



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE GRADUAÇÃO
DEPARTAMENTO DE ENGENHARIAS E TECNOLOGIA
CURSO DE BACHARELADO EM ENGENHARIA DE SOFTWARE

Brígido Conrado de Brito Freitas

Autoavaliação no Processo de Ensino-Aprendizagem de Programação Introdutória:
um estudo experimental

PAU DOS FERROS - RN

2025

Brígido Conrado de Brito Freitas

Autoavaliação no Processo de Ensino-Aprendizagem de Programação Introdutória:
um estudo experimental

Monografia apresentada a Universidade Federal Rural do Semi-Árido como requisito para obtenção do título de Bacharel em Engenharia de Software.

Orientadora: Laysa Mabel de Oliveira Fontes,
Profa. Dra.

PAU DOS FERROS – RN

2025

© 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 nº 9.279/1996 e Direitos Autorais: Lei nº 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.

F862a FREITAS, Brígido Conrado de Brito.

Autoavaliação no processo de ensino-aprendizagem de programação introdutória: um estudo experimental / Brígido Conrado de Brito FREITAS. - 2025.

47 f. : il.

Orientador: Laysa Mabel de Oliveira Fontes.
Monografia (graduação) - Universidade Federal Rural do Semi-árido, Curso de Engenharia de Software, 2025.

1. algoritmos. 2. aprendizagem de programação. 3. autoavaliação. 4. metodologia para ensino de programação. 5. programação introdutória. I. Fontes, Laysa Mabel de Oliveira, 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 Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas 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.

Brígido Conrado de Brito Freitas

Autoavaliação no Processo de Ensino-Aprendizagem de Programação Introdutória:
um estudo experimental

Monografia apresentada a Universidade Federal Rural do Semi-Árido como requisito para obtenção do título de Bacharel em Engenharia de Software.

Defendida em: 21/07/2025.

BANCA EXAMINADORA

Laysa Mabel de Oliveira Fontes, Profa. Dra. (UFERSA)
Presidente

Alysson Filgueira Milanez, Prof. Dr. (UFERSA)
Membro Examinador

Reudismam Rolim de Sousa, Prof. Dr. (UFERSA)
Membro Examinador

AGRADECIMENTOS

Quero agradecer primeiramente a minha família, em particular minha mãe Régia Meire por ser um exemplo de paciência e dedicação; ao meu pai Francisco Tarcísio que sempre esteve próximo me incentivando a continuar; a minha irmã Samalia Line que tirou minha paciência por grande parte da minha vida (e eu a dela) e que hoje é uma grande companheira; e ao meu sobrinho Matheus que tirei a paciência, mas que sempre está presente.

Ao grupo “Games”, formado pelos meus amigos de longa data Tiago Lemos (Titi/Kenray), João Douglas (JD/DagWho) e Rafael Henrique (Rafa/Krocios), minha eterna gratidão por mais de uma década de amizade. Mesmo com a distância física, jamais deixamos de manter contato, entre vitórias épicas, derrotas que viraram piada, construímos algo que vai muito além das partidas. Agradeço também aos agregados desse grupinho que carrego com tanto carinho, em especial Leticia Kesia, que participou de grandes momentos que tive.

Agradeço aos meus amigos que se tornaram presentes em minha vida, a citar: Pedro David, Maria Fabia, Cayo Ravel, Djanira Girão, Josymara Lima e em especial meu amigo Sebastião Costa, que me ajudou muito nessa jornada.

Agradeço aos meus amigos que a internet me proporcionou, no RPG de mesa: Marcus (Tid), Barbara e Paulo Augusto (mestre), na polícia do RP: Murilo Martins (Capitão Nome Sobrenome) e Leonardo Quartaroli (1º Tenente Carlos G.).

A UFERSA, na qual eu tive oportunidade de obter um amadurecimento pessoal e profissional, aos meus professores que tiveram um papel importante na construção da minha jornada, sendo grande fonte de inspiração.

Por fim, meu muito obrigado à Profa. Dra. Laysa Mabel, minha orientadora, por sua dedicação, inteligência e paciência. Sua orientação foi fundamental para a realização deste trabalho, e sua postura é, para mim, uma grande fonte de inspiração e espero levar toda sua força, garra e dedicação para minha vida.

Dedico este trabalho a todos os que caminharam ao meu lado, os que foram citados aqui e aqueles que, mesmo em silêncio, contribuíram de alguma forma para a minha formação.

“Uma lição indolor não tem sentido. Aquele que nada sacrifica, nada é capaz de obter...”.

Hiromu Arakawa

RESUMO

Trata-se de um estudo exploratório, com abordagem mista (quantitativa e qualitativa), realizado em uma turma do componente curricular de Algoritmos, do curso de Bacharelado Interdisciplinar em Tecnologia da Informação da Universidade Federal Rural do Semi-Árido, Campus Pau dos Ferros, cujo objetivo é fomentar a autoavaliação no processo de ensino-aprendizagem. A proposta surgiu diante das dificuldades enfrentadas por alunos ingressantes, refletidas nos altos índices de evasão e reprovação em componentes curriculares de programação introdutória. A metodologia adotada foi composta por autoavaliações estruturadas em cenários práticos e dicas orientativas, com o intuito de promover a autorreflexão, a identificação de dificuldades e o desenvolvimento da autonomia dos discentes. Para apoiar a aplicação da intervenção, também foi desenvolvido um sistema para organização, coleta e armazenamento das respostas dos estudantes. A análise dos dados revelou relações significativas entre as dicas acessadas e as dificuldades relatadas pelos discentes, indicando avanços na autorregulação da aprendizagem. Os resultados confirmam a efetividade da proposta, com 89,5% de aprovação entre os discentes e avaliação positiva do docente responsável pelo componente curricular. Portanto, conclui-se que a autoavaliação, aliada a estratégias pedagógicas contextualizadas, pode contribuir de forma relevante para a aprendizagem em componentes de programação introdutória. O trabalho ainda disponibiliza os artefatos desenvolvidos, com vistas à replicação e aprimoramento da metodologia proposta.

Palavras-chave: algoritmos; aprendizagem de programação; autoavaliação; metodologia para ensino de programação; programação introdutória.

ABSTRACT

This is an exploratory study with a mixed-methods approach (quantitative and qualitative), conducted in an introductory programming course (Algorithms) within the Interdisciplinary Bachelor's Program in Information Technology at the Federal Rural University of the Semi-Arid Region (UFERSA), Pau dos Ferros Campus. The main objective was to promote self-assessment in the teaching-learning process. The initiative emerged from the challenges faced by incoming students, reflected in high dropout and failure rates in introductory programming subjects. The methodology included structured self-assessments based on practical scenarios and guidance tips, aiming to foster self-reflection, identify learning difficulties, and encourage student autonomy. To support the intervention, a system was developed for organizing, collecting, and storing student responses. Data analysis revealed significant relationships between the tips accessed and the difficulties reported, indicating progress in students' self-regulated learning. The results confirmed the effectiveness of the proposal, with an 89.5% approval rate and positive feedback from the course instructor. Therefore, the study concludes that self-assessment, combined with contextualized pedagogical strategies, can significantly contribute to learning in introductory programming courses. The study also provides the developed artifacts to support replication and further enhancement of the proposed methodology.

Keywords: algorithms; programming learning; self-assessment; teaching methodology for programming; introductory programming.

LISTA DE FIGURAS

Figura 1 - Fluxo de atividades da metodologia proposta.....	26
Figura 2 - Tela de login.....	30
Figura 3 - Primeira seção da autoavaliação.....	30
Figura 4 - Exemplo de consulta à Dica 1.....	31
Figura 5 - Segunda seção da autoavaliação.....	32
Figura 6 - Terceira seção da autoavaliação.....	33

LISTA DE GRÁFICOS

Gráfico 1 - Distribuição dos conteúdos de maior dificuldade.....	39
Gráfico 2 - Principais fatores relacionados às dificuldades relatadas.....	39
Gráfico 3 - Percepção dos alunos quanto à identificação das dificuldades.....	40
Gráfico 4 - Percepção dos alunos quanto aos <i>feedbacks</i>	41
Gráfico 5 - Grau de aprovação da metodologia aplicada.....	41

LISTA DE QUADROS

Quadro 1 - Comparativo entre trabalhos relacionados e a proposta deste estudo.....	23
Quadro 2 - Classificação das dicas.....	25
Quadro 3 - Exemplo de cenário e suas respectivas dicas.....	25
Quadro 4 - Conteúdo recomendado para cada autoavaliação.....	27
Quadro 5 - Utilização de dicas por autoavaliação na primeira unidade.....	35
Quadro 6 - Dificuldades apontadas pelos discentes na primeira unidade.....	36
Quadro 7 - Utilização de dicas por autoavaliação na segunda unidade.....	36
Quadro 8 - Dificuldades apontadas pelos discentes na segunda unidade.....	37
Quadro 9 - Utilização de dicas por autoavaliação na terceira unidade.....	37
Quadro 10 - Dificuldades apontadas pelos discentes na terceira unidade.....	38

LISTA DE ABREVIATURAS E SIGLAS

UFERSA	Universidade Federal Rural do Semi-Árido.
BITI	Bacharelado Interdisciplinar em Tecnologia da Informação.
TI	Tecnologia da Informação.

SUMÁRIO

1 INTRODUÇÃO.....	15
1.1 Problemática.....	15
1.2 Justificativa.....	16
1.3 Objetivos.....	17
1.3.1 Objetivo Geral.....	17
1.3.2 Objetivos Específicos.....	17
1.4 Organização do Trabalho.....	18
2 REFERENCIAL TEÓRICO.....	19
2.1 Programação Introdutória.....	19
2.2 Autoavaliação.....	20
2.3 Trabalhos Relacionados.....	21
3 INTERVENÇÃO METODOLÓGICA.....	24
3.1 Processo Metodológico.....	24
3.1.1 Planejamento da Metodologia.....	26
3.1.2 Aplicação do Questionário Social.....	27
3.1.3 Explicação do Conteúdo.....	27
3.1.4 Elaboração da Autoavaliação.....	28
3.1.5 Aplicação da Autoavaliação.....	28
3.1.6 Sumarização dos Resultados da Autoavaliação.....	28
3.1.7 Aplicação do Questionário Final com os Discentes.....	29
3.1.8 Aplicação do Questionário Final com os Docentes.....	29
3.1.9 Sumarização Final dos Resultados para Análise.....	29
3.2 Sistema de Suporte à Aplicação das Autoavaliações.....	29
3.3 Aplicação do Estudo Experimental.....	33
4 RESULTADOS E DISCUSSÃO.....	35
4.1 Resultados da Primeira Unidade.....	35
4.2 Resultados da Segunda Unidade.....	36

4.3 Resultados da Terceira Unidade.....	37
4.4 Resultados Sobre o Impacto da Intervenção Metodológica.....	38
4.4.1 Conteúdos de Maior Dificuldade.....	38
4.4.2 Principais Motivos de Dificuldades.....	39
4.4.3 Percepção Sobre as Autoavaliações.....	40
4.4.4 <i>Feedbacks</i> das Autoavaliações.....	40
4.4.5 Aprovação da Metodologia Aplicada.....	41
4.4.6 Percepções dos Discentes sobre a Metodologia Aplicada.....	42
4.4.7 Percepção do Docente sobre a Metodologia Aplicada.....	43
5 CONSIDERAÇÕES FINAIS.....	44
5.1 Limitações do Estudo.....	44
5.2 Trabalhos Futuros.....	45
REFERÊNCIAS.....	46

1 INTRODUÇÃO

Este capítulo apresenta uma visão geral deste trabalho e está organizado da seguinte forma: a Seção 1.1 apresenta a problematização; a Seção 1.2 explicita os argumentos que justificam o desenvolvimento deste trabalho; a Seção 1.3 apresenta os objetivos; e, por fim, a Seção 1.4 apresenta a organização deste trabalho.

