Aprendizagem na Era do Offload Cognitivo e o Papel da Programação Competitiva

Gabriel P. Falcão, UFMS
Mariana G. Ramires, UFMS

Resumo

Este artigo propõe uma reflexão crítica sobre a integração da Inteligência Artificial, em especial dos modelos de linguagem, ao processo de aprendizagem e à prática da computação. Discutimos os desafios impostos pelo offload cognitivo e apontamos a prática da programação competitiva como uma prática pedagógica de contraposição a essa nova era de acesso à informação.

1. Introdução

Nos últimos anos, a Inteligência Artificial (IA) passou a incorporar um papel de destaque em diferentes áreas, como a educação básica [1], o judiciário [2] e a medicina [3]. Diante desse cenário, não é surpreendente que na computação sua presença seja ainda mais disruptiva e difundida. Práticas de desenvolvimento assistido por IA deixaram de ser puramente experimentais e estão transformando a esteira de produção de software, da geração de código com modelos de linguagem de grande escala (LLMs) [4] até a automação da geração de testes [5].

Porém, à medida que a produtividade técnica atinge novos patamares, a comunidade científica começa a alertar para as consequências psicológicas e educacionais dessa automação. O termo offload cognitivo expressa o fenômeno em que o esforço analítico e o processo de raciocínio são integralmente terceirizados para a IA, reduzindo os estímulos cognitivos essenciais para o desenvolvimento cerebral [6].

Voltamos nossa atenção, então, para um passo antes da indústria: o ambiente de aprendizagem. À primeira vista, o ato de aprender e a dependência de ferramentas de IA, como os chatbots, parecem opostos, sobretudo sob a ótica do estímulo cognitivo. Torna-se essencial a valorização de ambientes que estimulem o raciocínio autônomo e de alta complexidade. É nesse cenário que a programação competitiva — com destaque para a Maratona SBC de Programação, que anualmente reúne milhares de estudantes de centenas de instituições de ensino brasileiras —, ganha uma nova dimensão pedagógica.

O objetivo deste texto é propor uma reflexão sobre como a programação competitiva atua como um ambiente essencial para o desenvolvimento do raciocínio computacional independente, contrapondo-se à tendência de terceirização cognitiva imposta pelas ferramentas de IA.

2. IA na Computação

Nos últimos anos, a Inteligência Artificial deixou de ser uma tecnologia emergente para tornar-se parte integrante do cotidiano da Computação. De acordo com o Stack Overflow Developer Survey 2025 [7], 84% dos respondentes utilizam ou planejam utilizar ferramentas de IA em seu fluxo de desenvolvimento, evidenciando a rápida consolidação dessas tecnologias na indústria.

Ferramentas baseadas em LLMs vêm sendo aplicadas em diversas atividades da Engenharia de Software, incluindo geração e compreensão de código, documentação, testes, depuração, manutenção e apoio ao processo de desenvolvimento como um todo [8]. Esse cenário torna inevitável discutir não apenas o impacto da IA no ciclo de vida do software, mas também seus efeitos sobre a formação dos futuros profissionais da Computação.

Entre as mudanças observadas na prática profissional, destaca-se o chamado vibe coding, abordagem em que desenvolvedores utilizam modelos de IA como participantes ativos do processo de programação, concentrando seus esforços na formulação dos problemas e na avaliação das soluções produzidas. Sob essa perspectiva, a Inteligência Artificial representa mais uma etapa no processo de abstração que acompanha a evolução da Computação. Assim como linguagens de alto nível, bibliotecas e frameworks reduziram a necessidade de lidar com detalhes de implementação, os sistemas atuais ampliam a produtividade ao automatizar atividades relacionadas ao próprio desenvolvimento de software.

Diante desse cenário, torna-se difícil imaginar a formação e a atuação profissional em Computação sem a presença dessas ferramentas. A questão, portanto, deixa de ser se a Inteligência Artificial deve ser utilizada e passa a ser como incorporá-la aos processos de aprendizagem sem comprometer o desenvolvimento das competências fundamentais da área.

