

Relações entre o Pensamento Computacional e a Engenharia de Software: Uma Revisão Sistemática da Literatura*

Júlia de Avila dos Santos¹, Simone André da Costa Cavalheiro¹,
Luciana Foss¹, Leomar S. da Rosa Jr.¹

¹Programa de Pós Graduação em Computação
Centro de Desenvolvimento Tecnológico
Universidade Federal de Pelotas (UFPEL)
Rua Gomes Carneiro, 1 - CEP 96010-610 - Pelotas - RS - Brasil

{julia.asantos, simone.costa, lfoss, leomarjr}@inf.ufpel.edu.br

Abstract. *Computational Thinking and Software Engineering are areas that include methods and techniques aimed at solving problems. This article presents a systematic review to investigate how these two areas have been integrated. In total, 16 articles were analyzed in order to assess the goals of the relationships that are being established, and the theoretical foundations adopted. As a result, it is observed that most works highlight Computational Thinking as an important skill for the software development process. In turn, Software Engineering techniques and processes have been adopted for the development and evaluation of Computational Thinking.*

Resumo. *Pensamento Computacional e Engenharia de Software são áreas que incluem métodos e técnicas voltados a resolução de problemas. Este artigo apresenta uma revisão sistemática para investigar de que forma essas duas áreas vem sendo integradas. No total, 16 artigos foram analisados com a finalidade de avaliar os objetivos das relações que estão sendo estabelecidas e os fundamentos teóricos adotados. Como resultado, observa-se que a maioria dos trabalhos destacam o Pensamento Computacional como habilidade importante para o processo de desenvolvimento de software. Por sua vez, técnicas e processos de Engenharia de Software têm sido adotados para desenvolvimento e avaliação do Pensamento Computacional.*

1. Introdução

Pensamento Computacional (PC) é o processo de resolução de problemas, baseado em conceitos e técnicas que são fundamentadas na Computação. Wing afirma [Wing 2006] que o PC não é apenas valioso para cientistas da computação, mas é uma metodologia útil para todos.

Não há um consenso sobre a definição do PC e suas habilidades na literatura [Guarda and Pinto 2020]. Em particular, a Associação de Professores de

*O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001, do MCTIC/CNPq (Rede Sacci), da SMED/Pelotas e da PREC e PRPPG / UFPEL.

Ciência da Computação e a Sociedade Internacional de Tecnologia em Educação [ISTE and CSTA 2011] definem o PC como um processo de resolução de problemas que inclui (mas não está limitado a) as seguintes características: formular problemas de uma forma que nos permita usar computadores e outras ferramentas para ajudar a resolvê-los; organizar e analisar logicamente os dados; representar dados por meio de abstrações, como modelos e simulações; automatizar soluções por meio do pensamento algorítmico; identificar, analisar e implementar soluções possíveis com o objetivo de alcançar a combinação mais eficiente e eficaz de etapas e recursos; e generalizar e transferir este processo de resolução de problemas para uma ampla variedade de problemas.

De uma forma mais específica, outros autores apresentam técnicas fundamentais ao desenvolver o PC [Bordini et al. 2020, Souto et al. 2018, Bordini et al. 2017]. Em [BBC 2018] são destacadas quatro técnicas principais: (1) decomposição de um problema em partes menores; (2) reconhecimento de padrões por semelhanças entre e dentro dos problemas; (3) abstração de detalhes irrelevantes, focando apenas nas informações importantes; e (4) construção de algoritmos para o desenvolvimento de soluções.

Wing [Wing 2006] afirma que pensar como um cientista da computação significa mais do que ser capaz de programar um computador. Requer pensar em vários níveis de abstração, bem como utilizar as técnicas fundamentadas na Ciência da Computação. Neste sentido, o PC também está relacionado ao pensamento de engenharia, visto que a Computação preocupa-se com a definição de soluções algorítmicas (sistemas) que devem interagir com o mundo real.