1.1 Problemática

A programação introdutória desempenha um papel importante nos cursos da área da computação, sendo normalmente empregada por meio de componentes curriculares que visam introduzir os estudantes aos fundamentos da lógica computacional, abrangendo comandos, sintaxe e semântica da linguagem utilizada (Berssanette; Frencisco, 2018).

Entretanto, o processo de ensino-aprendizagem dos componentes curriculares de programação introdutória vem sendo motivo de preocupação devido aos altos índices de reprovação e evasão dos cursos, nos primeiros semestres (Santiago; Kronbauer, 2016). Em um estudo realizado por Souza, Batista e Barbosa (2016), foi constatado que esses problemas surgem porque a aprendizagem dos conceitos sintáticos e semânticos de programação constitui uma das maiores dificuldades dos alunos. Nesse contexto, Figueiredo e Garcia-Peñalvo (2018) identificaram alguns fatores que levam aos problemas descritos anteriormente, a citar: (i) os conceitos abstratos envolvidos na programação; (ii) as habilidades mentais necessárias para a decomposição de problemas; e (iii) para muitos estudantes, é o primeiro contato com o pensamento computacional e com a programação.

Analisando especificamente o componente curricular de programação introdutória do curso de Bacharelado Interdisciplinar em Tecnologia da Informação (BITI) da Universidade Federal Rural do Semi-Árido (UFERSA), Campus Pau dos Ferros, Moreira *et al.* (2018) identificaram dificuldades relacionadas à construção do raciocínio lógico e ao domínio da sintaxe da linguagem de programação. Complementando esse panorama, Holanda, Coutinho e Fontes (2018) apontam que as individualidades dos estudantes, especialmente, no que diz respeito ao ritmo de aprendizagem, constituem outro fator determinante para o insucesso.

Nesse cenário, diversas pesquisas vêm sendo desenvolvidas com o intuito de propor estratégias que auxiliem o processo de ensino-aprendizagem de programação (Silva *et al.*, 2015). Entre essas estratégias, destaca-se a necessidade de fomentar a autonomia dos alunos, incentivando-os a reconhecer e enfrentar suas dificuldades de maneira ativa (Silva *et al.*, 2015).

Para tanto, Silva (2013) argumenta que a participação do aluno no processo avaliativo é essencial, permitindo-lhe uma maior compreensão de suas limitações. Nessa linha, Vieira (2013) enfatiza que a autoavaliação constitui uma ferramenta pedagógica eficaz, pois promove ao discente um envolvimento crítico e reflexivo. Já Santos (2014) afirma que todo processo de regulação e autoavaliação contribui diretamente para a progressão da aprendizagem, sendo o aluno a peça central desse processo.

Diante desse contexto, emerge a questão central de pesquisa deste trabalho, que busca responder: **a aplicação de autoavaliações pode auxiliar no processo de ensino-aprendizagem de programação introdutória?**

1.2 Justificativa

O crescente interesse pela programação, impulsionado por sua relevância social e profissional, tem ampliado a diversidade de perfis entre os estudantes dos cursos da área, exigindo, portanto, novas abordagens pedagógicas (Marasco *et al.*, 2017). Graafsma *et al.* (2023) destacam que essa diversificação levou à simplificação dos métodos de ensino e ao desenvolvimento de linguagens de programação mais próximas da linguagem natural.

Os componentes curriculares de programação introdutória são fundamentais para o desenvolvimento de competências lógico-computacionais, uma vez que visam capacitar os alunos a resolverem problemas por meio de soluções baseadas em algoritmos e estruturas de dados (Pimentel *et al.*, 2003). Para Amaral (2015), esses componentes curriculares fornecem as bases do raciocínio lógico que sustentarão a formação do aluno ao longo do curso.

Apesar da importância reconhecida, o ensino de programação nos níveis iniciais ainda enfrenta desafios consideráveis (Coşkunserçe, 2023). Segundo Medeiros *et al.* (2019), mesmo com avanços metodológicos, muitos alunos demonstram dificuldades em aspectos cruciais do processo de aprendizagem, como a compreensão dos problemas propostos, a elaboração de soluções adequadas e a avaliação crítica dos resultados. Essa realidade é refletida nas elevadas taxas de reprovação, com estudos apontando que cerca de um terço dos estudantes não concluem com êxito os componentes introdutórios de programação (Cheng *et al.*, 2023).

É notável o crescente interesse acadêmico em investigar as dificuldades enfrentadas pelos alunos nesses componentes curriculares (Srivatanakul, 2022). Contudo, apesar do volume de pesquisas existentes, ainda é desafiador construir uma visão unificada do problema, dada a variedade de contextos institucionais e culturais envolvidos (Souza; Batista; Barbosa, 2016; Figueiredo; García-Peñalvo, 2018).

Tarouco *et al.* (2009) defendem que o sucesso na aplicação de novas metodologias depende da compreensão do ambiente em que o processo educacional está inserido, considerando o perfil dos alunos, os métodos de ensino dos professores, a infraestrutura disponível, entre outros fatores.

Assim, torna-se evidente que o ensino de programação introdutória é um processo complexo, influenciado por múltiplas variáveis. Diante disso, optou-se por desenvolver um estudo exploratório, uma vez que esse tipo de abordagem é recomendado em situações em que o fenômeno investigado ainda carece de aprofundamento empírico e teórico em contextos específicos. Segundo Gil (2008), estudos exploratórios são apropriados para proporcionar maior familiaridade com o problema, formular hipóteses e levantar elementos para investigações futuras, objetivos esses que se alinham diretamente à proposta deste trabalho.

1.3 Objetivos

A Subseção 1.3.1 define o objetivo geral deste trabalho e a Subseção 1.3.2 apresenta os objetivos específicos.

1.3.1 Objetivo Geral

O objetivo geral deste trabalho consiste em aplicar uma intervenção metodológica no componente curricular de Algoritmos do curso de BITI da UFERSA, Campus Pau dos Ferros, com o intuito de promover a autoavaliação dos discentes e fornecer subsídios aos docentes para identificação das principais dificuldades de aprendizagem, contribuindo para o aprimoramento do processo educacional.

1.3.2 Objetivos Específicos

Os objetivos específicos descritos a seguir foram definidos para atingir o objetivo geral deste trabalho.

- Desenvolver um sistema para auxiliar na aplicação das autoavaliações;
- Aplicar a intervenção metodológica no componente curricular de Algoritmos do curso de BITI da UFERSA, Campus Pau dos Ferros;
- Disponibilizar os principais artefatos da intervenção metodológica em um diretório *online*.

1.4 Organização do Trabalho

O presente trabalho está organizado da seguinte forma: o Capítulo 2 apresenta a fundamentação teórica utilizada para o desenvolvimento deste trabalho; o Capítulo 3 apresenta a interversão metodológica proposta; o Capítulo 4 apresenta os resultados obtidos e suas respectivas discussões; e, por fim, o Capítulo 5 expõe as considerações finais, limitações do estudo e sugestões para trabalhos futuros.

2 REFERENCIAL TEÓRICO

Este capítulo descreve a base teórica que fundamenta a elaboração deste trabalho, estando organizado da seguinte forma: a Seção 2.1 apresenta uma contextualização sobre programação introdutória; a Seção 2.2 apresenta a importância da autoavaliação; e a Seção 2.3 discute trabalhos correlatos ao tema desta pesquisa.

2.1 Programação Introdutória

Os componentes curriculares de programação introdutória representam um dos primeiros grandes desafios para os alunos ingressantes em cursos da área de computação, visto que muitos deles possuem pouca ou nenhuma experiência prévia com a prática de programação (Falkner *et al.*, 2014). Segundo Borges (2000), esses componentes curriculares oferecem ao aluno o primeiro contato com os passos necessários para a resolução de problemas computacionais, além de promoverem a familiarização com sintaxe, semântica, ambientes de desenvolvimento, construtores de programas e técnicas de testes.

A programação introdutória tem como objetivo principal habilitar os estudantes a resolverem problemas por meio da construção de algoritmos e da aplicação do raciocínio lógico na criação de soluções computacionais (Luxton-Reilly *et al.*, 2018).

Para Barcelos e Silveira (2012), a construção do pensamento computacional possibilita aos alunos ingressantes o desenvolvimento intelectual e a construção do raciocínio lógico. Nesse contexto, Petry (2005) afirma que o desenvolvimento do raciocínio lógico, relacionado à programação de computadores, favorece a promoção de resoluções eficazes de problemas.

Amaral (2015) observa que, embora as tecnologias sofram constantes mudanças e aperfeiçoamentos que impactam diretamente o contexto pedagógico dos cursos de tecnologia, o desenvolvimento da lógica obtido nas etapas iniciais se mantém inalterado, pois o pensamento lógico independe da tecnologia ou da linguagem empregada.

Gerhard e Brusilovsky (2001) elencam algumas competências que os estudantes precisam possuir e desenvolver nesse processo inicial para lograrem êxito acadêmico, a saber: (i) capacidade de abstração do problema, bem como das estratégias necessárias para resolvê-lo; (ii) conhecimento mínimo da sintaxe e da semântica da linguagem utilizada; e (iii) compreensão do processo de compilação do código. Farias *et al.* (2012) acrescentam que também é necessário que o aluno desenvolva habilidades para solucionar problemas de forma simples.

Sob essa perspectiva, Nouri *et al.* (2019) argumentam que o desenvolvimento do pensamento computacional integra não apenas conhecimentos técnicos de programação, mas também competências amplamente valorizadas no século XXI, como a capacidade de pensar criticamente, resolver problemas de maneira eficaz e gerenciar o próprio processo de aprendizagem em um contexto tecnológico em constante transformação.

Do ponto de vista dos docentes, ensinar programação não significa apenas repassar comandos e instruções aos alunos, mas sim conduzi-los por um processo que envolve lógica, raciocínio e a habilidade de compreender um problema e planejar sua resolução (Gossmann, 2017).

2.2 Autoavaliação

A autoavaliação pode ser definida como um exercício de autorreflexão, no qual o estudante analisa, de maneira crítica, suas ações, produções, preferências, capacidades, competências e desempenho. Tal prática permite não apenas o reconhecimento de suas dificuldades, mas também a compreensão das causas que as originam, promovendo um aprendizado mais consciente e autônomo (Silva, Fontes e Freitas, 2022).

De acordo com Nascimento (2024), a autoavaliação deve ser compreendida como um meio estratégico para atingir os objetivos educacionais, e não como um fim. Mesmo inserida em contextos específicos, essa prática pode contribuir de forma decisiva para a construção de um processo de aprendizagem mais democrática, crítica e participativa.

Lopes (2018) reforça essa perspectiva ao afirmar que, ao conduzir sua própria autoavaliação, o estudante fortalece seu protagonismo no processo formativo. Tal protagonismo implica a capacidade de refletir criticamente sobre sua trajetória de aprendizagem, reconhecer potencialidades e desafios, e estabelecer um diálogo mais significativo com os docentes, que atuam como mediadores e facilitadores desse processo reflexivo.

A promoção da autonomia discente, no entanto, não exime o papel do professor. Conforme apontam Both e Brandalise (2018), é fundamental que o docente oriente e acompanhe os estudantes ao longo do processo avaliativo, a fim de garantir que estes sejam capazes de compreender suas dificuldades e identificar, com base em seus contextos, as estratégias mais adequadas para superá-las.