3. Aprendizagem na era do offload cognitivo

A facilidade de acesso a respostas e soluções representa uma das principais vantagens das ferramentas modernas de Inteligência Artificial. Entretanto, essa mesma característica nos leva a refletir sobre a relação entre acesso à informação e construção do conhecimento.

A facilidade de acesso à informação não é um fenômeno novo, mas a Inteligência Artificial modifica sua escala e profundidade. Essa mudança intensifica um comportamento já observado por Sparrow, Liu e Wegner, segundo o qual indivíduos tendem a memorizar menos informações quando acreditam que poderão recuperá-las posteriormente por meios digitais, fenômeno conhecido como Google Effect [9]. Assim, a IA não apenas amplia o acesso à informação, mas também favorece a terceirização de etapas cognitivas tradicionalmente envolvidas na aprendizagem.

Um dos muitos usos da IA na computação é como ferramenta de aprendizado. Pereira Filho, Souza e Paula [10] realizaram experimentos qualitativos e quantitativos nos quais solicitaram ao ChatGPT que resolvesse questões comumente aplicadas em cursos de iniciação em programação. Embora o modelo tenha produzido respostas corretas em mais de 80% das instâncias, também foram identificados problemas como soluções excessivamente complexas ou desprovidas de explicação, o que pode ser prejudicial ao processo de aprendizado dos estudantes.

Além disso, Valovy e Buchalcevova [11] conduziram um estudo experimental com 38 estudantes de engenharia de software e 5 profissionais para avaliar o impacto do uso de IAs generativas (como ChatGPT e Copilot) no desenvolvimento. Nas entrevistas qualitativas subsequentes, conduzidas até a saturação dos dados, embora a percepção geral sobre as ferramentas tenha sido positiva, sobressaiu-se uma ressalva entre os discentes: a maioria dos alunos entrevistados (80%, ou 4 de 5) reportou que o aprendizado do conteúdo e dos fundamentos de programação seria mais efetivo sem a interferência inicial da IA.

Esses resultados não devem ser interpretados como uma rejeição ao uso da Inteligência Artificial no ensino. Pelo contrário, eles evidenciam uma questão mais profunda: a diferença entre ter acesso à informação e efetivamente transformá-la em conhecimento.

Sob a perspectiva da Ciência da Informação, informação e conhecimento não são conceitos equivalentes. Conforme argumenta Malheiro da Silva [12], a informação somente adquire significado quando é interpretada e apropriada pelo sujeito em um determinado contexto. Em outras palavras, o simples contato com uma resposta correta não garante sua compreensão, tampouco a capacidade de aplicá-la em novas situações.

Essa discussão torna-se ainda mais relevante em um cenário caracterizado pela abundância informacional. Satur e Malheiro da Silva [13] defendem que a sociedade contemporânea exige o desenvolvimento de competências que permitam localizar, compreender, avaliar criticamente e utilizar informações de forma efetiva. Assim, o desafio educacional deixa de ser apenas o acesso ao conteúdo e passa a envolver a capacidade de transformá-lo em conhecimento aplicável.

Ao aplicar essa reflexão na formação em Computação, observamos que grande parte da aprendizagem ocorre durante o próprio processo de resolução de problemas. A formulação de hipóteses, a identificação de erros, a comparação entre diferentes abordagens e o refinamento de soluções são etapas que contribuem para a construção do aprendizado e do pensamento computacional. Embora ferramentas de IA possam acelerar esse processo e oferecer suporte valioso aos estudantes, elas não eliminam a necessidade de participação ativa na construção do conhecimento.

Nesse contexto, torna-se necessário refletir sobre quais competências permanecem essenciais independentemente das tecnologias utilizadas.

4. A importância do raciocínio computacional

Se o acesso à informação e à geração de código torna-se cada vez mais simples, as competências exigidas dos profissionais da Computação também passam por uma transformação. Mais do que produzir código, espera-se que desenvolvedores sejam capazes de compreender problemas, avaliar diferentes estratégias de solução, interpretar requisitos e analisar criticamente os resultados produzidos por ferramentas inteligentes.

