REVISTA DE TECNOLOGIA APLICADA (RTA)
Vol. 15, No. 1, Jan-Abr 2026, p.01-12
ISSN: 2237-3713

Pensamento computacional na formação em engenharia de software*

Computational thinking in software engineering education

Marco Abraão Machado De Menezes Junior
Universidade de Vassouras
marcoabjr1994@gmail.com

Fabrício Tadeu Dias
Universidade Federal Fluminense
fabricio_dias@id.uff.br

RESUMO

O pensamento computacional constitui uma competência relevante para a formação em tecnologia, especialmente no desenvolvimento de habilidades associadas à resolução de problemas e à lógica de programação. Este estudo teve como objetivo analisar a percepção de estudantes de graduação em Engenharia de Software sobre as contribuições dessa competência para sua formação acadêmica. Trata-se de uma pesquisa descritiva, de abordagem qualitativa, fundamentada em levantamento bibliográfico e aplicação de questionário composto por seis afirmações em escala Likert de cinco pontos, respondido por 29 estudantes do primeiro ao oitavo período da Universidade de Vassouras, campus Maricá. Os dados foram analisados com auxílio da linguagem Python. Os resultados revelaram que 93,34% dos participantes perceberam contribuições para a resolução de problemas, 96,67% para a lógica de programação e a totalidade reconheceu a importância da inclusão da disciplina na formação. Conclui-se que o pensamento computacional é valorizado pelos estudantes, embora as percepções autorrelatadas não permitam inferir causalidade ou ganhos cognitivos efetivamente mensurados.

Palavras-chave: Pensamento computacional. Ensino Superior. Resolução de Problemas. Lógica de programação. Formação por competências.

ABSTRACT

Computational thinking is widely recognized as an essential competency in technology education, particularly for developing problem-solving skills and programming logic. This study aimed to examine undergraduate Software Engineering students' perceptions of its contribution to their academic development. A descriptive study with a qualitative approach was conducted, combining a literature review with a questionnaire comprising six statements rated on a five-point Likert scale. The instrument was administered to 29 students from the first to the eighth semester at the University of Vassouras, Maricá campus. Data were analyzed using Python. The findings showed that 93.34% of respondents perceived improvements in problem-solving skills, 96.67% acknowledged its contribution to programming logic, and all participants considered the inclusion of computational thinking as a subject in the curriculum important. These findings indicate that students recognize the educational value of computational thinking,

* Recebido em 27 de dezembro de 2025, aprovado em 23 de março de 2026, publicado em 30 de abril de 2026.

although self-reported perceptions do not establish causal relationships or demonstrate objectively measured cognitive gains.

Keywords: Computational thinking. Higher education. Problem-solving. Programming logic. Competency-based education.

1. INTRODUÇÃO

As tecnologias digitais mudaram a forma como pessoas e organizações acessam informações, tomam decisões e resolvem problemas, e essa mudança chega à formação oferecida pelos cursos superiores da área tecnológica. Nesse cenário, o pensamento computacional é discutido como uma competência que reúne estratégias sistemáticas de análise, organização e resolução de problemas, aplicáveis à computação e a outros campos do conhecimento. Wing (2008) o descreve como uma habilidade analítica que não deve ficar restrita a profissionais da computação. A discussão tem também uma dimensão social. Barboza e Winter (2026), ao tratarem de indicadores de ciência, tecnologia e inovação diante da vulnerabilidade social, relacionam o acesso aos benefícios do conhecimento científico e tecnológico à capacitação digital da população, o que situa a formação em competências tecnológicas entre as condições de participação na sociedade digital.

A relação entre sistemas computacionais e processos cognitivos humanos aparece, por outro ângulo, no estudo de Melo et al. (2026), que propõem uma arquitetura algorítmica de recomendação voltada à preservação da atenção e a testam por simulação com agentes virtuais. O estudo não trata do ensino de pensamento computacional nem de estudantes de Engenharia de Software, mas mostra, em ambiente simulado, que escolhas de projeto algorítmico alteram a reserva cognitiva dos agentes. A evidência é limitada, e ainda assim torna pertinente que a formação superior em tecnologia inclua a compreensão de como soluções computacionais são estruturadas, já que quem projeta sistemas decide, em parte, as condições em que outras pessoas os utilizam.