Os benefícios da autoavaliação são amplos e impactam diretamente o desenvolvimento acadêmico e pessoal dos estudantes. Essa prática favorece a construção de uma postura reflexiva, com efeitos positivos na vida acadêmica e profissional, ao mesmo tempo em que

possibilita a identificação do nível de compreensão dos conteúdos. Ao reconhecer suas limitações, o aluno torna-se mais apto a traçar estratégias para superá-las, promovendo seu crescimento pessoal e enfrentando com mais preparo os desafios em diferentes contextos (Capellato *et al.*, 2020).

2.3 Trabalhos Relacionados

As dificuldades relacionadas à aprendizagem de programação introdutória têm sido associadas aos altos índices de reprovação e evasão em cursos de computação, sendo uma temática amplamente discutida na literatura acadêmica, conforme os exemplos a seguir.

Visando à utilização de uma ferramenta que auxiliasse na autoavaliação dos alunos, Sirotheau *et al.* (2011) realizaram um estudo com a ferramenta *JavaTool*, integrada ao *Moodle*, para possibilitar o *feedback* entre os estudantes do curso de Computação. Segundo os autores, os resultados mostraram-se promissores, pois os estudantes demonstraram competência na autoavaliação, concentrando esforços no *feedback* colaborativo, o que reduziu a sobrecarga dos professores e promoveu maior autonomia dos alunos.

França e Tedesco (2015) realizaram uma revisão sistemática da literatura com o objetivo de identificar trabalhos que abordassem a temática da autoavaliação na aprendizagem de programação. Os autores destacaram a escassez de estudos voltados ao ensino básico e identificaram três estratégias mais utilizadas no processo de autoavaliação: (i) formulário/questionário; (ii) sistemas de apoio; e (iii) pré e pós-teste. Por fim, reforçaram que as intervenções de autoavaliação são recursos eficazes, pois favorecem o aprendizado dos envolvidos.

Chagas, Oliveira e Tavares (2016) desenvolveram uma arquitetura pedagógica com o intuito de expandir e apoiar o processo de ensino, aprendizagem e autoavaliação nos componentes de programação. No módulo do aluno, é possível realizar atividades individuais ou colaborativas, compartilhar atividades desenvolvidas, expor dúvidas e sugerir soluções para os colegas. Já no módulo do professor, é possível cadastrar e recomendar atividades pelo grau de dificuldade identificada, criar listas de exercícios e definir tarefas colaborativas com base nas características dos alunos. Entre os benefícios observados, destaca-se a possibilidade de monitoramento contínuo do progresso dos discentes e o desenvolvimento de suas capacidades intelectuais.

Vahldick *et al.* (2018) analisaram os efeitos da aplicação de avaliações frequentes ao longo do semestre. A proposta consistiu na aplicação de oito avaliações em dezoito semanas,

com o objetivo de manter alunos e professores cientes do progresso acadêmico. Embora tenha havido correlação positiva entre a motivação dos alunos e seu desempenho, os resultados foram insatisfatórios, com taxa de aprovação de apenas 27%. A transição do ensino de pensamento computacional para a linguagem Java, associada à ausência de *feedback* contínuo, foi apontada como fator de sobrecarga, impactando negativamente a motivação e o desempenho. O estudo reforça a importância do acompanhamento constante e do fornecimento de *feedback* ao longo de todo o processo de aprendizagem.

Complementarmente, Holanda e Coutinho (2022) desenvolveram o *World Prog*, um jogo educacional voltado à aprendizagem lúdica de conceitos básicos de programação, criado com a *Game Engine Unity*. O jogo foi avaliado por meio de teste preliminar com possíveis usuários, que relataram facilidade de uso e o consideraram adequado ao ensino de programação introdutória. O uso de desafios facilitou a abstração dos conteúdos e contribuiu para o engajamento dos participantes, reforçando a importância de metodologias inovadoras como jogos educacionais.

Mais recentemente, Lima e Menezes (2024) realizaram uma revisão sistemática da literatura com foco em publicações entre 2019 e 2023, analisando 85 estudos sobre as principais dificuldades enfrentadas por estudantes iniciantes na aprendizagem de programação. Os principais obstáculos identificados foram: dificuldade na resolução de problemas, construção de estruturas lógicas e compreensão da sintaxe das linguagens. Aspectos emocionais como ansiedade, frustração e desmotivação também foram destacados como barreiras significativas. Como estratégias para mitigar essas dificuldades, os autores apontam o uso de metodologias ativas, ferramentas visuais e abordagens baseadas em gamificação, que proporcionam um aprendizado mais interativo, lúdico e centrado no aluno. A diversidade de abordagens pedagógicas analisadas reforça que não há uma solução única, mas sim a necessidade de adaptação das metodologias ao perfil dos estudantes e ao contexto educacional.

Em uma perspectiva particular, este trabalho avalia a utilização de uma proposta metodológica baseada em autoavaliação em uma turma do componente curricular de Algoritmos do curso de BITI da UFERSA, Campus Pau dos Ferros, com os seguintes objetivos: (i) auxiliar o processo de ensino, fornecendo ao professor relatórios de desempenho dos discentes após cada autoavaliação, permitindo intervenções contínuas e personalizadas; e (ii) promover a autonomia dos alunos no processo de aprendizagem, permitindo que identifiquem suas dificuldades e desenvolvam estratégias para superá-las.

Diferentemente dos trabalhos anteriores, esta abordagem integra sistematicamente autoavaliação estruturada, suporte pedagógico orientativo e uma ferramenta tecnológica

desenvolvida para esse fim, configurando-se como uma contribuição relevante e replicável para o processo de ensino-aprendizagem de programação introdutória. O Quadro 1 apresenta um comparativo entre os trabalhos relacionados e a proposta aqui desenvolvida, com o intuito de evidenciar suas singularidades e avanços.

Quadro 1 - Comparativo entre trabalhos relacionados e a proposta deste estudo

TRABALHO	FOCO PRINCIPAL	ABORDAGEM UTILIZADA	DIFERENÇA EM RELAÇÃO A ESTE TRABALHO
Sirotheau et al. (2011)	Utilização do JavaTool para apoio à autoavaliação e ao <i>feedback</i> colaborativo entre estudantes.	Integração com Moodle para reduzir a sobrecarga docente e promover a autonomia dos discentes.	Não oferece acompanhamento com geração de relatórios personalizados para subsidiar a atuação docente.
França e Tedesco (2015)	Revisão da literatura sobre estratégias de autoavaliação aplicadas ao ensino de programação.	Mapeamento de métodos utilizados, como questionários, sistemas de apoio e estratégias de pré e pós-teste.	Trata-se de uma revisão teórica que não apresenta uma proposta metodológica aplicada com mecanismos estruturados de <i>feedback</i> .
Chagas, Oliveira e Tavares (2016)	Arquitetura pedagógica voltada ao processo de ensino-aprendizagem, com suporte à prática de autoavaliação.	Plataforma composta por módulos específicos para alunos e professores, com ênfase na colaboração e no acompanhamento do processo formativo.	Foco ampliado em recursos pedagógicos, sendo a autoavaliação tratada como funcionalidade complementar e não como elemento central.
Vahldick et al. (2018)	Implementação de avaliações periódicas distribuídas ao longo do semestre.	Realização de oito avaliações ao longo de 18 semanas, com análise da motivação e do desempenho dos estudantes.	Falta de <i>feedback</i> contínuo e de ênfase na autoavaliação, o que resultou em desempenho insatisfatório.
Holanda e Coutinho (2022)	Jogo desenvolvido para o ensino lúdico de conceitos básicos de programação.	Jogo criado para promover engajamento e facilitar a abstração de conceitos de programação.	Adota elementos de gamificação, porém sem ênfase na avaliação ou na autoavaliação estruturada.
Lima e Menezes (2024)	Revisão sistemática das principais dificuldades enfrentadas na aprendizagem de programação.	Identificação de barreiras cognitivas e emocionais, com recomendações para o uso de abordagens ativas e gamificadas.	Não apresenta metodologia prática, oferecendo análise teórica das dificuldades e recomendações gerais para intervenção.
Este trabalho	Proposta metodológica fundamentada em autoavaliação sistemática, complementada pela geração de relatórios detalhados.	Aplicação de autoavaliações em cenários práticos de programação, acompanhadas de dicas orientativas e relatórios de desempenho para suporte a intervenções pedagógicas.	Inova ao integrar autoavaliação estruturada, autorreflexão guiada e suporte direto ao professor por meio de dados objetivos e sistematizados.

Fonte: Autoria própria (2025)

3 INTERVENÇÃO METODOLÓGICA

Este trabalho trata-se de um estudo exploratório, com abordagem mista (quantitativa e qualitativa), cujo objetivo é investigar, de forma inicial, como a autoavaliação pode contribuir para o processo de ensino-aprendizagem do componente curricular de Algoritmos.

Este capítulo está estruturado da seguinte forma: a Seção 3.1 descreve o processo metodológico adotado no experimento; a Seção 3.2 apresenta o sistema desenvolvido para apoiar a aplicação das autoavaliações; e a Seção 3.3 descreve a aplicação do estudo experimental realizado.

3.1 Processo Metodológico

A intervenção metodológica proposta neste trabalho permite identificar os perfis e as principais características das turmas, com foco na autoavaliação da aprendizagem dos discentes. Trata-se de um instrumento que visa facilitar tanto o processo de aprendizagem quanto o desenvolvimento da autonomia dos estudantes. Segundo Silva (2013), a autoavaliação promove a autonomia, torna a aprendizagem mais responsável e consciente, além de auxiliar os alunos em seu processo de aprendizagem.

É importante destacar que esta proposta representa um aprimoramento da intervenção metodológica desenvolvida por Freitas (2020), que focava no uso de dicas para apoiar a resolução de questões em avaliações de componentes de programação introdutória, realizando a análise das dificuldades somente após os resultados obtidos. A abordagem atual mantém os fundamentos daquela intervenção, mas avança ao centralizar a autoavaliação, com o objetivo de estimular a reflexão individual e o protagonismo dos discentes.

No âmbito metodológico adotado, cada autoavaliação é composta por um cenário que representa um problema de programação, o qual demanda a aplicação de um conteúdo específico ou a combinação de múltiplos conteúdos para sua solução. Essa abordagem permite que os estudantes identifiquem, por meio das autoavaliações, suas dificuldades relativas aos conteúdos abordados.

Para apoiar a resolução de cada cenário, são disponibilizadas três dicas aos estudantes, que podem optar por utilizá-las conforme suas necessidades. O uso dessas dicas favorece a autorreflexão, ao possibilitar que os alunos reconheçam suas principais dificuldades em relação ao conteúdo. A categorização das dicas está apresentada no Quadro 2.

Quadro 2 - Classificação das dicas

DICA 1 INTERPRETAÇÃO DO CENÁRIO	Apresenta as entradas necessárias, as saídas esperadas e o processamento para transformar as entradas nas saídas de dados.
DICA 2 SINTAXE	Indica a sintaxe das estruturas necessárias para a resolução do cenário.
DICA 3 LÓGICA DE PROGRAMAÇÃO	Define a lógica de programação indicada para a resolução do cenário, em forma de descrição narrativa.

Fonte: Adaptado de Holanda, Coutinho e Fontes (2018)

Essa classificação baseia-se em um estudo conduzido por Moreira *et al.* (2018), que identificou as principais dificuldades enfrentadas pelos estudantes no componente curricular de Algoritmos no curso de BITI da UFERSA, Campus Pau dos Ferros. Cabe destacar que essas dicas são específicas para questões (cenários) de natureza subjetiva, ou seja, aquelas em que os alunos devem desenvolver um algoritmo ou programa. O Quadro 3 apresenta um exemplo trivial de cenário e suas respectivas dicas para ilustrar a estruturação de uma autoavaliação.

