



VI Encontro de Iniciação Científica e Tecnológica  
VI EnICT  
ISSN: 2526-6772  
IFSP – Câmpus Araraquara  
21 e 22 de outubro de 2021



## SOLUÇÃO DE PROBLEMAS COMPUTACIONAIS E O PENSAMENTO COMPUTACIONAL E CRIATIVO: UM ESTUDO DE CASO

GISLAINE CRISTINA MICHELOTI ROSALES<sup>1</sup>, EDNILSON GERALDO ROSSI<sup>1</sup>, JANAINA CINTRA ABIB<sup>1</sup>, LAURA STHEFANY COLOMBO<sup>2</sup>

<sup>1</sup> Docente no IFSP Câmpus Araraquara, gislaine@ifsp.edu.br; ednilsonrossi@ifsp.edu.br; janaina@ifsp.edu.br. <sup>2</sup> Estudante de Curso Técnico em Informática, IFSP Câmpus Araraquara, laura.sthefany@aluno.ifsp.edu.br.

Área de conhecimento (Tabela CNPq): 1.03.03.04-9

**RESUMO:** Este artigo apresenta uma investigação em andamento sobre possíveis relações entre habilidades para solução de problemas computacionais e o desenvolvimento do pensamento computacional e do pensamento criativo. O modelo utilizado é baseado em quatro dimensões: Compreender, Reproduzir, Aplicar e Avaliar. A pesquisa se aplica aos alunos do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, disciplina Estrutura de Dados; e alunos do Curso Técnico em Informática Integrado ao Ensino Médio, disciplina Algoritmos e Programação. Os alunos dessas disciplinas são divididos em duas turmas de 20 alunos, totalizando quatro amostras distintas, sendo duas para cada uma das duas disciplinas. O modelo é aplicado a apenas uma das duas turmas em cada disciplina. Inicialmente, realizou-se a caracterização das amostras que considera diferenças associadas ao ambiente e ao contexto dos participantes. Estratégias de ensino e aprendizagem são empregadas ao longo das disciplinas nas duas amostras de controle, assim como instrumentos de avaliação para determinar o progresso dos estudantes participantes da pesquisa, para as quatro amostras, a fim de identificar possíveis relações entre as variáveis de estudo nas duas turmas onde o modelo de fixação da aprendizagem foi aplicado e nas duas turmas em que isso não ocorreu.

**PALAVRAS-CHAVE:** programação; pensamento algorítmico; pensamento computacional; criatividade.

## INTRODUÇÃO

Pensamento Computacional (PC), do inglês *Computational Thinking*, assim como outros conceitos relacionados, tais como pensamento algorítmico, programação e criatividade computacional, tem promovido aumento crescente de pesquisas por educadores de todo o mundo (Israel-Fishelson e Hershkovitz, 2021), em que, muitas dessas pesquisas, sugerem que tal conceito seja encarado como habilidade fundamental, assim como a alfabetização (Bocconi et al., 2016). Os mesmos autores apontam, ainda, questões a serem investigadas, tais como Qual é a relação do PC com a programação/codificação? O que significa e como avaliar PC? Como promover PC?

Não há consenso sobre a definição de PC, uma definição amplamente usada como referência na área é proposta por Wing (2011) e oferece duas perspectivas: (i) PC é um processo de pensamento que independe da tecnologia; (ii) PC é uma forma de resolver problemas que envolve diferentes habilidades, incluindo a capacidade de projetar soluções que podem ser executadas por um computador ou por um ser humano.

Ademais, a Base Nacional Comum Curricular, Brasil (2018), utilizada como base para formulação dos currículos escolares, menciona o PC como conteúdo voltado para Matemática, principalmente para a segunda fase do Ensino Fundamental. Uma vez que essa estratégia de pensamento se faz relevante desde a Educação Básica (Batista, Horta e Fonseca, 2020), pois proporciona a capacidade solução de problemas em qualquer âmbito, há real necessidade de aprofundamento no conceito, desenvolvimento e relações do PC.

Por acreditar em possíveis associações entre o pensamento computacional, o pensamento criativo e a habilidade de programação (Paixão e Rodrigues, 2018), esta pesquisa considera esses três conceitos e analisa

suas relações. Sendo assim, objetivo geral deste trabalho é investigar relações e associações entre o desenvolvimento de habilidades e competências para a solução problemas computacionais e o desenvolvimento do pensamento criativo e o desenvolvimento do pensamento computacional.

