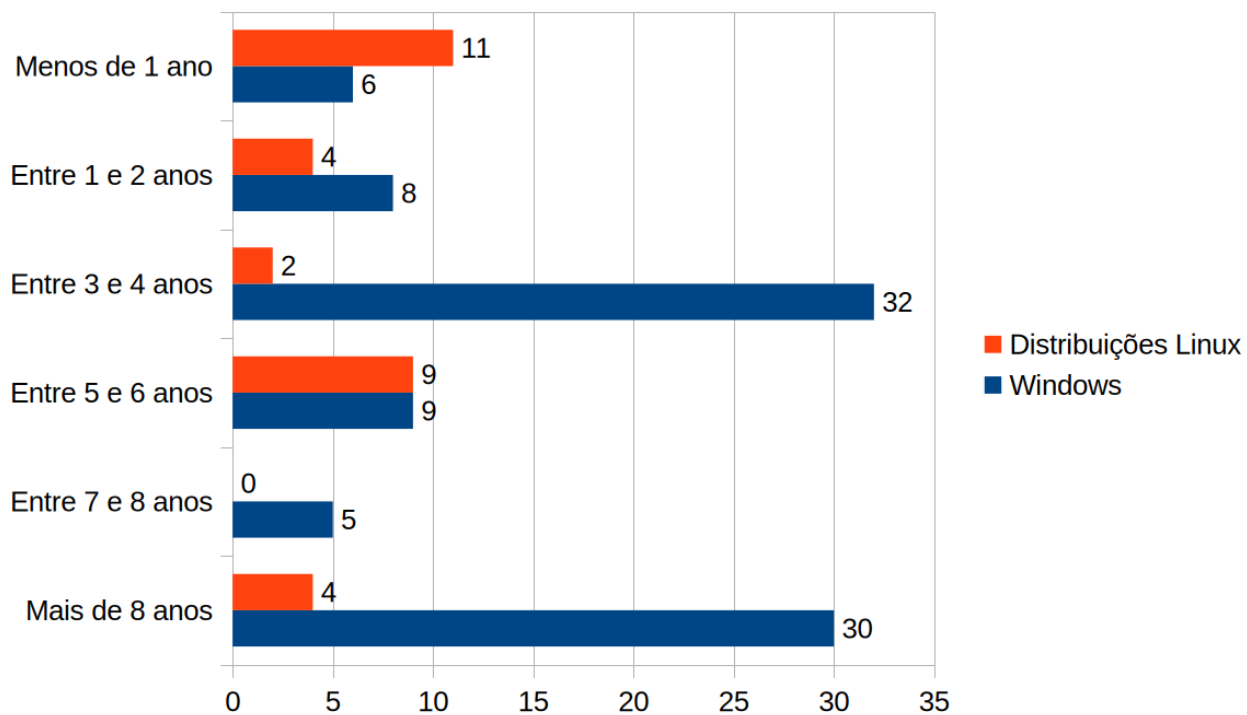
Fonte: Autoria Própria (2026)

Figura 35 - Relação para mais de oito anos de estudo de programação



Fonte: Autoria Própria (2026)

Figura 36 - Relação entre todos os intervalos de estudo de programação (Compilado das Figuras 30 a 35)



Fonte: Autoria Própria (2026)

A tendência do gráfico mostra que quanto mais tempo de estudo de programação, maior é o número de diferentes linguagens de programação utilizadas e maior é a utilização do Windows como SO principal. Em contrapartida, os discentes que utilizam alguma distribuição Linux iniciam sua jornada de estudos com um maior leque de linguagens de programação utilizadas, porém essa tendência cai ao longo dos anos, se igualando ao patamar do Windows entre 5 e 6 anos, depois retornando à tendência inicial.