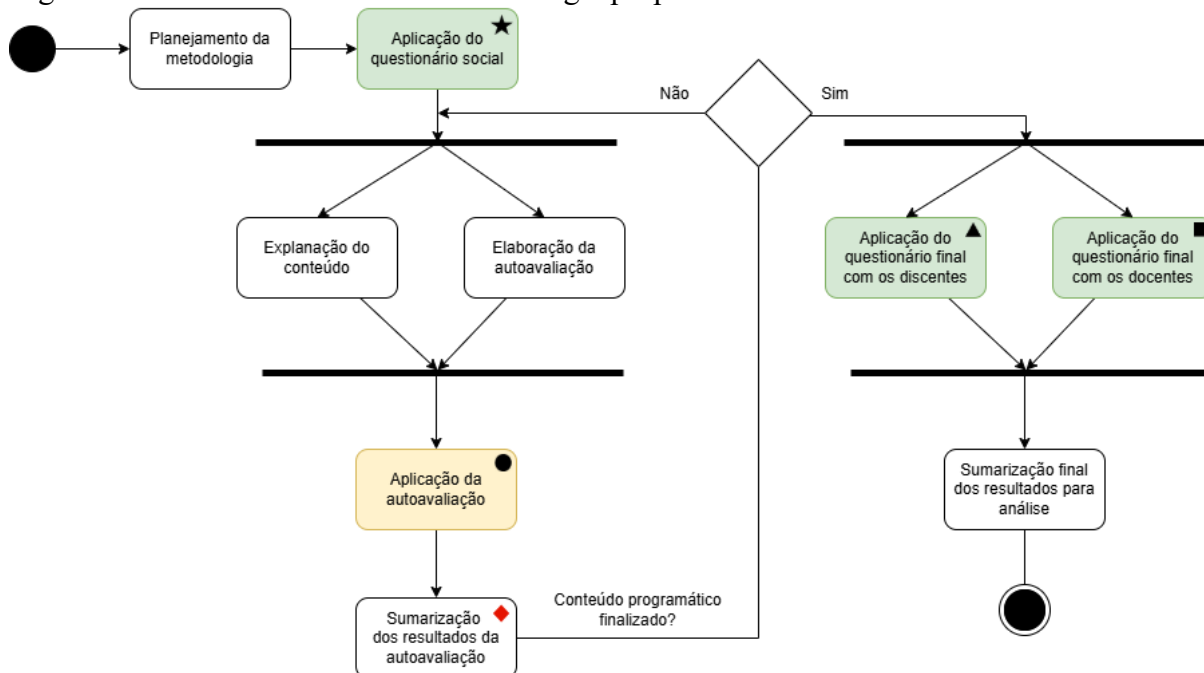
Quadro 3 - Exemplo de cenário e suas respectivas dicas

CENÁRIO: Faça um algoritmo em pseudocódigo para calcular a divisão de um número por outro. O algoritmo só deve realizar a divisão se o divisor for diferente de zero. Caso contrário, o algoritmo deve informar que o resultado é indeterminado.
DICA 1: Dados de entrada: dois números (A e B). Dados de saída: o resultado da divisão de A por B ou resultado indeterminado. Processamento: realizar a seguinte verificação: se B for igual zero, deve-se informar que o resultado é indeterminado, senão deve-se realizar a divisão de A por B (armazenando o resultado em R) e exibir o resultado da divisão (valor de R).
DICA 2: Este cenário necessitará da estrutura de decisão se-senao , conforme a sintaxe a seguir: se (<condição>) entao <bloco-v> senao <bloco-f> fimse
DICA 3: Solicite dois números. Receba os dois números, sendo A o dividendo e B o divisor. Se B for igual a zero: Informe que o resultado é indeterminado. Senão: Realize a divisão de A por B e armazene o resultado em R. Informe o resultado da divisão (valor armazenado em R).

Fonte: Autoria própria (2025)

Com o intuito de esclarecer o processo necessário para a aplicação da metodologia proposta, a Figura 1 apresenta o fluxo de atividades a ser seguido ao longo do período de intervenção, correspondente a um semestre letivo.

Figura 1 - Fluxo de atividades da metodologia proposta



Fonte: Autoria própria (2025)

Os retângulos destacados em verde, ilustrados na Figura 1, indicam as fases que envolvem a aplicação de questionários, enquanto o retângulo em amarelo refere-se ao sistema desenvolvido para auxiliar na aplicação das autoavaliações (ver Seção 3.2). Os questionários foram aplicados por meio do *Google Forms*, devido à facilidade de organização dos dados. Os principais artefatos utilizados na intervenção, incluindo os questionários, estão disponibilizados em um diretório *online*¹. As atividades definidas na Figura 1 são descritas nas subseções a seguir.

3.1.1 Planejamento da Metodologia

Essa etapa requer um conhecimento aprofundado da metodologia a ser aplicada, bem como do planejamento das ações ao longo do semestre letivo. Entre as ações previstas, está a definição das datas para aplicação de cada atividade, conforme o fluxo descrito na Figura 1.

¹ Disponível em: <<https://bit.ly/autoavapi>>.

Além disso, destaca-se a importância do diálogo com o docente responsável pelo componente, visto que a intervenção impacta diretamente o planejamento didático. Sugere-se a aplicar uma autoavaliação após cada conteúdo ministrado, sendo cada autoavaliação composta por um cenário.

Para auxiliar nesse planejamento, o Quadro 4 ilustra o conteúdo recomendado para as autoavaliações de cada unidade, baseado no conteúdo programático do componente curricular de Algoritmos do curso de BITI da UFRSA, Campus Pau dos Ferros. Essa recomendação é importante para padronizar o conteúdo cobrado em turmas diferentes.

Quadro 4 - Conteúdo recomendado para cada autoavaliação

UNIDADE	AUTOAVALIAÇÃO	CONTEÚDO
Primeira	1	Estrutura sequencial em pseudocódigo
	2	Estruturas de decisão em pseudocódigo
	3	Estruturas de repetição em pseudocódigo
Segunda	4	Estrutura sequencial em C
	5	Estruturas de decisão em C
	6	Estruturas de repetição em C
Terceira	7	Funções em C
	8	Vetores em C
	9	Matrizes em C

Fonte: Autoria própria (2025)

3.1.2 Aplicação do Questionário Social

Recomenda-se aplicar esse questionário na primeira semana de aula. Esse questionário tem o objetivo de identificar os perfis e as características predominantes das turmas. As informações obtidas permitem análises que consideram subgrupos formados a partir de variáveis como sexo, estilos de aprendizagem, relação entre ingressantes e repetentes, entre outras.

3.1.3 Explanação do Conteúdo

Momento em que o docente ministra o conteúdo, conforme o conteúdo programático definido no Quadro 4.

3.1.4 Elaboração da Autoavaliação

Cada autoavaliação é composta por um cenário. Para cada cenário, elaboram-se três dicas, seguindo a categorização definida no Quadro 2. Essas dicas visam orientar o aluno e identificar suas principais dificuldades.

3.1.5 Aplicação da Autoavaliação

Recomenda-se que cada autoavaliação seja aplicada após a conclusão do conteúdo correspondente e antes da introdução do próximo tema. Seu objetivo é permitir que os alunos identifiquem suas lacunas de conhecimento. Por exemplo, se um aluno utiliza a dica do tipo 2, infere-se que apresenta dificuldades com a sintaxe do conteúdo. Esse processo estimula a autorreflexão e permite ao discente buscar estratégias para superar suas dificuldades com o apoio do professor.

3.1.6 Sumarização dos Resultados da Autoavaliação

Após a aplicação de cada autoavaliação, são gerados dois tipos de artefatos:

- Para os discentes, são disponibilizados *feedbacks* personalizados que destacam tanto os acertos quanto os pontos que requerem maior atenção. Esses *feedbacks*, enviados por e-mail, podem incluir, sempre que necessário, sugestões de materiais complementares, como vídeos, livros ou conteúdos fornecidos pelo docente, com o objetivo de promover a revisão dos conceitos abordados e apoiar o desenvolvimento contínuo da aprendizagem.
- Para o docente, é elaborado um relatório com informações relevantes sobre o desempenho da turma. Esse documento contempla dados como: taxa de conclusão das autoavaliações, uso das dicas, principais erros observados e as percepções dos discentes em relação à autoavaliação, às aulas e às estratégias de estudo adotadas. O relatório é apresentado ao docente em uma reunião específica, em que são discutidos os resultados da turma. Essa análise conjunta permite compreender melhor os desafios enfrentados pelos alunos. A partir disso, o professor pode realizar intervenções pedagógicas mais eficazes e ajustar as estratégias de ensino, com o objetivo de aprimorar o processo de aprendizagem.

3.1.7 Aplicação do Questionário Final com os Discentes

Recomenda-se que esse questionário seja aplicado ao fim do experimento. Tal questionário serve para identificar quais foram as principais dificuldades dos alunos no decorrer do componente, quantas horas foram dedicadas aos estudos de programação, bem como quais foram os métodos utilizados para estudar. Essas informações possibilitam compreender e classificar as individualidades de aprendizagem, bem como confrontar os resultados obtidos pelos alunos com as metodologias de estudo e as horas dedicadas ao componente.

3.1.8 Aplicação do Questionário Final com os Docentes

Os professores que participaram do experimento também respondem a um questionário ao final do semestre. A partir dele, são identificadas as percepções docentes quanto às dificuldades da turma, comportamento dos alunos e os impactos positivos e/ou negativos da intervenção metodológica.

3.1.9 Sumarização Final dos Resultados para Análise

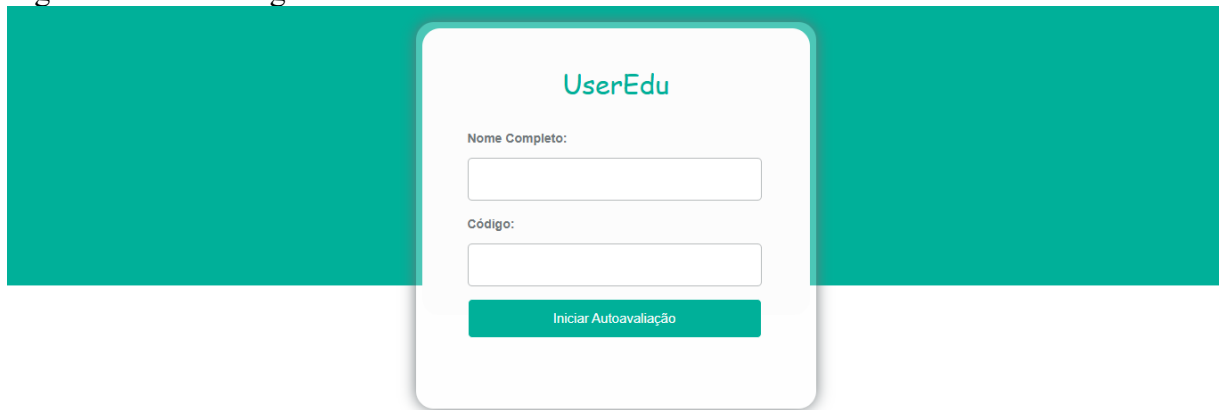
Os dados coletados por meio dos questionários finais são organizados e agrupados para possibilitar uma análise sistemática. A partir dessa estruturação, é possível avaliar o impacto da metodologia tanto sob a perspectiva dos alunos quanto dos professores, fornecendo subsídios para compreender a eficácia da intervenção.

3.2 Sistema de Suporte à Aplicação das Autoavaliações

Conforme mencionado na Seção 3.1, foi desenvolvido um sistema web com o objetivo de apoiar a aplicação das autoavaliações, permitindo estruturar a coleta e o armazenamento das respostas dos discentes de forma organizada e acessível. Embora ainda se encontre em fase inicial de desenvolvimento, o sistema já cumpre sua função principal: oferecer suporte à aplicação das autoavaliações.