## FUNDAMENTAÇÃO TEÓRICA

Dentre as diversas definições atuais, pode-se extrair conceitos e habilidades que aparecem, repetidamente, na literatura associados ao desenvolvimento do Pensamento Computacional, os quais incluem: abstração (Barr e Stephenson, 2011; Lee et al., 2011; Grover e Pea, 2013; Selby e Woollard, 2013; Angeli et al., 2016), pensamento algorítmico (Selby & Woollard, 2013) e algoritmos, automação (Barr e Stephenson, 2011; Lee et al., 2011), decomposição (Barr e Stephenson, 2011; Grover e Pea, 2013; Selby e Woollard, 2013; Angeli et al., 2016), depuração (Grover e Pea, 2013), generalização (Selby e Woollard, 2013; Angeli et al., 2016), análise (Lee et al., 2011), programação/codificação. A programação/codificação, como um componente do PC, pode ser vista como uma ferramenta de aprendizagem que pode estar presente no currículo escolar obrigatório, inclusive sendo abordado como habilidade transversal.

Diversos estudos apontam, ainda, que o pensamento criativo (Israel-Fishelson et al., 2021) está relacionado com o pensamento computacional e há influências em ambos os sentidos. Amabile e Pillemer (2012) apontam que a criatividade pode ser afetada por fatores individuais e fatores externos, que seriam a influências dos fatores socioambientais. Os fatores individuais mencionados pelas autores incluem: (i) habilidades no domínio (habilidade técnica, talento inato); (ii) habilidades relevantes para a criatividade, como o estilo cognitivo flexível, personalidade, persistência; e, (iii) motivação para realização de tarefas. De acordo com as autoras, o componente externo, ambiente social, pode influenciar cada um dos componentes individuais. Todavia, a influência mais imediata e prevalente do ambiente é exercida sobre o componente motivacional. Pode-se considerar que a criatividade pode ser vista como um processo cognitivo e social, e ainda como uma habilidade que pode ser aprendida, praticada e melhorada.

A taxonomia de Bloom (Bloom 1956/2001) utiliza seis níveis de classificação da aprendizagem cognitiva que vão desde ganhar conhecimento em seu nível mais fundamental, compreender ideias e fatos básicos, ser capaz de aplicar a aprendizagem na resolução de problemas, ser capaz de analisar ideias de novas maneiras e, em um nível mais sofisticado, ser capaz de avaliar opiniões e ideias e fazer apreciações com base em evidências e critérios objetivos.

De acordo com Brown et al. (2018) colocar os novos conhecimentos em um contexto mais amplo, dando significado a ele e conectando-o a algo concreto que o aluno já sabe ajuda alicerçar a aprendizagem. Uma maneira de fazer isso é extrair ideias essenciais dos novos conteúdos e organizá-las em modelo mental conectando esse modelo aos conhecimentos prévios. O uso de práticas diferenciadas e intercaladas a fim de extrair “regras” ou princípios básicos que diferenciam os problemas é mais eficiente que a pura repetição.

A aprendizagem ocorre a partir de três etapas: a codificação, a consolidação e a recuperação (Brown et al., 2018). A codificação é o processo de converter percepções sensoriais em representações significativas no cérebro. A consolidação é o processo de fortalecimento das representações mentais para a memória de longo prazo, onde o cérebro reorganiza e estabiliza os vestígios de memória. Isso pode levar várias horas e envolve o processamento mais profundo dos novos conteúdos em que o cérebro dá significado e faz conexões com experiências passadas e outros conhecimentos de longo prazo armazenados na memória. A recuperação envolve a associação de pistas eficientes que permitem a recuperação da informação e do conhecimento.

A aprendizagem de novos conhecimentos ocorre a partir da conexão desses com o que já sabemos. Exposição espaçada e intercalada aos conteúdos e sua revisitação em diferentes momentos, auxilia na percepção e detecção de suas diferenças e semelhanças e conduz à representações mais complexas. Desta forma, a aprendizagem intercalada de conteúdos (Rohrer et al., 2007; Kantak et al. 2010) aumenta as habilidades de diferenciação e indução e constroem novas conexões permitindo a fixação dos novos conhecimentos na memória de longo prazo.