No campo educacional, o construcionismo de Papert (1980) sustenta que o uso de computadores para construir artefatos pode favorecer a aprendizagem, e Blikstein (2008) associa o pensamento computacional à ampliação das capacidades cognitivas e operacionais humanas. A produção brasileira mobilizada neste trabalho concentra-se em boa parte na educação básica e no ensino de programação. Brackmann (2017) investiga o desenvolvimento do pensamento computacional por atividades desplugadas na educação básica, Conforto et al. (2018) o discutem como competência do século XXI nesse nível de ensino, e França e Tedesco (2015), Garcia et al. (2008) e Zanetti et al. (2016) tratam, respectivamente, de desafios e oportunidades de seu ensino, do ensino de lógica de programação no ensino médio e do pensamento computacional no ensino de programação, nesse último caso em revisão sistemática da literatura brasileira.

Os estudos mobilizados oferecem uma base conceitual consistente, mas deixam em aberto uma questão própria do ensino superior: como estudantes de graduação em Engenharia de Software, cuja formação envolve análise, modelagem e implementação de sistemas, percebem a contribuição do pensamento computacional para a própria trajetória. Conhecer essa

percepção importa porque a presença de tecnologias no ambiente educacional não garante, por si só, o desenvolvimento das competências esperadas, e porque a avaliação que os estudantes fazem do que aprendem orienta decisões sobre currículo e sobre a oferta de disciplinas. Percepção, contudo, não equivale a desempenho. Este estudo trata de percepções autorrelatadas e não mede desenvolvimento cognitivo.

Para abordar a questão, realizou-se uma pesquisa descritiva, de abordagem qualitativa, fundamentada em levantamento bibliográfico e complementada por questionário estruturado aplicado a 29 estudantes do primeiro ao oitavo período do curso de Engenharia de Software da Universidade de Vassouras, campus Maricá. O instrumento reuniu seis afirmações avaliadas em escala Likert de cinco pontos, e as respostas foram organizadas e analisadas com a linguagem Python. O objetivo foi analisar a percepção desses estudantes sobre a contribuição do pensamento computacional para o desenvolvimento de habilidades relacionadas à resolução de problemas e à lógica de programação. Especificamente, buscou-se examinar como avaliam a importância da competência em sua formação, sua relação com a lógica de programação e sua aplicabilidade em outras disciplinas, em situações cotidianas e em etapas anteriores da educação formal.

2. FUNDAMENTAÇÃO TEÓRICA

Esta seção organiza a literatura em cinco eixos: a origem e o conceito do pensamento computacional; seus pilares e as distinções necessárias em relação a noções vizinhas; sua relação com cognição e resolução de problemas; o construcionismo; e as aplicações e dimensões formativas que interessam à Engenharia de Software.

2.1 Origem e conceito do pensamento computacional

O termo pensamento computacional ganhou ampla circulação com Wing (2006), que propôs somá-lo à leitura, à escrita e à aritmética entre as habilidades analíticas de toda criança. Em texto posterior, a autora o associou ao poder e aos limites dos processos de computação, executados por pessoas ou por máquinas, e o apresentou como habilidade de todos, e não apenas de profissionais da computação (Wing, 2008; Wing, 2021).

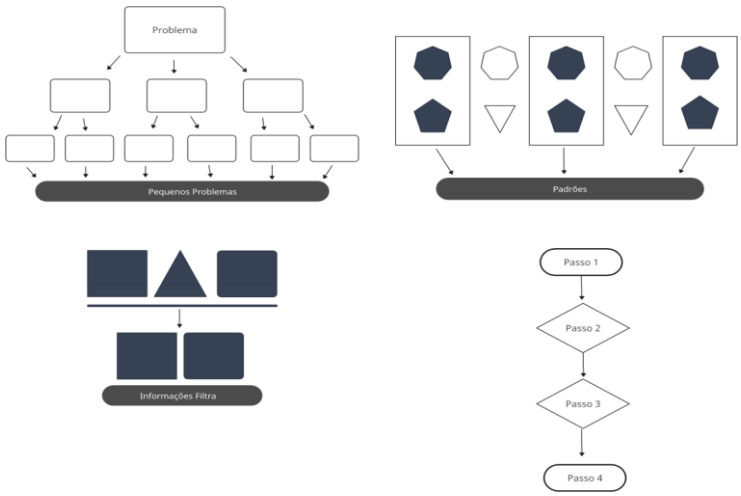
A expressão, porém, é anterior a esse texto. Papert (1980) a utiliza ao discutir como crianças que programam computadores aprendem a pensar sobre problemas e sobre o próprio pensamento, em obra que também introduz as noções de ideias poderosas e de conhecimento procedural. Denning (2017) examina a trajetória do conceito e discute pontos que permanecem sem consenso em sua definição, o que recomenda cautela ao tratá-lo como noção estabilizada.