Especialmente, a Engenharia de Software (ES) trabalha com diferentes aspectos do desenvolvimento de software, focada em uma solução adequada e de qualidade. Pressman et al. [Pressman and Maxim 2016] afirma que ES abrange processos, métodos e ferramentas que possibilitam o desenvolvimento de software de altíssima qualidade. Os processos de software estão relacionados a cinco atividades: comunicação, planejamento, modelagem, construção e entrega. Já os métodos de engenharia de software envolvem: comunicação, análise de requisitos, modelagem de projeto, construção de programa, testes e suporte [Pressman and Maxim 2016].

Visto que a ES sistematiza o processo de desenvolvimento de software e o PC é uma metodologia para solucionar problemas fundamentada na Computação, ao buscar estratégias que fundamentem esse processo de resolver problemas, faz sentido analisar técnicas já consolidadas da ES e tentar relacioná-las com o PC. Deste modo, este trabalho visa identificar as relações que estão sendo estabelecidas entre essas duas áreas, onde realize-se uma investigação para analisar os conceitos/fundamentos de ES e do PC que estão sendo considerados, bem como constatar com quais objetivos as áreas estão sendo relacionadas.

Este artigo está organizado como segue. Na Seção 2 aborda-se a metodologia de pesquisa, apresentando as etapas de planejamento e execução. Na Seção 3 apresenta-se os resultados e discussões. A conclusão é descrita na Seção 4.

2. Metodologia de Pesquisa

Esta pesquisa seguiu uma metodologia de pesquisa chamada Revisão Sistemática de Literatura (RSL). De acordo com [Kitchenham and Charters 2007] uma RSL “é um meio de

identificar, avaliar e interpretar todas as pesquisas disponíveis relevantes para uma determinada questão de pesquisa, ou tópico de área, ou fenômeno de interesse”.

Existem três fases principais para uma RSL: planejamento, condução e relatório [Kitchenham and Charters 2007]. A primeira etapa está relacionada com a pré-revisão. Tem como objetivo abordar a necessidade da RSL, definindo procedimentos básicos e essenciais da revisão. Na etapa de condução é realizada a seleção dos artigos, de acordo com os critérios estabelecidos na etapa anterior. Nesta etapa é realizada a extração e síntese dos dados, com o intuito de responder as questões de pesquisa. A última etapa é a divulgação dos resultados da revisão, sendo apresentados de acordo com o objetivo da RSL.

2.1. Planejamento

Na etapa de planejamento são definidos os objetivos da RSL, as questões de pesquisa, as estratégias de busca, os repositórios e a definição dos critérios de inclusão e exclusão dos artigos.

O objetivo desta pesquisa é o de investigar as formas de integração das áreas do PC e da ES, bem como os fundamentos considerados neste contexto. Sendo importante ressaltar, que não foi encontrada nenhuma outra RSL que realiza investigações semelhantes. As questões de pesquisas foram elaboradas de acordo com o objetivo do trabalho a ser desenvolvido. Deste modo, foram elaboradas quatro questões de pesquisa:

- **QP1:** Com qual(is) objetivo(s) as relações entre ES e PC foram estabelecidas?
- **QP2:** Quais são os fundamentos da ES (referencial teórico) que estão sendo considerados nos trabalhos?
- **QP3:** Quais conceitos ou habilidades do PC (referencial teórico) estão sendo considerados nos trabalhos?
- **QP4:** Quais são as relações estabelecidas entre a ES e o PC?

As buscas dos artigos foram realizadas nos meses de maio e junho de 2021. Os repositórios escolhidos foram ACM Transactions on Computing Education, Institute of Electrical and Electronics Engineers (IEEE), Microsoft Academic e ScienceDirect. A string de busca estabelecida para ser encontrada nos resumos dos artigos foi (“software development” OU “software engineering”) E “computational thinking”.

Os critérios de inclusão e exclusão considerados para avaliar se os artigos estão relacionados com o objetivo da RSL e com as questões de pesquisa estão descritos na Tabela 1.

Tabela 1. Critérios de inclusão e exclusão de artigos

Critérios de Inclusão	Critérios de Exclusão
1. Trabalhos relacionados ao objetivo dessa RSL.	1. Revisões sistemáticas e mapeamentos de estudos.
2. Trabalhos que permitem identificar relações entre conceitos de ES/desenvolvimento de software e PC.	2. Trabalhos já selecionados por outra base.
	3. Versões anteriores de trabalhos selecionados.
	4. Pôster e resumos.
	5. Trabalhos sem acesso livre.
	6. Artigos que não estejam em Inglês ou Português.