Antes de responderem aos cenários, os discentes devem realizar um login simplificado, informando o nome completo e um código específico, previamente fornecido pelo professor. Ressalta-se que é gerado um código para cada nova autoavaliação. A Figura 2 ilustra a tela de login.

Figura 2 - Tela de login



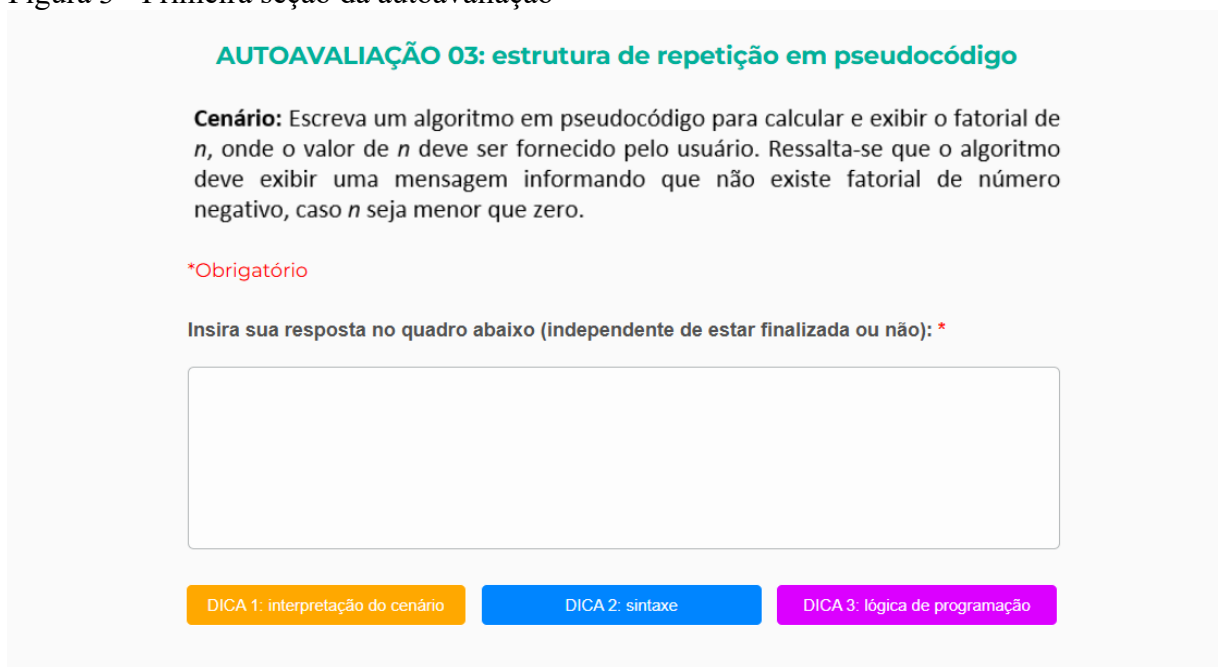
A tela de login do sistema UserEdu apresenta um formulário centralizado sobre um fundo verde-água. O formulário é branco com bordas arredondadas e contém o seguinte conteúdo:

- Logo "UserEdu" em verde.
- Rótulo "Nome Completo:" seguido de um campo de entrada de texto.
- Rótulo "Código:" seguido de um campo de entrada de texto.
- Botão "Iniciar Autoavaliação" em verde.

Fonte: Autoria própria (2025)

Ao acessar o sistema², o aluno é direcionado para uma interface única, na qual estão disponibilizadas todas as informações relacionadas à autoavaliação. Essa interface está organizada em três seções principais: (i) autoavaliação, que apresenta o cenário proposto e suas respectivas dicas; (ii) questionário sobre as dificuldades identificadas pelos discentes; e (iii) questionário sobre as aulas e as estratégias de estudo utilizadas. A Figura 3 ilustra a primeira seção da autoavaliação, correspondente ao cenário e às dicas associadas.

Figura 3 - Primeira seção da autoavaliação



A primeira seção da autoavaliação, intitulada "AUTOAVALIAÇÃO 03: estrutura de repetição em pseudocódigo", apresenta o seguinte conteúdo:

- Cenário:** Escreva um algoritmo em pseudocódigo para calcular e exibir o fatorial de n , onde o valor de n deve ser fornecido pelo usuário. Ressalta-se que o algoritmo deve exibir uma mensagem informando que não existe fatorial de número negativo, caso n seja menor que zero.
- *Obrigatório**
- Instrução: "Insira sua resposta no quadro abaixo (independente de estar finalizada ou não): *"
- Um grande campo de entrada de texto para a resposta.
- Três botões de dicas:
 - DICA 1: interpretação do cenário (laranja)
 - DICA 2: sintaxe (azul)
 - DICA 3: lógica de programação (roxo)

Fonte: Autoria própria (2025)

² Disponível em: <<https://intervencao metodologica.netlify.app/exibicaoartigo/teste>>.

Conforme ilustrado na Figura 3, a primeira seção da autoavaliação é composta pelo cenário, seguido de um campo destinado à inserção da resposta e três botões, cada um correspondente a uma das dicas, que podem ser acessadas conforme a necessidade do aluno durante a resolução do cenário. A Figura 4 exemplifica a visualização da Dica 1, exibida após ser acionada pelo discente.

Figura 4 - Exemplo de consulta à Dica 1

AUTOAVALIAÇÃO 03: estrutura de repetição em pseudocódigo

Cenário: Escreva um algoritmo em pseudocódigo para calcular e exibir o fatorial de n , onde o valor de n deve ser fornecido pelo usuário. Ressalta-se que o algoritmo deve exibir uma mensagem informando que não existe fatorial de número negativo, caso n seja menor que zero.

*Obrigatório

Insira sua resposta no quadro abaixo (independente de estar finalizada ou não): *

DICA 1: interpretação do cenário
DICA 2: sintaxe
DICA 3: lógica de programação

DICA 1:

Dados de Entrada: Um valor inteiro (n).

Dados de Saída: O fatorial de n (caso n seja maior ou igual a zero) ou a mensagem “Não existe fatorial de número negativo” (caso n seja menor que zero).

Processamento: Após receber o valor de n , deve-se verificar se n é negativo. Se for, deve-se exibir a mensagem “Não existe fatorial de número negativo”, senão, deve-se calcular e exibir o fatorial de n . Lembrando que o fatorial de n é obtido a partir da multiplicação de n por todos os seus antecessores até chegar em 1. Por exemplo: se $n = 3$, logo seu fatorial é: $3 * 2 * 1 = 6$.

Fonte: Autoria própria (2025)

A segunda seção da autoavaliação corresponde a um questionário voltado à identificação das principais dificuldades enfrentadas pelos discentes durante a resolução do cenário proposto.

Esse questionário inclui questões como: qual foi a principal dificuldade enfrentada (por exemplo, interpretação do cenário, sintaxe ou lógica de programação); quais os motivos que impediram a conclusão da resolução (como tempo insuficiente ou dificuldades técnicas); e uma avaliação subjetiva do nível de dificuldade do cenário (classificado como fácil, mediano ou difícil). A Figura 5 ilustra a estrutura e os elementos presentes nesse questionário.

Figura 5 - Segunda seção da autoavaliação

Sobre o Cenário

Qual foi sua principal dificuldade ao tentar resolver o cenário: *

☐ Dificuldade com a interpretação do cenário.

☐ Dificuldade com a sintaxe.

☐ Dificuldade com a lógica de programação.

☐ Outro: _____

Caso NÃO tenha conseguido resolver o cenário, indique o principal motivo: *

☐ Tempo Insuficiente.

☐ Dificuldade com a interpretação do cenário.

☐ Dificuldade com a sintaxe.

☐ Dificuldade com a lógica de programação.

☐ Outro: _____

Como você avaliaria o cenário? *

☐ Fácil.

☐ Mediana.

☐ Difícil.

Fonte: Autoria própria (2025)

Com base nos dados coletados pelo sistema, são gerados relatórios destinados aos docentes, que destacam as principais dificuldades enfrentadas pelos discentes. Esses relatórios são organizados segundo três abordagens: (i) dificuldades identificadas a partir das respostas submetidas pelos discentes; (ii) dificuldades relatadas diretamente pelos discentes na segunda seção da autoavaliação; e (iii) percepções dos alunos sobre as aulas e métodos de estudo, coletadas na terceira seção da autoavaliação.

A principal vantagem desses relatórios é possibilitar que os docentes planejem estratégias específicas para mitigar as dificuldades detectadas nas turmas, relacionadas aos conteúdos abordados na respectiva semana.

Por fim, a terceira seção da autoavaliação consiste em um questionário destinado a coletar informações sobre a percepção dos alunos em relação às aulas e às estratégias de estudo

adotadas. Essa etapa busca compreender aspectos como: se o estudante teve dúvidas durante a explicação do conteúdo, se considera necessária uma revisão do material, e se realizou algum tipo de preparação prévia antes da aula. Também são investigadas questões relacionadas à suficiência das aulas para a resolução do cenário proposto e à busca por métodos complementares de estudo, como vídeos, tutoriais ou outras fontes.

As respostas são apresentadas em uma escala simples — “Não”, “Talvez” e “Sim” —, o que permite uma análise objetiva das condições de aprendizagem. A Figura 6 ilustra a estrutura dessa última seção da autoavaliação.

Figura 6 - Terceira seção da autoavaliação

Sobre as Aulas e Formas de Estudo			
	Não.	Talvez.	Sim.
Você ficou com dúvida na explicação do conteúdo? *	☐	☐	☐
Você acha necessário revisar o conteúdo? *	☐	☐	☐
Estudou o conteúdo antes da aula? *	☐	☐	☐
Você acha que as aulas foram suficientes para responder o cenário? *	☐	☐	☐
Fora as aulas, você buscou outra forma de estudo para aprofundar os conteúdos vistos? *	☐	☐	☐

Fonte: Autoria própria (2025)

3.3 Aplicação do Estudo Experimental

Com o intuito de verificar a eficácia da proposta metodológica, foi realizado um estudo experimental no componente curricular de Algoritmos do curso de BITI da UFERSA, Campus Pau dos Ferros, no semestre remoto de 2021.1, compreendido no período de julho a novembro

de 2021. O número de discentes participantes variou ao longo das unidades: 32 na primeira, 24 na segunda e 16 na terceira.

Os dados coletados nas autoavaliações compreendem: (i) os algoritmos ou programas desenvolvidos pelos discentes, que permitem identificar os principais erros cometidos; (ii) o uso das dicas, que sinaliza as lacunas de conhecimento dos discentes; (iii) as dificuldades autorrelatadas pelos próprios estudantes; e (iv) as percepções dos alunos quanto às aulas e estratégias de estudo adotadas.

Essas informações possibilitam uma análise mais aprofundada do processo de aprendizagem e da eficácia da metodologia aplicada.

4 RESULTADOS E DISCUSSÃO

Os dados coletados foram organizados e analisados em três categorias para cada uma das três unidades: (i) número de dicas utilizadas em cada autoavaliação; (ii) dificuldades apontadas pelos alunos em cada autoavaliação; e (iii) relação entre as dicas utilizadas e as dificuldades indicadas pelos alunos em cada autoavaliação. Além disso, analisou-se os resultados referentes ao impacto da intervenção metodológica na percepção da turma e do professor.

É importante destacar que houve variação no número de participantes em relação ao número de respostas registradas, tanto quanto às dicas utilizadas quanto às dificuldades apontadas. Isso sugere que alguns alunos não selecionaram nenhuma das opções disponíveis, ou que assinalaram mais de uma alternativa.