Blikstein (2008) distingue o pensamento computacional do simples uso da internet e de suas ferramentas e o define como o uso do computador para ampliar o poder cognitivo e operacional humano, em favor da produtividade, da inventividade e da criatividade. Para o autor, pensar computacionalmente envolve identificar as tarefas cognitivas que um computador executa de modo mais rápido e eficiente e, em seguida, saber programá-lo para realizá-las. Blikstein observa ainda que cientistas, engenheiros e economistas passaram a dedicar mais tempo ao computador no exercício de suas funções, sinal de transformações na academia e na indústria.

2.2 Pilares e distinções conceituais

Brackmann (2017) descreve o pensamento computacional como um conjunto de habilidades e conhecimentos de computação que torna mais simples identificar e resolver problemas, de modo individual ou colaborativo, por meio de passos que uma pessoa ou um computador possa executar corretamente, e o organiza em quatro pilares. A decomposição divide um problema em partes menores, o que facilita sua resolução e seu gerenciamento. O reconhecimento de padrões identifica regularidades entre problemas, ou entre partes de um mesmo problema, de modo que soluções conhecidas possam ser reaproveitadas. A abstração seleciona os elementos essenciais de uma situação e ignora detalhes irrelevantes, o que exige decidir o que descartar sem perder informação importante. O algoritmo é uma sequência finita e ordenada de passos, cada um executável em tempo finito por um agente humano ou computacional, para resolver um problema ou realizar uma tarefa. A Figura 1 representa esquematicamente esses pilares.

Figura 1 - **Representação esquemática dos pilares do pensamento computacional**[†]



Fonte: Os autores.

Os pilares delimitam o pensamento computacional, mas não o tornam equivalente a noções vizinhas, e a distinção orienta a leitura dos dados deste estudo. Valente (2019) discute a diferença entre pensamento computacional, letramento computacional e competência digital, o que reforça a necessidade de empregar os termos com precisão. Neste trabalho, o pensamento computacional é entendido como um modo de abordar problemas por decomposição, padrões, abstração e algoritmos. A programação de computadores é uma das atividades em que esse modo de pensar pode ser exercitado, mas não a única, e pode ser praticada com pouca reflexão sobre esses processos. O raciocínio lógico é a capacidade de encadear premissas e conclusões, e está presente em contextos que não envolvem computação. O desenvolvimento cognitivo refere-se a mudanças duradouras nas capacidades de pensar de uma pessoa, e sua constatação exige delineamentos que acompanhem essas mudanças. A aprendizagem designa a aquisição

[†] Nota. Decomposição (superior esquerdo), reconhecimento de padrões (superior direito), abstração (inferior esquerdo) e algoritmo (inferior direito).

de conhecimentos e habilidades, e a competência tecnológica, a capacidade de mobilizá-los no uso e na criação de soluções tecnológicas.

Esses conceitos se relacionam, mas não são sinônimos. Atribuir à programação todas as competências cognitivas que a literatura associa ao pensamento computacional seria uma generalização que os trabalhos mobilizados aqui não autorizam, e que os dados desta pesquisa também não sustentam.

2.3 Pensamento computacional, cognição e resolução de problemas

Parte da literatura associa o pensamento computacional ao desenvolvimento de competências cognitivas e de resolução de problemas. Conforto et al. (2018) sustentam que sua inclusão na educação básica favorece a autonomia dos estudantes ao resolver problemas e a capacidade de descrever e explicar situações complexas, e que estudantes com acesso a essa ferramenta cognitiva tendem a analisar dados logicamente, representá-los de forma abstrata e tratar problemas complexos por meio de variáveis e estruturas condicionais. De Carvalho et al. (2013), ao tratarem do ensino médio, associam a introdução do pensamento computacional ao planejamento, ao aprendizado e ao raciocínio lógico.

Os trabalhos citados dizem respeito à educação básica. Outros estudos examinam o tema sob a perspectiva da programação, como a análise dos efeitos do pensamento computacional em habilidades de estudantes do ensino básico (Rodrigues et al., 2015) e a revisão sistemática da produção brasileira sobre seu uso no ensino de programação (Zanetti et al., 2016). Por terem populações e delineamentos distintos do adotado aqui, esses estudos servem de referência conceitual para a discussão, mas seus resultados não podem ser tomados como confirmação dos achados desta pesquisa.

A cognição interessa à computação também por outra via, a do efeito dos sistemas sobre quem os usa. Melo et al. (2026) propõem o modelo Be-Productive, uma arquitetura de recomendação que busca preservar a atenção e a reserva cognitiva dos usuários, e o avaliam em simulação com agentes virtuais, ilustrando que a concepção de algoritmos se relaciona com as condições de processamento de informação, mas não fornece evidência sobre a aprendizagem de pensamento computacional por estudantes.