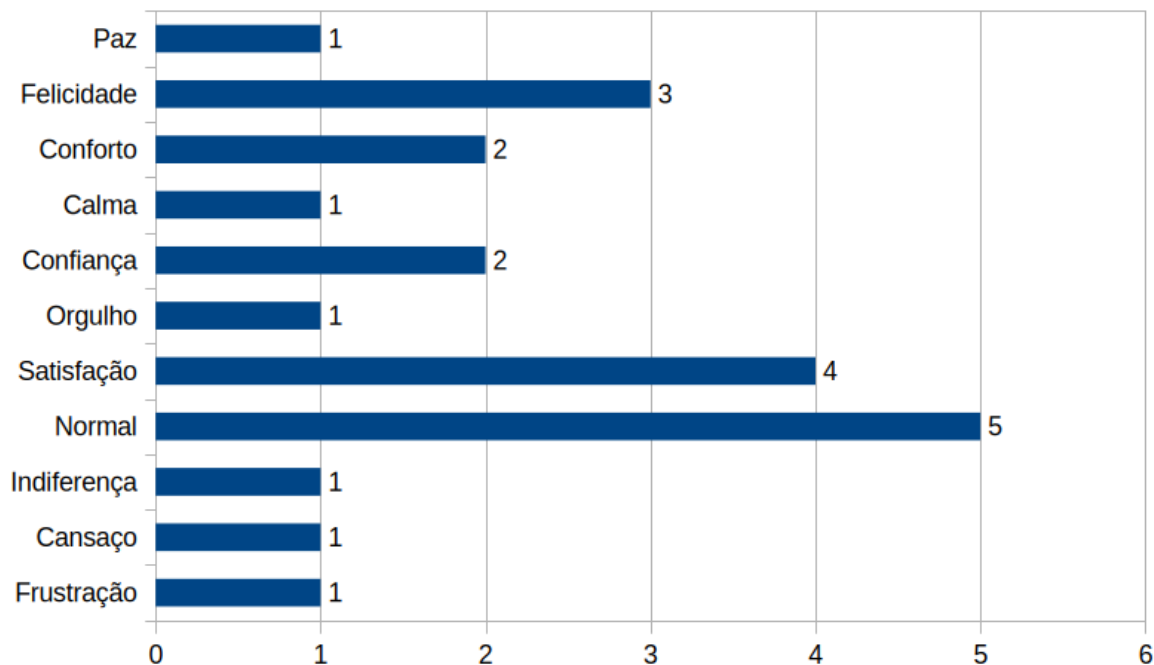
4.11 Perguntas Subjetivas

Nesta seção, abordam-se apenas as questões de âmbito subjetivo aplicadas no questionário.

4.11.1 Sentimentos Causados ao Usar um SO

Quando os discentes foram perguntados: “Descreva uma emoção que você sente ao usar este SO para programar”, obteve-se o gráfico da Figura 37:

Figura 37 - Sentimento causados pelo SO



Fonte: Autoria Própria (2026)

Verifica-se que a maioria das respostas destacaram sentimentos positivos (paz, felicidade, conforto, calma, confiança, orgulho e satisfação), alguns destacaram sentimentos negativos (frustração, esforço) e poucos descreveram sentimentos mais moderados (normal e indiferença).

4.11.2 Dificuldades e Facilidades Encontradas no SO

Para a questão: “Descreva uma situação em que seu SO principal facilitou/dificultou seu aprendizado de programação”, dentre as 38 (trinta e oito) respostas coletadas, 35 (trinta e cinco) foram consideradas importantes para a pesquisa, conforme mencionado no início deste capítulo. A maioria delas afirma que o SO influencia significativamente nesse processo.

Com base nas respostas dos discentes, o Windows se mostrou um sistema operacional com pontos fortes e fracos no que diz respeito ao aprendizado de programação. As principais vantagens citadas foram: a facilidade de uso, a compatibilidade com diversas ferramentas, como o VSCode, Python, Node.js e Java, a familiaridade do sistema, a interface intuitiva e a facilidade de instalar e configurar ferramentas de desenvolvimento, que permitiram que os discentes se concentrassem no aprendizado da programação em si, sem se frustrarem com configurações complicadas.

No entanto, certas configurações de ambientes de desenvolvimento, especialmente na linguagem C, foram citadas como um ponto de dificuldade. Problemas inesperados, como conflitos de atualização do SO, também causaram interrupções e dificultaram a identificação da causa raiz. A dependência de máquinas virtuais para executar determinados comandos ou ferramentas Linux também foi vista como ineficiente.

Além disso, a ausência de ferramentas nativas para tarefas específicas, como a falta de um editor de texto integrado, representou um obstáculo. A instalação e configuração de bibliotecas externas, como Allegro, exigiu conhecimento técnico e, em alguns casos, levou a erros de configuração. O uso do WSL (*Windows Subsystem for Linux*) foi citado como uma solução para mitigar alguns desses problemas, permitindo o uso de ferramentas e comandos utilitários do Linux.

Houve uma percepção positiva, em relação ao aprendizado, de que as distribuições Linux forçaram os discentes a entenderem melhor como o seu SO funciona.