Nesse contexto, ganha destaque o conceito de raciocínio computacional, entendido como a capacidade de formular problemas de maneira que suas soluções possam ser descritas e executadas de forma sistemática. Esse processo envolve habilidades como abstração, decomposição de problemas, reconhecimento de padrões e construção de algoritmos — competências que permanecem essenciais independentemente da linguagem de programação ou das ferramentas utilizadas.

Paradoxalmente, acreditamos que o avanço da Inteligência Artificial torna essas habilidades ainda mais relevantes. Embora modelos de linguagem sejam capazes de gerar implementações cada vez mais sofisticadas, sua utilização efetiva depende da capacidade do usuário de definir corretamente o problema, avaliar a adequação das soluções propostas, identificar limitações e adaptar respostas a diferentes contextos. Em outras palavras, quanto menor o esforço necessário para produzir código, maior passa a ser a importância da compreensão dos problemas que esse código busca resolver.

Dessa forma, a formação em Computação não deve concentrar seus esforços apenas no domínio de novas ferramentas, mas também na criação de oportunidades para que estudantes desenvolvam o raciocínio computacional por meio da resolução ativa de problemas. É nesse contexto que ambientes voltados à prática sistemática da resolução de problemas adquirem papel fundamental na formação de futuros profissionais.

5. O Papel da Programação Competitiva

Diante da necessidade de desenvolver competências relacionadas ao raciocínio computacional, a Programação Competitiva apresenta-se como um ambiente de aprendizagem particularmente adequado para a formação em Computação. Diversos trabalhos têm investigado a Programação Competitiva como ferramenta de apoio ao ensino da área. No contexto brasileiro, Nogueira e Pozzer [14] analisaram um projeto preparatório para a Maratona SBC de Programação e observaram que os participantes relataram evolução tanto no desempenho competitivo quanto na percepção do próprio aprendizado.

Ao contrário de atividades baseadas na reprodução de exemplos ou na simples consulta a soluções prontas, a Programação Competitiva coloca o estudante diante de problemas inéditos, exigindo a interpretação cuidadosa do enunciado, a identificação das restrições, a formulação de

hipóteses e a construção gradual de uma solução. Durante esse processo, erros tornam-se parte natural da aprendizagem, levando o competidor a revisar estratégias, comparar diferentes abordagens e refinar seu raciocínio. Sob a perspectiva discutida na Seção 3, esse processo favorece a apropriação da informação e sua transformação em conhecimento efetivo.

As habilidades desenvolvidas nesse contexto vão além da implementação de algoritmos. Resolver problemas competitivos exige abstração, decomposição de problemas, reconhecimento de padrões, análise de complexidade e tomada de decisão sob restrições — competências diretamente relacionadas ao raciocínio computacional e que permanecem relevantes independentemente das ferramentas utilizadas para escrever código [15].

O crescimento da Inteligência Artificial não reduz a importância desse processo; ao contrário, reforça sua necessidade. Considere, por exemplo, um problema clássico envolvendo caminhos mínimos em grafos. Um modelo de linguagem pode rapidamente sugerir uma implementação correta do algoritmo de Dijkstra. Entretanto, identificar que o problema pode ser modelado como um grafo, compreender que as restrições permitem a utilização desse algoritmo, avaliar sua complexidade e verificar a correção da solução continuam sendo responsabilidades do estudante. A geração automática de código não substitui a compreensão do problema.

Isso não significa que a Inteligência Artificial deva ser excluída do processo de aprendizagem da Programação Competitiva. Durante o treinamento, essas ferramentas podem atuar como tutores, auxiliando na compreensão de conceitos, explicando algoritmos, sugerindo casos de teste e analisando implementações. Entretanto, a resolução efetiva dos problemas continua dependendo da participação ativa do estudante. Nas competições oficiais, inclusive, consultas externas são proibidas, fazendo com que o competidor dependa exclusivamente do próprio raciocínio para enfrentar os desafios propostos.