2.4 Construcionismo e aprendizagem

O construcionismo de Papert tem origem no construtivismo de Piaget, com quem o autor trabalhou, segundo o qual o conhecimento é construído pelo sujeito e não simplesmente transmitido (Papert, 1980, 1994). Papert acrescenta que a construção do conhecimento é favorecida quando o aprendiz cria algo que lhe interessa, e identificou no computador um meio para isso.

Nessa perspectiva, o computador é um meio para que o estudante crie artefatos e, assim, construa conhecimento, o que desloca o foco do consumo para a produção de soluções. A perspectiva fundamenta propostas de ensino de pensamento computacional baseadas na prática, como as atividades desplugadas estudadas por Brackmann (2017) e o ensino de lógica de programação no ensino médio tratado por Garcia et al. (2008). A formação em Engenharia de

Software compartilha com o construcionismo o princípio de aprender por meio da construção de soluções, embora a adesão a esse princípio não demonstre, por si só, ganhos de aprendizagem.

2.5 Engenharia de Software e aplicações de competências computacionais

Na Engenharia de Software, a capacidade de compreender problemas e estruturar soluções está ligada às atividades de análise, modelagem, desenvolvimento e implementação de sistemas. Estudos de outras áreas ilustram como procedimentos computacionais se aplicam a problemas concretos, sem que isso os torne evidência sobre o ensino do pensamento computacional.

Gonçalves et al. (2025) desenvolveram e validaram um sistema de automação de baixo custo para o gerenciamento de sistemas de bombeamento, com microcontrolador, sensores de nível e lógica de fila FIFO (first in, first out), que organiza as solicitações de acionamento de modo sequencial e evita a sobreposição de comandos. O caso ilustra a organização de instruções, a definição de condições e a estruturação de etapas lógicas na solução de um problema de engenharia, procedimentos próximos aos que o pilar do algoritmo descreve.

Neves et al. (2025), identificam crescimento dos estudos voltados à otimização do consumo energético, sobretudo em computação em nuvem, a emergência de aplicações de aprendizado de máquina nesse contexto e a necessidade de métricas padronizadas para avaliar a eficiência energética. O trabalho trata de qualidade e sustentabilidade de sistemas, sendo relevante para lembrar que construir soluções computacionais envolve decisões de arquitetura e critérios de eficiência, além da sequência lógica que resolve o problema.

No campo da organização de informações, Bruno et al. (2022), ao relatar o registro e a catalogação sistemáticos do conteúdo de vídeos para facilitar sua localização e recuperação em um acervo digital, ilustram como classificar e estruturar dados torna a informação acessível, operações que se relacionam com a abstração e o reconhecimento de padrões sendo úteis além da programação.

Santos et al. (2026) ressaltam a importância da proteção de dados pessoais, da transparência e do controle humano nas decisões administrativas, ilustrando o emprego de recursos computacionais para organizar e interpretar informações em problemas concretos e indicando que a competência técnica convive com exigências de responsabilidade no tratamento de dados.

2.6 Formação tecnológica, ética e responsabilidade social

A formação em tecnologia tem também dimensões éticas e sociais. Telles et al. (2023), discutem a importância de considerar a formação ética como parte da preparação acadêmica para o exercício profissional, complementando o desenvolvimento das habilidades técnicas e analíticas exigidas pelas atividades relacionadas à tecnologia, ou seja, a formação em engenharia é um processo que não se limita às competências técnicas.

Riegert et al. (2023), concluem que a educação é necessária para que o indivíduo conheça

e exerça seus direitos e que desigualdades de acesso à educação e à informação geram condições desiguais de exercício da cidadania. Se a formação tecnológica é entendida como parte da educação para a vida em sociedade, a ampliação do acesso a ela pode ser vista como uma das condições de participação, argumento coerente com o de Barboza e Winter (2026) sobre capacitação digital e acesso aos benefícios da ciência e da tecnologia.

Braga et al. (2026), analisam comparativamente plataformas digitais e indicam que elas podem favorecer o acesso a mercados digitais, a capacitação empreendedora, a formação de redes colaborativas e o monitoramento de impactos, desde que articulando universidades, políticas públicas e organizações da sociedade civil. Essa articulação sugere que estudantes de tecnologia podem atuar na compreensão e no desenvolvimento de soluções digitais para necessidades sociais e profissionais.