Em última análise, a escolha do Windows por comodidade e familiaridade foi um fator importante, especialmente para aqueles que não tinham experiência com outros sistemas operacionais. A facilidade de uso, a compatibilidade com ferramentas populares e a familiaridade com o ambiente foram os principais fatores que contribuíram para o uso do Windows como principal ambiente de programação.

4.11.3 Motivações e Desmotivações Advindas do SO

Ao serem perguntados sobre: “O que mais te motiva ou desmotiva a continuar usando seu SO atual para programar”, obteve-se 21 (vinte e uma) respostas.

A principal motivação para muitos está na capacidade de personalização, na customização e na comunidade, sendo Arch Linux, Garuda Linux e Pop OS particularmente atraentes por permitirem a criação de um ambiente de trabalho totalmente adaptado às necessidades individuais. O desejo de aprender e dominar o sistema também é um forte motivador, especialmente em Pop OS e Linux Mint, já o Ubuntu se destaca pela sua eficiência.

Além disso, a compatibilidade com ferramentas e a praticidade no dia a dia impulsionam a escolha do Windows, que continua sendo o sistema preferido pela sua facilidade de uso. Entretanto, existem também desmotivações significativas. Dentre elas, a dificuldade em encontrar ferramentas e ambientes de desenvolvimento que funcionem corretamente, especialmente no Linux Mint e no Windows, a complexidade de algumas configurações, incluindo ambientes como Linux, Docker e servidores locais, problemas de otimização, particularmente em computadores menos potentes e a falta de suporte, em alguns casos, contribuem para a desmotivação.

4.11.4 Adequação do SO ao Aprendizado

Ao serem perguntados sobre: “Se você pudesse mudar algo no SO principal que utiliza para torná-lo mais adequado ao aprendizado de programação, o que seria”, todos os 38 (trinta e oito) discentes responderam, porém somente 35 (trinta e cinco) foram válidas, conforme mencionado no início deste capítulo.

A maioria dos discentes que utilizam o Windows expressam frustração com a complexidade inerente ao desenvolvimento, principalmente quando comparada com o Linux. A principal demanda é uma integração nativa e intuitiva com ambientes Linux, em especial o WSL.

Muitos desejam que o Windows simplificasse a instalação de ferramentas, bibliotecas e linguagens de programação, reduzindo a necessidade de ajustes manuais no ambiente de trabalho. Assim, como enfatizado por um discente, poderia-se focar mais na programação.

A integração do WSL como subsistema de arquivos padrão, com gerenciamento de permissões e caminhos otimizados, surge como um desejo central para facilitar o fluxo de trabalho. Além disso, há um anseio por uma maior otimização do sistema, com menos atualizações desnecessárias e maior foco em correções e novos recursos.

A ideia de ter ferramentas de desenvolvimento pré-instaladas ou uma configuração inicial voltada para programadores também é comum, bem como a adição de elementos lúdicos e interativos para tornar o aprendizado mais interessante. A compatibilidade com aplicativos e a facilidade de uso do terminal também são pontos importantes.

Para o Linux, a flexibilidade e a personalização são valores fundamentais. Usuários de Arch Linux, Garuda Linux e Linux Mint destacam a liberdade de modificar o sistema para atender às suas necessidades específicas. O Linux Mint, por sua vez, busca melhorar a interface visual e a inclusão de ferramentas de acessibilidade. O Pop OS e Ubuntu enfatizam a facilidade de uso, especialmente na instalação de programas e no desempenho, porém uma maior velocidade de transferência de arquivos é desejada. Os usuários do Ubuntu também ressaltam a incompatibilidade com aplicativos populares, como o pacote Office da Microsoft.

Além disso, alguns discentes afirmaram que não mudariam nada no seu sistema, seja porque já fizeram essas mudanças ou por outro motivo não esclarecido.

Em suma, tanto no Windows quanto no Linux, a demanda por um ambiente de desenvolvimento mais intuitivo, otimizado e integrado é evidente, com foco particular na simplificação do fluxo de trabalho e na redução da "fricção" inicial para novos programadores.