As seções a seguir estão organizadas da seguinte forma: a Seção 4.1 apresenta os resultados obtidos na primeira unidade; a Seção 4.2 discute os resultados obtidos na segunda unidade; a Seção 4.3 apresenta os resultados obtidos na terceira unidade; e a Seção 4.4 discute o impacto da intervenção metodológica na percepção da turma e do professor.

4.1 Resultados da Primeira Unidade

Na primeira unidade, 32 (trinta e dois) alunos participaram do experimento. O Quadro 5 apresenta a quantidade de dicas utilizadas em cada autoavaliação da primeira unidade.

Quadro 5 - Utilização de dicas por autoavaliação na primeira unidade

AUTOAVALIAÇÃO	NÚMERO DE DICAS UTILIZADAS		
	DICA 1	DICA 2	DICA 3
1	5	8	7
2	8	6	8
3	8	8	8
TOTAL	21	22	23

Fonte: Autoria própria (2025)

Conforme apresentado no Quadro 5, observa-se que houve pouca variação no uso dos três tipos de dicas na primeira unidade. A Dica 3 foi a mais utilizada, o que sugere dificuldades relacionadas à lógica de programação. Em seguida, a Dica 2 apresentou frequência ligeiramente inferior, indicando dificuldades com a sintaxe. Por fim, a Dica 1 foi a menos utilizada, apontando dificuldades na interpretação do cenário proposto.

O Quadro 6 apresenta as dificuldades apontadas pelos alunos em cada uma das autoavaliações realizadas na primeira unidade.

Quadro 6 - Dificuldades apontadas pelos discentes na primeira unidade

AUTOAVALIAÇÃO	DIFICULDADES		
	INTERPRETAÇÃO DO CENÁRIO	SINTAXE	LÓGICA DE PROGRAMAÇÃO
1	2	6	10
2	3	3	9
3	2	2	10
TOTAL	7	11	29

Fonte: Autoria própria (2025)

A análise do Quadro 6 revela que a maioria dos discentes afirmou que teve dificuldade com a lógica de programação, seguida da dificuldade com a sintaxe e, por fim, a dificuldade com a interpretação do cenário foi a menos apontada.

Ao relacionar a quantidade de dicas de cada tipo utilizadas e as dificuldades apontadas pelos alunos, observa-se um alinhamento entre os dois conjuntos de dados. Em ambos os casos, a principal dificuldade evidenciada foi a lógica de programação, reforçando a consistência entre os indicadores diretos (dicas escolhidas) e as percepções autorrelatadas pelos alunos.

4.2 Resultados da Segunda Unidade

Na segunda unidade, 24 (vinte e quatro) alunos participaram do experimento. O Quadro 7 apresenta a quantidade de dicas utilizadas em cada autoavaliação da segunda unidade.

Quadro 7 - Utilização de dicas por autoavaliação na segunda unidade

AUTOAVALIAÇÃO	NÚMERO DE DICAS UTILIZADAS		
	DICA 1	DICA 2	DICA 3
4	3	4	5
5	1	5	4
6	1	4	3
TOTAL	4	13	12

Fonte: Autoria própria (2025)

A partir do Quadro 7, observa-se que, diferentemente da primeira unidade, a Dica 2 foi a mais utilizada, indicando dificuldades relacionadas à sintaxe. Em seguida, a Dica 3 foi a segunda mais recorrente, sugerindo dificuldades com a lógica de programação. Por fim, a Dica 1 apresentou menor frequência, sinalizando dificuldades na interpretação do cenário.

O Quadro 8 apresenta as dificuldades apontadas pelos alunos em cada autoavaliação da segunda unidade.

Quadro 8 - Dificuldades apontadas pelos discentes na segunda unidade

AUTOAVALIAÇÃO	DIFICULDADES		
	INTERPRETAÇÃO DO CENÁRIO	SINTAXE	LÓGICA DE PROGRAMAÇÃO
4	4	6	7
5	2	5	4
6	0	3	10
TOTAL	6	14	21

Fonte: Autoria própria (2025)

Com base nos dados apresentados no Quadro 8, observa-se que a maioria dos alunos relatou dificuldades com a lógica de programação, seguida da dificuldade com a sintaxe e, por último, dificuldade na interpretação do cenário.

Ao investigar a relação entre as quantidades de dicas de cada tipo utilizadas e as dificuldades apontadas pelos alunos, nota-se uma pequena divergência: enquanto o uso das dicas indicou a sintaxe como principal obstáculo, os discentes apontaram a lógica de programação como a maior dificuldade. No entanto, essa discrepância foi mínima, uma vez que a Dica 2 (sintaxe) foi utilizada por 13 alunos e a Dica 3 (lógica de programação) por 12 alunos, conforme apresentado no Quadro 7.

4.3 Resultados da Terceira Unidade

Na terceira unidade, apenas 16 (dezesseis) alunos participaram do experimento. O Quadro 9 apresenta a quantidade de dicas utilizadas em cada autoavaliação da terceira unidade.

Quadro 9 - Utilização de dicas por autoavaliação na terceira unidade

AUTOAVALIAÇÃO	NÚMERO DE DICAS UTILIZADAS		
	DICA 1	DICA 2	DICA 3
7	2	3	2
8	1	1	2
9	0	0	0
TOTAL	3	4	4

Fonte: Autoria própria (2025)

Conforme apresentado no Quadro 9, observa-se uma redução geral no uso de dicas, bem como pouca variação entre os três tipos utilizados na terceira unidade. A Dica 2 e a Dica 3

foram utilizadas com a mesma frequência, indicando dificuldades relacionadas à sintaxe e à lógica de programação. Já a Dica 1 foi a menos recorrente, sugerindo menor dificuldade na interpretação do cenário. O Quadro 10 apresenta as dificuldades apontadas pelos alunos em cada autoavaliação da terceira unidade.

Quadro 10 - Dificuldades apontadas pelos discentes na terceira unidade

AUTOAVALIAÇÃO	DIFICULDADES		
	INTERPRETAÇÃO DO CENÁRIO	SINTAXE	LÓGICA DE PROGRAMAÇÃO
7	0	5	5
8	0	4	3
9	0	4	2
TOTAL	0	13	11

Fonte: Autoria própria (2025)

Com base nos dados apresentados no Quadro 10, observa-se que a Dica 2 foi a mais utilizada, indicando dificuldades com a sintaxe, seguida pela Dica 3, que sinaliza dificuldades com a lógica de programação e, por fim, a Dica 1 não foi selecionada por nenhum aluno, o que sugere ausência de dificuldades relacionadas à interpretação do cenário nesta unidade.

Ao relacionar a quantidade de dicas utilizadas com as dificuldades apontadas, verifica-se um alinhamento entre os dados: as principais dificuldades da turma foram a sintaxe e a lógica de programação. Em consonância com o uso das dicas, 13 discentes indicaram a sintaxe como sua principal dificuldade, e 11 relataram dificuldades com a lógica de programação. Esses números indicam que alguns alunos apontaram simultaneamente dificuldades em ambas as áreas.

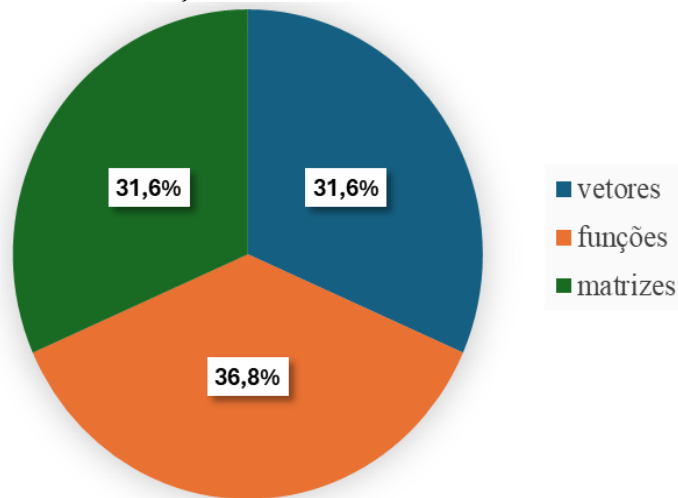
4.4 Resultados Sobre o Impacto da Intervenção Metodológica

Com o objetivo de avaliar o impacto da intervenção metodológica, foi aplicado um questionário ao final do experimento, obtendo-se um total de 19 respostas. As informações coletadas foram organizadas de acordo com as categorias previamente definidas para análise.

4.4.1 Conteúdos de Maior Dificuldade

Inicialmente, solicitou-se aos discentes que indicassem os conteúdos que apresentaram maior dificuldade durante as aulas, conforme o resultado ilustrado no Gráfico 1.

Gráfico 1 - Distribuição dos conteúdos de maior dificuldade



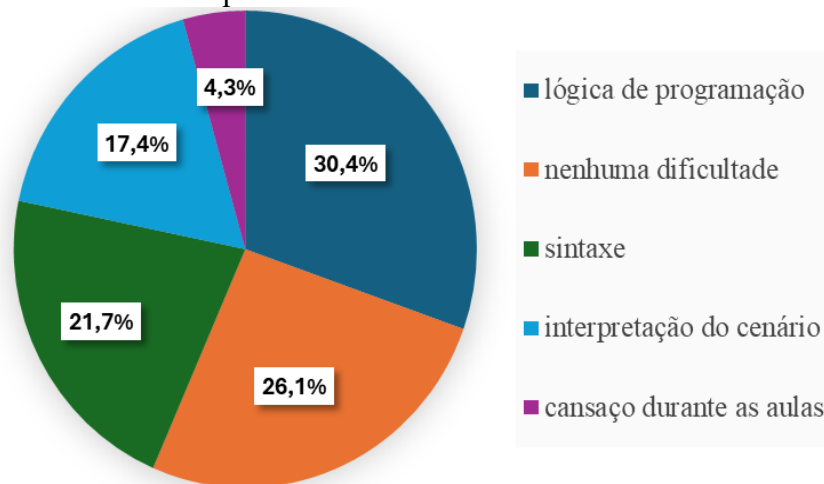
Fonte: Autoria própria (2025)

Os resultados demonstraram que 36,8% (7 alunos) apontaram dificuldades com funções, seguidos por 31,6% (6 alunos) que indicaram dificuldades com vetores e outros 31,6% (6 alunos) que mencionaram maior dificuldade com matrizes.

4.4.2 Principais Motivos de Dificuldades

Questionados sobre os motivos das dificuldades encontradas (com possibilidade de múltiplas respostas), os discentes forneceram diversas justificativas, conforme ilustrado no Gráfico 2.

Gráfico 2 - Principais fatores relacionados às dificuldades relatadas



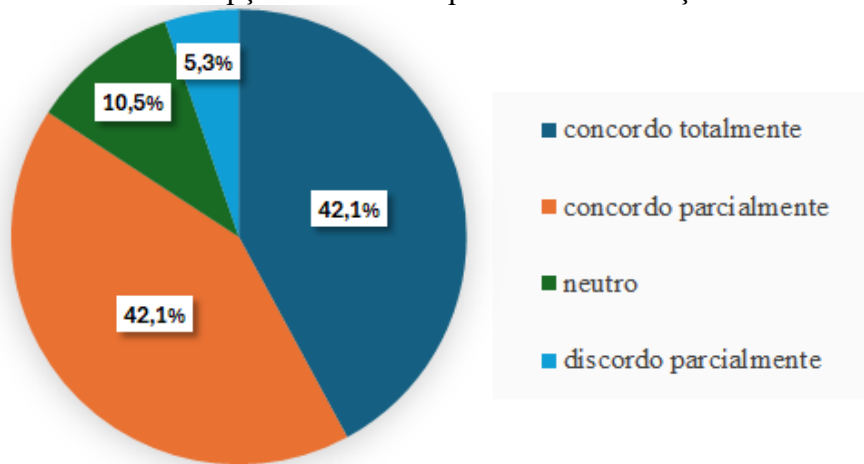
Fonte: Autoria própria (2025)

Os alunos apontaram a lógica de programação como o principal fator associado às dificuldades, sendo mencionada por 30,4% dos respondentes (7 alunos). Outros fatores relatados incluíram: ausência de dificuldades (26,1% – 6 alunos), sintaxe (21,7% – 5 alunos), interpretação do cenário (17,4% – 4 alunos) e cansaço durante as aulas (4,4% – 1 aluno).