3. METODOLOGIA

Trata-se de uma pesquisa descritiva, de abordagem qualitativa, composta por levantamento bibliográfico e por levantamento de campo com questionário. O levantamento bibliográfico buscou publicações sobre pensamento computacional, construcionismo e temas relacionados, com base em artigos científicos de análise, revisão e estudos empíricos, entre eles trabalhos de Wing, Papert, França e Tedesco e Blikstein.

Participaram 29 estudantes do curso de Engenharia de Software da Universidade de Vassouras, campus Maricá, matriculados do primeiro ao oitavo período, que já haviam cursado ou estavam cursando a disciplina de Pensamento Computacional. O instrumento foi um questionário estruturado com seis afirmações, avaliadas em escala Likert de cinco pontos, na qual o participante indica seu grau de concordância com notas de 1 a 5. As afirmações, identificadas de A1 a A6 e reproduzidas integralmente na seção seguinte, abordaram a compreensão dos conceitos fundamentais do pensamento computacional (A1), a melhora percebida na resolução de problemas após aprender sobre o tema (A2), a importância de uma disciplina de pensamento computacional na formação acadêmica (A3), a contribuição para o desenvolvimento da lógica de programação (A4), o reconhecimento de sua importância em diversas áreas da vida, incluindo educação, trabalho e problemas cotidianos (A5) e a importância de aplicar o conhecimento também em escolas de nível fundamental e médio (A6).

Os dados foram organizados e analisados com a linguagem de programação Python, que permitiu calcular a distribuição percentual das respostas em cada afirmação. Na tabulação, as respostas distribuíram-se nas categorias neutro ou indiferente, concordo e concordo totalmente. A análise é descritiva: não foram aplicados testes estatísticos inferenciais, e os resultados descrevem a amostra, sem generalização para outras populações. Como o instrumento reúne autoavaliações dos participantes, os resultados expressam percepções, e não medidas diretas de habilidades cognitivas ou de desempenho acadêmico.

4. RESULTADOS E DISCUSSÃO

A Tabela 1 apresenta a distribuição percentual das respostas às seis afirmações. Em todas, as respostas concentraram-se nas categorias de concordância: a proporção de respostas neutras

variou de ausente a 10,00%, e a tabulação não registra respostas de discordância.

Tabela 1 - **Distribuição percentual das respostas às afirmações do questionário**

Afirmação	Neutro/ indiferente (%)	Concordo (%)	Concordo totalmente (%)
A1. Compreendo os conceitos fundamentais do Pensamento Computacional e sua importância	6,67	66,67	26,67
A2. Percebi melhora em minha capacidade de resolução de problemas após aprender sobre Pensamento Computacional	6,67	56,67	36,67
A3. Considero importante ter uma disciplina de Pensamento Computacional na minha formação acadêmica	—	30,00	70,00
A4. Compreender o Pensamento Computacional auxiliou-me no desenvolvimento de lógica de programação	3,33	40,00	56,67
A5. Reconheço a importância do Pensamento Computacional em diversas áreas da vida, incluindo educação, trabalho e resolução de problemas cotidianos	10,00	36,67	53,33
A6. Considero importante a aplicação do conhecimento sobre Pensamento Computacional não apenas no ambiente universitário, mas também em escolas de nível fundamental e médio para melhor formação pessoal	—	36,67	63,33

Fonte: Os autores.

Em A1, 66,67% dos participantes concordaram e 26,67% concordaram totalmente que compreendem os conceitos fundamentais do pensamento computacional e sua importância, o que totaliza 93,34% de concordância, e 6,67% foram neutros. O resultado indica que a maioria se considera familiarizada com o conceito, o que é coerente com o fato de os participantes terem cursado ou estarem cursando a disciplina. A concordância total (26,67%) foi bem menor que a simples, o que pode refletir uma autoavaliação cautelosa diante de um conceito cuja definição segue em discussão (Denning, 2017; Valente, 2019), hipótese que os dados não permitem testar. Trata-se, em todo caso, de compreensão declarada, e não verificada por prova de conhecimento.

Em A2, 56,67% concordaram e 36,67% concordaram totalmente que perceberam melhora em sua capacidade de resolução de problemas após aprender sobre pensamento computacional, totalizando 93,34% de concordância, com 6,67% de respostas neutras. A percepção de melhora é compatível com o que Conforto et al. (2018) e De Carvalho et al. (2013) discutem para a educação básica e o ensino médio, a saber, a associação entre o pensamento computacional, a autonomia na resolução de problemas e o raciocínio lógico. A comparação é de natureza conceitual, pois aqueles estudos têm outras populações e delineamentos e não confirmam o resultado aqui observado. Do ponto de vista educacional, o dado sugere que os estudantes atribuem à disciplina utilidade para enfrentar problemas, o que pode orientar o planejamento de atividades que explicitem decomposição, abstração e construção de algoritmos. O limite é metodológico: a afirmação pede ao participante que compare

retrospectivamente sua capacidade antes e depois, sem medida prévia, sem grupo de comparação e sem controle de outras experiências formativas do curso. A melhora percebida não permite afirmar melhora efetiva nem atribuir causa.