5 CONSIDERAÇÕES FINAIS

Neste trabalho foi realizado um estudo objetivando compreender os impactos do uso de SOs no aprendizado de programação pelos discentes. Como estudo de caso foi considerado o contexto dos cursos de BITI, BICT, ES e EC. Como metodologia foi realizado um *Survey*, instrumentado por meio de um questionário para considerar diferentes aspectos da relação entre SOs e programação. Como resultado, diferentes impactos foram identificados, dentre eles a presença de uma diferença entre as médias finais dos discentes quando comparados diretamente aos SOs utilizados naquele componente curricular, principalmente quando comparado às distribuições Linux em relação aos outros grupos. Além disso, foi possível perceber um direcionamento dos discentes que utilizam Windows pela facilidade de uso, pela praticidade, pela compatibilidade de *softwares*, pela familiaridade com o sistema após longos anos de exposição, pela integração intuitiva do WSL por motivos de necessidade de integração com ferramentas Linux, como servidores. Por sua vez, o uso do Linux foi apontado por aspectos como a flexibilidade e a personalização. Nesse contexto, a escolha do SO se torna, portanto, uma decisão estratégica, que deve levar em consideração as ferramentas, a compatibilidade e a filosofia de trabalho que melhor se adequam às necessidades e aos objetivos do discente. Para trabalhos futuros pretende-se estender a pesquisa para considerar outros cursos e universidades, assim como investigar contextos específicos, como o uso do SO nos componentes curriculares iniciais de programação no contexto nacional.

REFERÊNCIAS

- BENNEDSEN, Jens; CASPERSEN, Michael E. Failure rates in introductory programming. **SIGCSE Bulletin**, v. 39, n. 2, p. 32-36, 2007. DOI: <https://doi.org/10.1145/1272848.1272879>. Disponível em: <https://doi.org/10.1145/1272848.1272879>. Acesso em: 29 jun. 2026.
- BRYANT, Randal E.; O'HALLARON, David R. **Computer Systems: a programmer's perspective**. 3. ed. Boston: Pearson, 2016. Disponível em: <https://www.amazon.com.br/Computer-Systems-OHallaron-Randal-Bryant/dp/9332573905>. Acesso em: 29 jun. 2026.
- CASTRO, Matheus Sampaio. **PP - 8 (Practice Programming): uma ferramenta para o ensino de programação de maneira lúdica e visual**. 2025. Trabalho de Conclusão de Curso (Curso de Graduação em Sistemas de Informação) - Campus de Crateús, Universidade Federal do Ceará, Crateús, 2025. Disponível em: <https://repositorio.ufc.br/handle/riufc/82418>. Acesso em: 29 jun. 2026.
- CANÑAS, José Juan; BAJO, Maria Teresa; GONZALVO, Pilar. Mental models and computer programming. **International Journal of Human-Computer Studies**, v. 40, n. 5, p. 795-811, 1994. DOI: <https://doi.org/10.1006/ijhc.1994.1038>. Disponível em: <https://www.sciencedirect.com/science/article/pii/S107158198471038X>. Acesso em: 18 jun. 2026.
- CHAN, M. C.; ZHANG, J. Comparing Text-based and Graphic User Interfaces for Novice and Expert Users. In: AMIA ANNUAL SYMPOSIUM PROCEEDINGS, 2007, Washington. **Anais [...]**. Washington: American Medical Informatics Association, 2007. p. 126-130. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2655855/>. Acesso em: 29 jun. 2026.
- GUZDIAL, Mark. **Learner-centered design of computing education: research on computing for everyone**. San Rafael: Morgan & Claypool Publishers, 2015. (Synthesis Lectures on Human-Centered Informatics). Disponível em: <https://link.springer.com/book/10.1007/978-3-031-02216-6>. Acesso em: 29 jun. 2026.
- INTERNATIONAL ORGANIZATION FOR STANDARDIZATION; INTERNATIONAL ELECTROTECHNICAL COMMISSION. **ISO/IEC 25000:2014: Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Guide**. Geneva: ISO, 2014. Acesso em: 29 jun. 2026.
- JENKINS, T. On the difficulty of learning to program. In: Annual LTSN-ICS Conference, 3., 2002, Loughborough. **Anais [...]**. Disponível em: <https://steve.psy.gla.ac.uk/loaled/jenkins.html>. Loughborough: Loughborough University, 2002. p. 53-58. Acesso em: 29 jun. 2026.
- KENSKI, V. M. **Educação e tecnologias: o novo ritmo da informação**. 13. reimpressão. Campinas: Papirus, 2024. Disponível em: <https://www.amazon.com.br/Educação-Tecnologias-Novo-Ritmo-Informação/dp/8530808282>. Acesso em: 29 jun. 2026.