4.4.3 Percepção Sobre as Autoavaliações

Os discentes também foram convidados a se posicionar em relação à seguinte afirmativa: “*As autoavaliações ajudaram a identificar minhas principais dificuldades.*”. O Gráfico 3 ilustra os resultados obtidos.

Gráfico 3 - Percepção dos alunos quanto à identificação das dificuldades

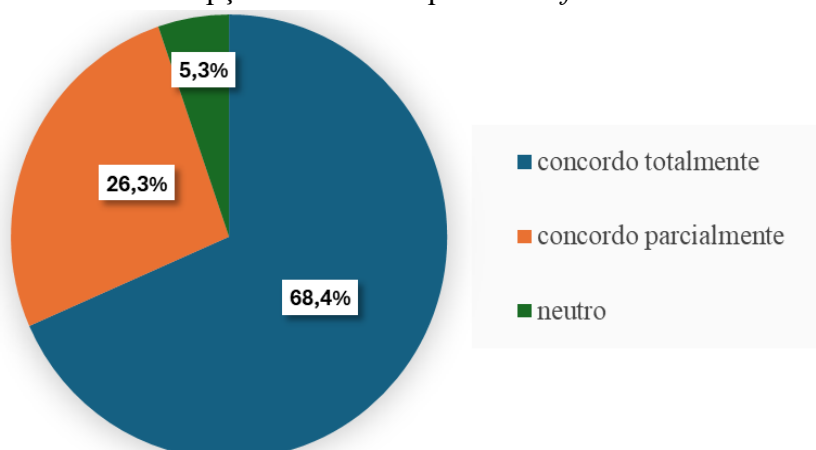


Fonte: Autoria própria (2025)

Dentre os participantes, 42,1% (8 alunos) afirmaram concordar totalmente que as autoavaliações contribuíram para a identificação de suas dificuldades, 42,1% (8 alunos) concordam parcialmente com a afirmativa. Por outro lado, 10,5% (2 alunos) declararam-se neutros, enquanto 5,3% (1 aluno) manifestou discordância parcial quanto à importância das autoavaliações.

4.4.4 *Feedbacks* das Autoavaliações

O Gráfico 4 apresenta os resultados obtidos em relação à afirmativa: “*Os feedbacks das autoavaliações foram úteis.*”.

Gráfico 4 - Percepção dos alunos quanto aos *feedbacks*

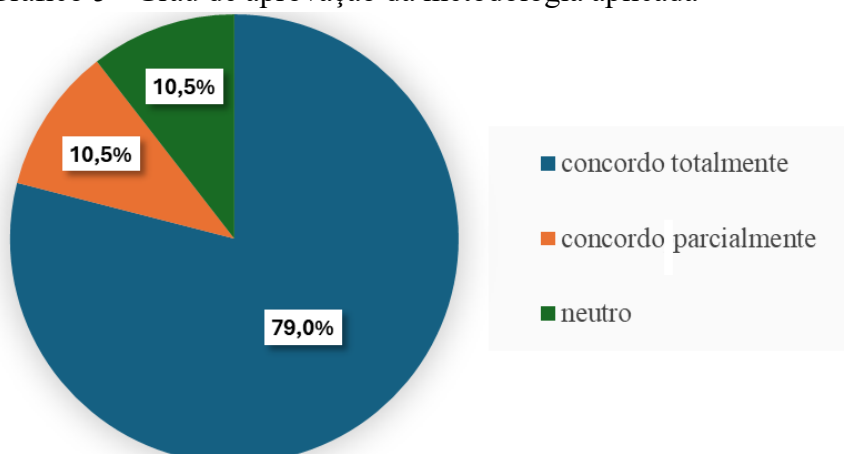
Fonte: Autoria própria (2025)

Os resultados foram amplamente positivos, indicando que 68,4% (13 alunos) concordaram totalmente que os *feedbacks* das autoavaliações foram úteis, enquanto 26,3% (5 alunos) concordam parcialmente com a afirmativa. Apenas 5,3% (1 aluno) se posicionou de forma neutra, ou seja, nem concordou nem discordou da afirmativa.

4.4.5 Aprovação da Metodologia Aplicada

O Gráfico 5 apresenta os resultados obtidos em relação à afirmativa: “*Aprovo e recomendo a aplicação da metodologia no próximo semestre.*”.

Gráfico 5 - Grau de aprovação da metodologia aplicada



Fonte: Autoria própria (2025)

Os resultados indicam que 79,0% (15 alunos) aprovam e recomendam totalmente a aplicação da metodologia no próximo semestre. Outros 10,5% (2 alunos) concordam

parcialmente com a afirmativa. Apenas 10,5% (2 alunos) se posicionaram de forma neutra, ou seja, nem concordaram nem discordaram.

4.4.6 Percepções dos Discentes sobre a Metodologia Aplicada

Para avaliar a percepção dos discentes, foi solicitado, de forma opcional, que comentassem se a metodologia contribuiu para o processo de aprendizagem. Ao todo, foram coletadas 11 respostas. A maioria das opiniões reforça a efetividade da proposta, evidenciando que os estudantes perceberam benefícios concretos em seu aprendizado. Alguns dos comentários registrados incluem:

“A metodologia me ajudou muito, pois com ela eu pude praticar os conteúdos e observar onde eu estava com dificuldade. A metodologia contribuiu bastante para o meu aprendizado.” (Discente 1)

“Contribuiu bastante, pois com o retorno da autoavaliação dava pra refazer e entender de certa forma os erros.” (Discente 2)

“A metodologia foi muito útil para o aprendizado, e ajudava muito no caso de alguma dúvida.” (Discente 3)

“Foi bastante proveitosa, visto que, era sempre frisada a importância da prática após o ensino dos assuntos.” (Discente 4)

“Ajuda bastante a compreender a linguagem e o desenvolvimento do programa através das dicas e de qual sintaxe usar no programa.” (Discente 5)

As respostas refletem uma percepção predominantemente positiva em relação à metodologia adotada. Os estudantes destacaram aspectos como a possibilidade de praticar os conteúdos, identificar suas dificuldades, receber *feedback* e compreender melhor os temas abordados.

A análise qualitativa dessas opiniões reforça a relevância de metodologias que incentivem a prática, o acompanhamento contínuo e a autorreflexão no processo de aprendizagem, especialmente em componentes tradicionalmente marcados por altos índices de evasão e reprovação.

4.4.7 Percepção do Docente sobre a Metodologia Aplicada

Além da avaliação realizada pelos discentes, também foi considerada a percepção do docente responsável pelo componente. O professor destacou a eficácia da metodologia aplicada,

especialmente, por contemplar um aspecto fundamental do processo de ensino-aprendizagem: a autoavaliação. Segundo ele, “a intervenção metodológica permite ao docente identificar as principais dificuldades dos discentes ao longo do semestre e, por conseguinte, intervir para minimizá-las”.

De acordo com o professor, a metodologia possibilitou uma personalização mais eficaz do ensino, promovendo o acompanhamento contínuo do desempenho dos estudantes. Essa abordagem facilitou intervenções pedagógicas direcionadas, com o objetivo de mitigar, de forma mais assertiva, as dificuldades enfrentadas por cada aluno.

Por fim, o docente afirmou aprovar e recomendar a continuidade da aplicação dessa metodologia em semestres futuros, reforçando seu potencial como estratégia de apoio ao ensino de programação introdutória.

5 CONSIDERAÇÕES FINAIS

Este trabalho apresentou uma proposta metodológica fundamentada na autoavaliação, com o objetivo de apoiar o processo de ensino-aprendizagem do componente curricular de Algoritmos, no curso de BITI da UFERSA, Campus Pau dos Ferros. A intervenção metodológica teve como principais objetivos estimular a autonomia dos discentes, promover a autorreflexão sobre a própria aprendizagem e fornecer aos docentes subsídios para intervenções pedagógicas mais direcionadas.

A análise dos dados revelou uma relação entre o tipo de dica utilizada e as dificuldades relatadas pelos discentes, especialmente, no que se refere à lógica de programação e à sintaxe. Essa relação evidencia que os alunos foram capazes de reconhecer suas principais dificuldades, o que representa um avanço significativo no desenvolvimento da autorregulação da aprendizagem. Além disso, a adoção de cenários contextualizados contribuiu para uma aproximação mais efetiva entre teoria e prática, favorecendo uma aprendizagem mais significativa.

Com base nos resultados obtidos, é possível afirmar que a aplicação de autoavaliação pode, sim, auxiliar de maneira significativa no processo de ensino-aprendizagem de programação introdutória. A metodologia adotada favoreceu a construção de um processo avaliativo mais reflexivo e participativo, promovendo a consciência crítica dos discentes sobre sua trajetória de aprendizagem. A proposta foi bem recebida pelos discentes, com 79,0% concordando totalmente com sua aplicação em semestres futuros, e 10,5% concordando parcialmente, totalizando 89,5% de aprovação, além da avaliação positiva do docente responsável.

Contudo, reconhece-se que a metodologia, por si só, não soluciona integralmente os desafios enfrentados no processo de ensino-aprendizagem de programação introdutória. Ainda assim, trata-se de um recurso relevante no aprimoramento desse processo, especialmente por sua capacidade de promover a autonomia discente e fornecer dados úteis ao professor para uma atuação pedagógica mais eficaz.

5.1 Limitações do Estudo

Apesar dos resultados positivos, este estudo apresenta algumas limitações que devem ser consideradas. Primeiramente, a intervenção foi aplicada em um único componente curricular (Algoritmos) e em apenas uma instituição (UFERSA – Campus Pau dos Ferros), o

que restringe a generalização dos resultados para outros contextos. Além disso, o número de participantes diminuiu ao longo do semestre, o que pode ter afetado a consistência dos dados nas unidades subsequentes.

Outra limitação diz respeito à fase inicial de desenvolvimento do sistema de suporte às autoavaliações, que, embora funcional, ainda carece de aprimoramentos. Por fim, fatores externos ao ambiente de sala de aula, como acesso limitado à internet, tempo disponível para estudos e fatores emocionais, não foram profundamente investigados, embora possam impactar significativamente o desempenho discente.

5.2 Trabalhos Futuros

Como desdobramento deste estudo, propõe-se a realização de investigações em outros contextos educacionais e em componentes distintos da área de computação, com o intuito de avaliar a eficácia da metodologia proposta em cenários variados. Além disso, sugere-se a ampliação da base de dados com maior número de participantes, o que poderá fortalecer as evidências obtidas e permitir análises estatísticas mais robustas.

Também se recomenda o aperfeiçoamento do sistema de suporte, incorporando funcionalidades como *dashboards* interativos e recomendações automáticas baseadas no desempenho do aluno. Outra frente relevante consiste na análise longitudinal do impacto da metodologia na trajetória acadêmica dos estudantes, investigando, por exemplo, sua relação com o desempenho em componentes curriculares subsequentes.

Essas futuras iniciativas poderão contribuir para a consolidação de práticas pedagógicas mais eficazes, responsivas às necessidades dos alunos e sensíveis às especificidades do ensino de programação introdutória.