Em A4, 56,67% concordaram totalmente e 40,00% concordaram que compreender o pensamento computacional os auxiliou no desenvolvimento da lógica de programação, o que soma 96,67% de concordância, e 3,33% foram neutros. A concordância total foi mais frequente em A4 do que em A2 (36,67%). O resultado é compatível com a relação entre pensamento computacional e programação, já que escrever programas exige decompor problemas, reconhecer padrões e definir sequências de instruções, como ilustra a aplicação de lógica sequencial a um controle de bombeamento descrita por Gonçalves et al. (2025). O limite está na separação das contribuições: em um curso de Engenharia de Software, os estudantes cursam outras disciplinas de programação, e a pesquisa não distingue a contribuição específica do pensamento computacional da de outras experiências formativas. Lógica de programação e pensamento computacional também não são sinônimos, de modo que a resposta expressa a relação que os participantes percebem entre ambos.

Quanto à importância de ter uma disciplina de pensamento computacional na formação acadêmica (A3), 70,00% concordaram totalmente e 30,00% concordaram, sem respostas neutras, de modo que todos os participantes consideraram a disciplina importante. É a afirmação com maior proporção de concordância total. O resultado indica forte aceitação da disciplina entre quem a cursou ou cursa e se alinha à defesa de Wing (2008) de que o pensamento computacional seja incorporado à formação, e não restrito a profissionais da computação. A implicação educacional é que a manutenção da disciplina no currículo conta com apoio dos estudantes, dado relevante para a gestão do curso, mas que não substitui a avaliação de sua efetividade. Como os respondentes têm vínculo com a disciplina, a unanimidade pode refletir também a experiência positiva com ela, e o estudo não permite estimar como estudantes sem esse contato a avaliariam.

Em A5, 53,33% concordaram totalmente e 36,67% concordaram que reconhecem a importância do pensamento computacional em diversas áreas da vida, incluindo educação, trabalho e resolução de problemas cotidianos, o que soma 90,00% de concordância, e 10,00% foram neutros, a maior proporção de respostas neutras do questionário. A concordância menor que a das demais afirmações pode indicar que a aplicabilidade ampla é mais difícil de perceber do que a utilidade imediata para a programação, hipótese que os dados não testam. A literatura traz exemplos de procedimentos computacionais em contextos variados, como a organização e a recuperação de registros audiovisuais (Bruno et al., 2022), o apoio à decisão em política socioassistencial com análise territorial e inteligência artificial (Santos et al., 2026), a automação de sistemas de bombeamento (Gonçalves et al., 2025) e a busca por eficiência na arquitetura de software (Neves et al., 2025). Esses estudos servem como ilustração de que as competências associadas ao conceito encontram aplicação fora da programação. No plano educacional, o resultado sugere que atividades com problemas de áreas diversas podem tornar essa aplicabilidade mais visível aos estudantes. O dado mede reconhecimento declarado, não transferência efetiva das competências a outros domínios.

Em A6, 63,33% concordaram totalmente e 36,67% concordaram que é importante aplicar o conhecimento sobre pensamento computacional também em escolas de nível fundamental e médio, de modo que todos os participantes concordaram. Esse posicionamento converge com o foco da literatura brasileira na educação básica (Brackmann, 2017; Conforto et al., 2018; França & Tedesco, 2015) e com a proposta de Wing (2006) de que o pensamento computacional integre a formação desde a infância. Pode ser lido também à luz do argumento de Riegert et al. (2023), para quem a educação é condição do exercício consciente de direitos e as desigualdades de acesso a ela geram desigualdades de cidadania, e do argumento de Barboza e Winter (2026) sobre a capacitação digital e o acesso aos benefícios da ciência e da tecnologia. Para o desenho curricular, o dado indica que estudantes que cursaram a disciplina valorizam sua oferta em etapas anteriores, mas não informa como a inserção deveria ocorrer nem quais seriam seus efeitos, pois os participantes opinam sobre níveis de ensino que não estão cursando.