## MODELO DE FIXAÇÃO DA APRENDIZAGEM

Um modelo de fixação da aprendizagem, aplicado no presente estudo possui quatro dimensões: Compreender, Reproduzir, Aplicar e Avaliar. A dimensão *Compreender* é a primeira habilidade do domínio

cognitivo e inclui o entendimento do significado de algo novo, neste estudo os novos conceitos e conteúdos de programação são introduzidos de forma gradual, porém não linear. Para melhorar o desenvolvimento desta habilidade, são utilizadas, principalmente, as estratégias 2, 3 e 4 do quadro 1. *Reproduzir* consiste na consolidação dos conhecimentos a fim de torná-los recuperáveis a longo prazo. Neste estudo, consiste na proposição de soluções algorítmicas para problemas computacionais semelhantes ao que foi apresentado e explicado anteriormente. Para o desenvolvimento desta habilidade, é utilizada, em especial, estratégia número 4 do quadro 1. *Aplicar* é a terceira habilidade a ser desenvolvida e consiste em aplicar os conhecimentos construídos anteriormente para a resolução de problemas novos e mais complexos. Nesta dimensão são trabalhadas, particularmente, as estratégias 2, 4 e 5. A quarta habilidade consiste em *Avaliar* novos problemas computacionais em contextos diferentes dos já vistos e em maior nível de complexidade e propor melhor solução a partir da avaliação de diferentes possibilidades. Inclui, nesta categoria, a avaliação de soluções propostas por seus pares e a identificação de vantagens e desvantagens. As estratégias 1 e 5 são utilizadas para desenvolver esta habilidade.

QUADRO 1. Estratégias para aplicação do Modelo de Fixação da Aprendizagem.

	<b>Categoria</b>	<b>Estratégia utilizada e justificativa</b>	<b>Técnicas e Ferramentas utilizadas</b>
1.	Socialização de resultados	A socialização de soluções de problemas, incluindo decisões de trabalho, e resultados obtidos pelos alunos para novos problemas de programação, seguida de discussões entre alunos e professora com apresentação do raciocínio utilizado, permite conhecer a visão do outro e expandir as possibilidades de abstração e compreensão de uma situação/problema.	Apresentação oral e visual de resultados (assíncrona ou síncrona).
2.	Autoavaliação da aprendizagem em relação aos pares e a si.	Aplicação de testes semanais com correção automática que possibilitam ao aluno autoavaliação da aprendizagem dos conteúdos em relação aos conhecimentos já construídos anteriormente. Os testes são aplicados dois dias após a aula e possibilitam a autoavaliação a partir de informações armazenadas a curto prazo e auxilia na revisitação dos conteúdos a fim de favorecer a transposição para memória de longo prazo. Os testes avaliam a compreensão dos conteúdos e incluem estratégias de generalização e decomposição. Em todos os testes, são incluídas questões, uma ou duas, pessoais que possibilitam a autorreflexão do estudante sobre seu processo de aprendizagem na última semana, favorecendo a aprendizagem autorregulada (Pedrosa et al., 2019).	Ferramenta questionário do Moodle Institucional.
3.	Construção de Modelos Mentais	Ao final de cada aula, realizada semanalmente de forma remota, os alunos utilizam uma folha em branco, papel físico ou digital, e, por 10 minutos, registram, de forma escrita e/ou visual por meio de mapa mental, por exemplo, os conceitos aprendidos durante aquela aula. Neste exercício as ideias essenciais dos novos conteúdos estão organizadas e conectadas aos conhecimentos prévios. Este recurso também é utilizado ao final de cada módulo (conjunto de conteúdos previamente distribuídos) onde os alunos adicionam ao mapa mental da disciplina novas palavras-chaves, e reveem relações entre os demais módulos/conteúdos.	Mapa Mental
4.	Resolução de problemas computacionais	A resolução de problemas é fundamental para a fixação de conteúdos de programação e desenvolvimento do pensamento algorítmico e computacional. Tais exercícios exploram os níveis de compreensão, reprodução e aplicação dos conteúdos estudados.	Exercícios com feedback (individual ou times).
5.	Busca por soluções para novos problemas	Desafios com aumento progressivo do grau de complexidade e introdução de novos conteúdos para que os alunos busquem soluções antes dessa ser apresentada. Mesmo em caso de respostas erradas e interpretações equivocadas, o feedback corretivo aos alunos pode auxiliar na aprendizagem e na retenção mais duradoura de informações.	Aprendizagem baseada em problemas.

## METODOLOGIA

O presente trabalho consiste em um estudo de caso a partir de amostras pré-selecionadas e será conduzido a partir da aplicação do modelo apresentado na seção anterior, sendo aplicado em duas disciplinas de programação de computadores, uma disciplina do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas (ADS) e uma disciplina do Curso Técnico em Informática Integrado ao Ensino Médio. Ambas as disciplinas são divididas em 2 grupos de alunos, sendo que o modelo será aplicado apenas para o grupo 1 de cada disciplina, grupo de controle, metade da turma. Os demais alunos, grupo 2, cursarão as disciplinas sem aplicação do modelo.