6. Considerações finais

Assim, compreende-se a programação competitiva como uma poderosa aliada do processo de ensino da computação em uma era propensa ao offload cognitivo. Em um cenário caracterizado pelo acesso imediato à informação e pela crescente automação da implementação de software, não se diminui a relevância do fator humano; pelo contrário, desloca-se o eixo de exigência profissional para a capacidade profunda de abstração e formulação de problemas. Nesse contexto, práticas como a Maratona SBC de Programação assumem o papel fundamental de oferecer um ambiente controlado que estimule os estudantes a desenvolverem as competências necessárias para compreender problemas, construir soluções e, consequentemente, utilizar ferramentas de IA de forma crítica, consciente e efetiva.

Referências

[1] Thiago da Silva Fialho, Christian de Vargas Silva, Fabrício Herpich, Marina Sérgio Carradore, Eliane Pozzebon. Inteligência artificial e educação básica: uma revisão sistemática das aplicações. TEyET, v. 41, 2025.
[2] Carolina Watanabe, Taynara França. O Impacto da Inteligência Artificial no Judiciário. Revista Em Tempo, v. 24, n. 1, p. 47–73, 2025.
[3] Vanessa Schmidt Bortolini, Alexandre de Souza Garcia, Wilson Engelmann. USO ÉTICO DA INTELIGÊNCIA ARTIFICIAL NA SAÚDE: UMA REVISÃO SISTEMÁTICA DA LITERATURA. Revista Jurídica Gralha Azul - TJPR, v. 1, n. 28, 2025.
[4] Juyong Jiang, Fan Wang, Jiasi Shen, Sungju Kim, Sunghun Kim. A Survey on Large Language Models for Code Generation. ACM Trans. Softw. Eng. Methodol., v. 35, n. 2, 2026.
[5] Junjie Wang, Yuchao Huang, Chunyang Chen, Zhe Liu, Song Wang, Qing Wang. Software Testing With Large Language Models: Survey, Landscape, and Vision. IEEE Transactions on Software Engineering, v. 50, n. 4, p. 911–936, 2024.
[6] Michael Gerlich. AI Tools in Society: Impacts on Cognitive Offloading and the Future of Critical Thinking. Societies, v. 15, n. 1, 2025.
[7] Stack Overflow. 2025 Stack Overflow Developer Survey: AI, 2025. Disponível em: https://survey.stackoverflow.co/2025/ai. Acesso em: 7 jul. 2026.
[8] Xinyi Hou, Yanjie Zhao, Yue Liu, Zhou Yang, Kailong Wang, Li Li, Xiapu Luo, David Lo, John Grundy, Haoyu Wang. Large Language Models for Software Engineering: A Systematic Literature Review. ACM Transactions on Software Engineering and Methodology, v. 33, n. 8, p. 220:1–220:79, 2024.
[9] Betsy Sparrow, Jenny Liu, Daniel M. Wegner. Google Effects on Memory: Cognitive Consequences of Having Information at Our Fingertips. Science, v. 333, n. 6043, p. 776–778, 2011.
[10] Luiz Pereira Filho, Talita Souza, Luciano Paula. Análise das Respostas do ChatGPT em Relação ao Conteúdo de Programação para Iniciantes. In: Anais do XXXIV Simpósio Brasileiro de Informática na Educação. SBC, p. 1738–1748, 2023.
[11] Marcel Valovy, Alena Buchalcevova. The Psychological Effects of AI-Assisted Programming on Students and Professionals, 2023.
[12] Armando Malheiro da Silva. A Informação: da Compreensão do Fenómeno e Construção do Objecto Científico. Porto: Edições Afrontamento, 2006.
[13] Roberto Vilmar Satur, Armando Malheiro da Silva. A Aprendizagem Visando a Competência em Informação na Sociedade em Tempos de Infoesfera. Perspectivas em Gestão & Conhecimento, v. 10, n. Especial, p. 2–22, 2020.
[14] Jonathan Weber Nogueira, Fernando Pedrazzi Pozzer. Uma análise dos resultados obtidos no primeiro ano do projeto preparatório para a Maratona de Programação. Revista ComInG, v. 7, n. 1, p. 31–38, 2023.
[15] Jeannette M. Wing. Computational Thinking. Communications of the ACM, v. 49, n. 3, p. 33–35, 2006.