REFERÊNCIAS

- AMARAL, É. **Processo de Ensino-Aprendizagem de algoritmos integrando ambientes imersivos e o paradigma de blocos de programação visual**. 2015. Tese (Doutorado em Informática na Educação) - Universidade Federal do Rio Grande do Sul, 2015.
- BARCELOS, T. S.; SILVEIRA, I. F. Pensamento computacional e educação matemática: relações para o ensino de computação na educação básica. *In: Anais do XX Workshop Sobre Educação em Computação (WEI)*, Curitiba v. 20, 2012,
- BERSSANETTE, J. H.; FRENCISCO, A. C. Proposta de abordagem prática para o ensino de programação baseada em Ausubel. *In: Anais do XXIX Simpósio Brasileiro de Informática na Educação (SBIE)*, Fortaleza, v. 29, n. 1, 2018.
- BORGES, M. A. F. Avaliação de uma metodologia alternativa para a aprendizagem de programação. *In: VIII Workshop de Educação em Computação (WEI)*, Curitiba, v. 3, n. 1, 2000.
- BOTH, I. J.; BRANDALISE, Â. M. T. Interferência pedagógica na avaliação da aprendizagem no ensino presencial e a distância com o uso de tecnologias. **Ibero-Americana de Estudos em Educação**, Araraquara, v. 13, n. 3, p. 807–821, 2018.
- CAPELLATO, P.; VASCONCELOS, L. V. B.; RANIERI, M. G. A.; SACHS, D. Peer and self-evaluation using active teaching method. **Research, Society and Development**, [s. l.], v. 9, n. 7, p. e21973495, 2020.
- CHAGAS, L. B. C.; OLIVEIRA, M. G.; TAVARES, O. L. APP- Uma arquitetura pedagógica para aprendizagem de programação de computadores. **Revista Eletrônica Argentina-Brasil de Tecnologias da Informação e da Comunicação**, [s.l.], v. 1, n. 5, 2016.
- CHENG, G.; Zou D.; Xie H.; Wang, F. L. Exploring differences in self-regulated learning strategy use between high- and low-performing students in introductory programming: An analysis of eye-tracking and retrospective think-aloud data from program comprehension. **ScienceDirect Computers & Education**, [s.l.], v. 208, p. 1-18, 2023.
- COŞKUNSERÇE, O. Comparing the use of block-based and robot programming in introductory programming education: Effects on perceptions of programming self-efficacy. **Computer Applications in Engineering Education**. [s.l.], v. 5, p. 1234–1255, 2023.
- FALKNER, K.; VIVIAN, R.; FALKNER, N. J. G. Identifying computer science self-regulated learning strategies. *In: Conference on Innovation and Technology in Computer Science Education*. [s.l.], v. 3, p. 291-296, 2014.
- FARIAS, M. A. F.; JÚNIOR, G. P. S.; ANDRADE, E. M. A. O. R. Um estudo preliminar sobre as dificuldades no processo de ensino-aprendizagem das disciplinas básicas de programação no ifs-Campus Lagarto. *In: IV Colóquio Internacional “Educação e Contemporaneidade”*, São Cristóvão, v. 1, p. 1-11, 2012.

FRANÇA, R. S.; TEDESCO, P. C. Caracterizando a pesquisa sobre autoavaliação na aprendizagem de programação para iniciantes. *In: XXVI Anais do Simpósio Brasileiro de Informática na Educação (SBIE)*, [s.l.], v. 26, n. 1, p. 549, 2015.

FIGUEIREDO, J.; GARCÍA-PEÑALVO, F. J. Building skills in introductory programming. *In: Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality*, [s.l.], v. 6, p. 46-50, 2018.

FREITAS, B. C. B. **Uma intervenção metodológica para auxiliar a aprendizagem de programação introdutória da UFERSA, campus Pau dos Ferros**. 2020. Monografia (Graduação em Bacharelado em Tecnologia da Informação), Universidade Federal Rural do Semi-Árido, Pau dos Ferros, 2020.

GARCÍA-PEÑALVO, F. J. Computational thinking and programming education principles. *In: International Conference on Technological Ecosystems for Enhancing Multiculturality*, Nova York, v. 6, p. 14–17, 2018.

GERHARD, W.; BRUSILOVSKY, P. Elm-art: An adaptive versatile system for web based instruction. *In: International Journal of Artificial Intelligence in Education*, [s.l.], v. 1, p. 351-384, 2001.

GIL, A. C. **Métodos e técnicas de pesquisa social**. 6. ed. São Paulo: Atlas, 2008.

GOSSMANN, T. A. **Utilização de algoritmos nos processos de ensino e de aprendizagem de programação de computadores em cursos técnicos**. 2017. Artigo (Especialização) – Curso de Docência na Educação Profissional, Universidade do Vale do Taquari - Univates, Lajeado, 2017.

GRAAFSMA, I.; ROBIDOUX, S.; NICKELS, L; ROBERTS, M.; POLITO, V.; D. ZHU, J.; MARINUS, E. The cognition of programming: logical reasoning, algebra and vocabulary skills predict programming performance following an introductory computing course. *In: Journal of Cognitive Psychology*. v. 3, p. 364–381, 2023.

HOLANDA, W. D.; COUTINHO, J. C. World Prog: Um jogo educacional para aprendizagem de conceitos básicos de programação. **Revista Novas Tecnologias na Educação (RENOTE)**, Porto Alegre, v. 20, n. 1, p. 213–222, 2022.

HOLANDA, W. D.; COUTINHO, J. C. S.; FONTES, L. M. O. Uma intervenção metodológica para auxiliar a aprendizagem de programação introdutória: um estudo experimental. *In: Anais dos Workshops do VII Congresso Brasileiro de Informática na Educação (CBIE 2018)*, Fortaleza, v. 7, p. 699-708, 2018.

LIMA, J.; MENEZES, C. As dificuldades enfrentadas pelos estudantes na aprendizagem de programação de computadores: Uma Revisão Sistemática da Literatura. **Revista Novas Tecnologias na Educação (RENOTE)**, Porto Alegre, v. 22, n. 1, p. 130-140, 2024.

LOPES, N. F. L. A prática da autoavaliação no ensino superior. **Revista de psicologia**, [s. l.], v. 12, n. 39, p. 839–850, 2018.

LUXTON-REILLY, A.; ALBLUWI, I.; BECKER, B. A.; GIANNAKOS, M.; KUMAR, A. N.; OTT, LINDA.; PATERSON, J.; SCOTT, M. J.; SHEARD, J.; SZABO, C. Introductory programming: a systematic literature review. *In: conference on innovation and technology in computer science education*, New York, v. 23, p. 55-106, 2018.

MARASCO, E. A.; MOSHIRPOUR, M.; MOUSSAVI, M. Flipping the foundation: A multi-year flipped classroom study for a large-scale introductory programming course. *In: Annual Conference & Exposition*, Ohio, USA, 2017.

MEDEIROS R. P.; RAMALHO G. L.; FALCÃO T. P. A systematic literature review on teaching and learning introductory programming in higher education. *In: IEEE Transactions on Education*, [s.l.], v. 62, n. 2, p. 77-90, 2019.

MOREIRA, G. L.; HOLANDA, W.; COUTINHO, J. C. S.; CHAGAS, F. S. Desafios na aprendizagem de programação introdutória em cursos de TI da UFERSA, campus Pau dos Ferros: um estudo exploratório. *In: Anais do III Encontro de Computação do Oeste Potiguar (ECOP)*, Pau dos Ferros, v. 1, n. 2, 2018.

NASCIMENTO, P. O. Concepções e práticas da autoavaliação na Educação Profissional e Tecnológica: o caso do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas – Ifam (2012/2023). *Horizontes*, [s.l.], v. 42, n. 1, 2024.

NOURI, J.; ZHANG, L.; MANNILA, L.; NORÉN, E. Development of computational thinking, digital competence and 21st century skills when learning programming. *In: K-9. Education Inquiry*, [s.l.], v. 11, n. 1, p. 1-17, 2019.

PETRY, P. G. **Um sistema para o Ensino-Aprendizagem de algoritmos utilizando um companheiro de aprendizagem colaborativo**. 2005. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Santa Catarina, Florianópolis, 2005.

PIMENTEL, E. P.; FRANÇA, V. F.; NORONHA, R. V.; OMAR, N. Avaliação contínua da aprendizagem, das competências e habilidades em programação de computadores. *In: IX Workshop de informática na Escola - WIE*, São Paulo, v. 2, p. 533-544, 2003.

SANTIAGO, A. D.; KRONBAUER, A. Um modelo lúdico para o ensino de conceitos de programação de computadores. *In: Simpósio Brasileiro de Informática na Educação (SBIE)*, [S.l.], v. 25, p. 1-29, 2016.

SANTOS, L. Auto-avaliação regulada: porquê, o quê e como? *In: ABRANTES, P.; ARAÚJO, F (Org.). Avaliação das aprendizagens, das concepções às práticas*. Ministério da Educação e Departamento da Educação Básica, Lisboa, p. 75 – 84, 2014.

SIROTHEAU, S.; BRITO, S. R.; SILVA, A. S.; ELIASQUEVICI, M. K.; FAVERO, E. L.; TAVARES, O. L. O. Aprendizagem de iniciantes em algoritmos e programação: foco nas competências de autoavaliação. *In: Anais do XXII SBIE - XVII WIE*, [s.l.], v. 1, p. 750-759, 2011.

SILVA, B. G. S.; FONTES, L. M. de O.; FREITAS, B. C. de B. Autoavaliação na aprendizagem de programação introdutória: uma revisão sistemática da literatura / Self-assessment in

introductory programming learning: a systematic literature review. **Brazilian Journal of Development**, [s.l.], v. 8, n. 5, p. 32616–32643, 2022.

SILVA, M. F. C. M. **Aprender a aprender: O papel da autoavaliação no processo de ensino-aprendizagem**. 2013. Dissertação (mestrado em Humanidades) - Universidade do Porto. [s.l.], 2013.

SILVA, T. R.; MEDEIROS, T.; MEDEIROS, H.; LOPES, R.; ARANHA, E. Ensino aprendizagem de programação: uma revisão sistemática da literatura. *In: Revista Brasileira de Informática na Educação (SBIE)*, [s.l.], v. 23, n. 1, 2015.

SOUZA, D. M.; BATISTA, M. H. S.; BARBOSA, E. F. B. Problemas e dificuldades no ensino e na aprendizagem de programação: um mapeamento sistemático. *In: Revista Brasileira de Informática na Educação*, [s.l.], v. 24, p. 39-52, 2016.

SRIVATANAKUL, T. Emerging from the pandemic: instructor reflections and students' perceptions on an introductory programming course in blended learning. *In: Springer Science*, [s.l.], v. 28, p. 5673–5695, 2022.

TAROUCO, L. M. R.; SANTOS, P. M. E.; ÁVILA, B. GRANDO, A. ABREU, C. S. Multimídia interativa: princípios e ferramentas. *In: Revista Novas Tecnologias na Educação (RENOTE)*, Porto Alegre, v. 7, n. 1, 2009.

VAHLICK, A.; SCHOEFFEL, P.; LIZ, F. B.; WAZLAWICK, R.; RAMOS, V. Maior frequência na aplicação de instrumentos de avaliação em uma disciplina introdutória de programação: impactos no desempenho e motivação. *In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO*, Fortaleza/CE. v. 7, p. 739-748, 2018.

VIEIRA, I. M. A. **A autoavaliação como instrumento de regulação da aprendizagem**. 2013. Dissertação (Mestrado em Supervisão Pedagógica) - Universidade Aberta, Lisboa. p. 161, 2013.