Tomados em conjunto, os resultados mostram percepção predominantemente positiva: nenhuma afirmação teve respostas de discordância e a concordância variou de 90,00% a 100%. A concordância total foi mais frequente em A3 (70,00%) e A6 (63,33%), afirmações sobre a importância atribuída ao pensamento computacional, do que em A2 (36,67%) e A1 (26,67%), que tratam da compreensão e da melhora percebida pelos próprios participantes. Os estudantes expressam, assim, concordância mais intensa quanto à relevância da competência do que quanto ao próprio domínio e progresso. Esses resultados descrevem percepções de 29 estudantes de um único campus e não indicam causalidade nem desenvolvimento cognitivo mensurado.

4. CONCLUSÕES

Este estudo investigou a contribuição percebida do pensamento computacional para a formação acadêmica de estudantes de Engenharia de Software, com ênfase nas habilidades de resolução de problemas e na lógica de programação. Os resultados revelam uma percepção consistentemente positiva de sua relevância educacional, indicando que os estudantes reconhecem o pensamento computacional como um componente importante de sua formação acadêmica. Em particular, os participantes relataram melhorias percebidas em sua capacidade de resolver problemas e reconheceram a contribuição do pensamento computacional para o desenvolvimento da lógica de programação. Esses achados corroboram a proposição central do estudo de que o pensamento computacional é valorizado pelos estudantes como uma competência relevante para enfrentar desafios acadêmicos.

A contribuição deste estudo consiste em examinar o pensamento computacional sob a perspectiva de estudantes de graduação em Engenharia de Software, ampliando uma discussão predominantemente voltada aos seus fundamentos conceituais e potencial educacional para o contexto do ensino superior. Ao relacionar perspectivas teóricas consolidadas sobre pensamento computacional e desenvolvimento cognitivo às experiências acadêmicas relatadas pelos estudantes, a pesquisa oferece evidências empíricas sobre como essa competência é percebida no âmbito da formação tecnológica profissional. Os resultados também evidenciam a relevância atribuída ao pensamento computacional para além das atividades de programação, especialmente quanto à sua aplicabilidade em diferentes disciplinas acadêmicas e em situações cotidianas de resolução de problemas.

Sob a perspectiva educacional, os achados reforçam a importância da incorporação do pensamento computacional aos currículos de Engenharia de Software como componente da formação acadêmica e profissional. Sua contribuição percebida para o raciocínio lógico e a resolução de problemas sugere que o pensamento computacional pode oferecer uma fundamentação conceitual relevante para atividades de aprendizagem que exigem análise sistemática, abstração e estruturação de soluções. Além disso, o reconhecimento unânime dos participantes quanto à sua importância na formação acadêmica favorece a reflexão sobre sua integração em contextos educacionais mais amplos, incluindo etapas anteriores da educação formal. Entretanto, essas implicações devem ser compreendidas no âmbito das percepções dos estudantes, e não como evidências de melhorias comprovadas no desempenho cognitivo.

Os resultados devem ser interpretados considerando os limites metodológicos da pesquisa, particularmente seu delineamento descritivo, a participação de 29 estudantes de um único campus universitário e a utilização de respostas autorrelatadas em questionário. Dessa forma, os achados não estabelecem relações causais entre o contato com o pensamento computacional e o desenvolvimento cognitivo mensurável, tampouco demonstram melhorias objetivas no desempenho acadêmico. Investigações futuras poderão avançar nessa discussão por meio de amostras maiores e mais diversificadas, delineamentos longitudinais e avaliações diretas das habilidades de resolução de problemas e programação. Em síntese, este estudo contribui para a compreensão do pensamento computacional no ensino superior ao evidenciar sua relevância percebida pelos estudantes de Engenharia de Software e reforçar sua importância como competência educacional no contexto contemporâneo da formação tecnológica.

Referências Bibliográficas

Barboza, D. V., & Winter, E. (2026). Indicadores de ciência, tecnologia e inovação no combate à vulnerabilidade social. *Revista da Seção Judiciária do Rio de Janeiro – Auditorium, 30*(64), 236–252. https://doi.org/10.30749/2177-8337.v30n64p236-252

Blikstein, P. (2008). *O pensamento computacional e a reinvenção do computador na educação*.

Brackmann, C. P. (2017). *Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica* [Tese de doutorado, Universidade Federal do Rio Grande do Sul]. Porto Alegre.