Serão aplicados instrumentos próprios de avaliação do pensamento computacional, elaborados pelos pesquisadores, e teste de avaliação do pensamento criativo (Torrance, 1966) em três momentos do curso e em ambas as amostras (os 2 grupos de cada disciplina). As evidências coletadas servirão para determinar possíveis relações no desenvolvimento dos alunos dos grupos de controle em relação aos demais alunos das duas disciplinas em que o modelo não é aplicado.

Para análise dos resultados obtidos, consideram-se diferenças associadas ao ambiente e contexto dos participantes da pesquisa, incluindo conhecimento prévio de codificação, afinidade com tecnologia, situação econômica, disponibilidade e acesso às tecnologias fora do ambiente escolar (computador, banda larga).

## CONCLUSÕES

Este trabalho encontra-se em andamento e, portanto, não é possível apresentar resultados da pesquisa, os quais serão obtidos ao final do estudo. Todavia, considera-se trivial socializar a pesquisa em andamento com a comunidade acadêmica e científica dada a importância e atualidade do tema estudado e as diversas questões abertas sobre o assunto, acredita-se que este trabalho tem potencial para contribuir com futuras pesquisas e oferecer evidências importantes para o desenvolvimento de novas ferramentas e *frameworks* para apoio ao desenvolvimento do pensamento computacional e da criatividade computacional.

## REFERÊNCIAS

- Amabile, T. M., and Pillemer, J. **Perspectives on the Social Psychology of Creativity**. Journal of Creative Behavior 46, no. 1 (2012): 3–15.
- Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 **Computational Thinking Curriculum Framework- Implications for Teacher Knowledge**. Educational Technology & Society, 19(3), 47–57.
- Barr, D., Harrison, J., & Conery, L. (2011). **Computational Thinking: A Digital Age Skill for Everyone**. Learning & Leading with Technology, 38(6), 20–23.
- Batista, Riann & Horta, Euler & Fonseca, Alexandre. (2020). **Programação em Blocos: impacto de um projeto de extensão executado em Escolas públicas de Diamantina/MG**. Informática na educação: teoria & prática. 23. 10.22456/1982-1654.96741.
- Bloom, B. S. (1956). **Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain**. New York: David McKay Co.
- Bocconi, S. et al. (2016). **Developing Computational Thinking in Compulsory Education**. Disponível em: <<https://ec.europa.eu/jrc/en/publication/eur-scientific-and-technical-research-reports/developing-computational-thinking-compulsory-education-implications-policy-and-practice>>. Acesso em: dez. de 2020.
- BRASIL. **Ministério da Educação. Base Nacional Comum Curricular**. Brasília, 2018.
- Amabile, T. M., and Pillemer, J. "Perspectives on the Social Psychology of Creativity." Journal of Creative Behavior 46, no. 1 (2012): 3–15.
- Brown, P. C.; Roediger III, H. L.; McDaniel, M. A. **Fixe o conhecimento: a ciência da aprendizagem bem sucedida**. Tradução: Henrique de Oliveira Guerra; revisão técnica: Claudio de Moura Castro. – Porto Alegre: Penso, 2018.
- Grover, S., & Pea, R. (2013). **Computational Thinking in K–12 A Review of the State of the Field**. Educational Researcher, 42(1), 38–43.
- Israel-Fishelson R., Hershkovitz A. (2021) **Studying interrelations of computational thinking and creativity: A scoping review (2011–2020)**. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S036013152100230X>>. Acesso em: out. de 2021.
- Israel-Fishelson R., Hershkovitz A., Eguíluz A., Garaizar P., Guenaga M. (2021). **The Associations Between Computational Thinking and Creativity: The Role of Personal Characteristics**. Journal of Educational Computing. 2021; 58(8):1415-1447.
- Kantak S. S. et al., **Neural substrates of motor memory consolidation depend on practice structure**. NAture Neuroscience, v. 13, n. 8, p. 923-925, 2010.
- Lee, I.; Denner, J.; Martin, F. L.; Coulter, B.; Allan, W.; Erickson, J.; Malyn-Smith, J.; Werner, L. (2011). **Computational thinking for youth in practice**. ACM Inroads, 2(1), 32–37.
- Pedrosa D., Cravino J., Morgado L., Barreira C. (2019) **Co-regulated Learning in Computer Programming: Students Co-reflection About Learning Strategies Adopted During an Assignment**. In: Tsitouridou M., A. Diniz J., Mikropoulos T. (eds) Technology and Innovation in Learning, Teaching and Education. TECH-EDU. Communications in Computer and Information Science, vol 993. Springer, Cham.