KOLGATIN, Oleksandr H.; HOLUBNYCHYI, Dmytro Yu.; KOLGATINA, Larysa S. The "Operating Systems" Course as a base of students' learning activity parameters investigation. In: INTERNATIONAL CONFERENCE ON COMPUTER SUPPORTED EDUCATION, 13., 2021, Online. **Anais[...]**. Online: SCITEPRESS, 2021. p. 445-452. DOI: 10.5220/0010426904450452. Disponível em: <https://www.scitepress.org/Papers/2021/120623/120623.pdf>. Acesso em: 29 jun. 2026.

LISTER, Raymond et. al.. A multi-national study of reading and tracing skills in novice programmers. In: Working Group Reports from ITICSE on Innovation and Technology In Computer Science Education, 2004, Leeds, United Kingdom. **Anais [...]**. New York: Association for Computing Machinery, 2004. p. 119-150. DOI: <https://doi.org/10.1145/1044550.1041673>. Disponível em: <https://doi.org/10.1145/1044550.1041673>. Acesso em: 28 jun. 2026.

MANZANO, José Augusto N. G. **Algoritmos: lógica para desenvolvimento de programação de computadores**. 29. ed. São Paulo: Érica, 2019. Disponível em: <https://www.amazon.com.br/Algoritmos-Desenvolvimento-Programação-Computadores-Atualizada/dp/8536531452>. Acesso em: 29 jun. 2026.

MAZIERO, Carlos A. **Sistemas operacionais: conceitos e mecanismos**. Curitiba: UFPR, 2019. Disponível em: <https://wiki.inf.ufpr.br/maziero/doku.php?id=socm:start>. Acesso em: 29 jun. 2026.

McCONNELL, Steve. **Code complete: a practical handbook of software construction**. 2. ed. Redmond: Microsoft Press, 2004. Disponível em: <https://www.amazon.com.br/Code-Complete-Steve-McConnell/dp/0735619670>. Acesso em: 29 jun. 2026.

RAUTERBERG, M. An empirical comparison of menu-selection (CUI) and desktop (GUI) computer programs carried out by beginners and experts. **Behaviour & Information Technology**, [S. l.], v. 11, n. 4, p. 227-236, 1992. Disponível em: <https://www.tandfonline.com/doi/abs/10.1080/01449299208924341>. Acesso em: 29 jun. 2026.

SILBERSCHATZ, A.; GALVIN, P. B.; GAGNE, G. **Fundamentos de sistemas operacionais**. 9. ed. Rio de Janeiro: LTC, 2016. Disponível em: <https://www.amazon.com.br/Fundamentos-Sistemas-Operacionais-Abraham-Silberschatz/dp/8521629397>. Acesso em: 29 jun. 2026.

SILVA, Rodolfo Gonçalves da et. el. IMPORTÂNCIA DA PROGRAMAÇÃO NA EDUCAÇÃO FUNDAMENTAL: PREPARANDO discentes PARA O FUTURO DIGITAL. **Revista Ibero-Americana de Humanidades, Ciências e Educação**, [S. l.], v. 10, n. 7, p. 1849–1864, 2024. DOI: 10.51891/rease.v10i7.14955. Disponível em: <https://periodicorease.pro.br/rease/article/view/14955>. Acesso em: 29 jun. 2026.

SILVA, Idovaldo Cunha da. **Um modelo de sistema tutor inteligente para o ensino no domínio de lógica de programação**. 2016. Trabalho de Conclusão de Curso – Universidade Estadual do Maranhão, São Luís, 2016. Disponível em: <https://repositorio.uema.br/jspui/handle/123456789/730>. Acesso em: 29 jun. 2026.

SOUSA, R. R. de; LEITE, F. T. Usando gamificação no ensino de programação introdutória / Using gamification in teaching introductory programming. **Brazilian Journal of Development**, [S. l.], v. 6, n. 6, p. 33338–33356, 2020. DOI: 10.34117/bjdv6n6-043. Disponível em: <https://ojs.brazilianjournals.com.br/ojs/index.php/BRJD/article/view/10995>. Acesso em: 29 jun. 2026.

TANENBAUM, Andrew S.; BOS, Herbert. **Sistemas operacionais modernos**. 4. ed. São Paulo: Pearson Education do Brasil, 2016. Disponível em: <https://www.amazon.com.br/Sistemas-operacionais-modernos-Andrew-Tanenbaum-ebook/dp/B01IPIN4W4r>. Acesso em: 29 jun. 2026.

ZIAEE, Alexander. **About FreeBSD**. The FreeBSD Project, 2025. Disponível em: <https://www.freebsd.org/about/>. Acesso em: 29 jun. 2026.