Braga, R. O. do C., Storino, A. L. B., & Avelar, K. E. S. (2026). Plataformas digitais para inovação social: uma análise comparativa para o desenvolvimento de um ecossistema digital de apoio ao empreendedorismo sustentável. *RECIMA21 – Revista Científica Multidisciplinar, 7*(7). https://doi.org/10.47820/recima21.v7i7.8659

Bruno, S. F., Barboza, D. V., Pereira, G. da S., Lopes, F. da S., Maciel, A. C. da S. I., Maior, L. L. S., Pontes, T. M. M. C. S., & Balzi, Y. V. (2022). Decupagem de registros audiovisuais do Projeto Fauna Brasil – UFF. *Humanidades & Inovação, 9*(9), 375–379.

Conforto, D., et al. (2018). Pensamento computacional na educação básica: interface tecnológica na construção de competências do século XXI. *Revista Brasileira de Ensino de Ciências e Matemática, 1*(1).

De Carvalho, M. L. B., Chaimowicz, L., & Moro, M. M. (2013). Pensamento computacional no ensino médio mineiro. In *Workshop de Educação em Informática (WEI)*.

Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM, 60*(6), 33–39.

França, R., & Tedesco, P. (2015). Desafios e oportunidades ao ensino do pensamento computacional na educação básica no Brasil. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação* (p. 1464).

Garcia, R. E., Correia, R. C. M., & Shimabukuro, M. H. (2008). Ensino de lógica de programação e estruturas de dados para alunos do ensino médio. In *XVII WEI – Workshop sobre o Ensino de Computação* (pp. 246–249). Belém do Pará.

Gonçalves, D. A., Neves, A. C. M., Garrido, M. A. D., Barboza, D. V., & Castro, T. R. de. (2025). Automação de baixo custo para gestão eficiente de sistemas de bombeamento: desenvolvimento e validação de um controle sequencial. *Revista da Micro e Pequena Empresa, 19*(2), 79–93. https://doi.org/10.6034/rmpe.v19i2.2179

Melo, F. S. S. dos, Palopoli, F. F., Santos, R. M. dos, Mello, S. T. A. de, Carlos, A. S., Ferreira, V. M. da S., Boente, A. N. P., Carvalho, M. G. P. de, D'Durand, T. M. L., & Costa, C. B. de J. da. (2026). Arquitetura algorítmica para atenção sustentável: o modelo Be-Productive como resposta à sobrecarga cognitiva no capitalismo de vigilância. *Revista Tópicos, 4*(34), 1–21. https://doi.org/10.70773/revistatopicos/781363235

Neves, A. C. M., Gonçalves, D. A., Castro, T. R. de, Garrido, M. A. D., & Barboza, D. V. (2025). Arquitetura de software para eficiência energética: uma revisão bibliométrica. *Revista da Micro e Pequena Empresa, 19*(1), 21–40. https://doi.org/10.6034/rmpe.v19i1.2163

Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas*. Basic Books.

Papert, S. (1994). *A máquina das crianças: repensando a escola na era da informática*. Artes Médicas.

Riegert, M. G., Cavalcanti, M. T., & Friede, R. (2023). Educação para a construção da cidadania. *Humanidades & Inovação, 10*(13), 245–262.

Rodrigues, R. S., et al. (2015). Análise dos efeitos do Pensamento Computacional nas habilidades de estudantes no ensino básico: um estudo sob a perspectiva da programação de computadores. In *Simpósio Brasileiro de Informática na Educação (SBIE)* (p. 121).

Santos, R. M. dos, Rodrigues, F. C. T., Vilela, J. A. R., & Ferreira, V. M. da S. (2026). Análise territorial e inteligência artificial na política socioassistencial: o aplicativo Visível como instrumento de apoio à decisão no enfrentamento das vulnerabilidades sociais. *Revista da Seção Judiciária do Rio de Janeiro – Auditorium, 30*(64), 103–128. https://doi.org/10.30749/2177-8337.v30n64p103-128

Telles, B. H. M. V., Souza-Rodrigues, D. de, Esteves, A. C. L., Silva, F. A. da, Silva, A. F. de O., & Barboza, D. V. (2023). Perception of future production engineers on organizational ethics and social responsibility. *Concilium, 23*(12), 123–138. https://doi.org/10.53660/CLM-1500-23H33

Valente, J. A. (2019). Pensamento computacional, letramento computacional ou competência digital? Novos desafios da educação. *Revista Educação e Cultura Contemporânea, 16*(43), 147–168.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35.

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 366*(1881), 3717–3725.

Wing, J. M. (2021). Pensamento computacional. *Educação e Matemática, 162*, 2–4.

Zanetti, H., Borges, M., & Ricarte, I. (2016). Pensamento computacional no ensino de programação: uma revisão sistemática da literatura brasileira. In *Simpósio Brasileiro de Informática na Educação (SBIE)* (p. 21).