2.2. Condução

Na etapa de condução aplicou-se o plano elaborado na etapa anterior, ou seja, nessa etapa foram selecionados os artigos a serem analisados. Esses artigos foram selecionados nos repositórios considerados, pela identificação da string de busca no resumo, aplicando os critérios de inclusão e exclusão. Logo após as análises desses artigos, foi possível responder as questões de pesquisa.

A condução ocorreu em duas etapas: (1) após o resultado das buscas fez-se a leitura do título e do resumo, selecionando-se os artigos que iriam para próxima etapa; (2) fez-se a leitura dos artigos selecionados, dando foco à introdução e conclusão. Os critérios de inclusão e exclusão foram aplicados nas duas etapas.

A Tabela 2 sumariza esta fase de condução da pesquisa, apresentando a quantidade de artigos selecionados em cada uma das etapas para cada um dos repositórios considerados, bem como totaliza os resultados obtidos.

Tabela 2. Quantidade de artigos encontrados no passo de triagem

Repositório	String de Busca	1ª seleção	2ª seleção
<i>Microsoft Academic</i>	212	6	4
<i>ACM</i>	18	6	4
<i>ScienceDirect</i>	2	2	2
<i>IEEE</i>	12	7	6
Total	244	21	16

Aplicando a string de busca retornaram 244 artigos. Na primeira etapa 21 artigos foram selecionados, dos quais restaram 16 após a segunda etapa. Na Tabela 3 estão listados os trabalhos selecionados para análise e discussão desta RSL.

Tabela 3. Artigos selecionados para a RSL

Repositório	Identificação	Artigo
<i>Microsoft Academic</i>	01	[Higgins et al. 2017]
	02	[Apiola and Sutinen 2020]
	03	[Kwon and Kim 2018]
	04	[Corral and Fronza 2019]
<i>ACM</i>	05	[Trimble and Keeling 2013]
	06	[Fronza et al. 2016]
	07	[Fronza et al. 2017]
	08	[Fronza et al. 2015]
<i>ScienceDirect</i>	09	[Ciancarini et al. 2019]
	10	[Monteiro et al. 2017]
<i>IEEE</i>	11	[Deng et al. 2009]
	12	[Hurson and Sedigh 2010]
	13	[Wang and Wang 2019]
	14	[Lester 2008]
	15	[Moreno-León et al. 2016]
	16	[Dong and Jia 2020]

3. Resultados e Discussão

Os resultados são apresentados de acordo com as questões de pesquisa estabelecidas na etapa de planejamento. As subseções que seguem descrevem os resultados encontrados.

3.1. Objetivos das relações estabelecidas entre ES e PC

Respondendo a QP1, foi possível categorizar os artigos em quatro grupos principais, conforme as integrações estabelecidas entre as áreas, diferenciando-se aqueles que: utilizam conceitos/técnicas do PC no processo de desenvolvimento de software (PCPDS); aplicam a metodologia do PC no ensino de ES (PCEES); adotam processos de ES para avaliar o PC (ESAPC); aplicam técnicas de ES no ensino do PC (ESEPC). Na Tabela 4 é possível identificar quais artigos correspondem a cada uma das categorias.

Os artigos nas categorias PCPDS e PCEES consideram o PC como uma metodologia importante para a resolver problemas. Tendo em vista que a ES e o processo de desenvolvimento de software buscam a solução de problemas, o PC pode auxiliar no desenvolvimento dessas soluções. De forma similar, os artigos na categoria ESEPC, se utilizam de conceitos da ES para promover o desenvolvimento do PC.

Os objetivos dos trabalhos categorizados em PCPDS são: propor frameworks para o processo (01) e ensino (09) de desenvolvimento de software integrando conceitos do PC; analisar o impacto do PC no ensino de desenvolvimento de software (03 e 08); estudar a dinâmica socioeconômica do processo de desenvolvimento de software do ponto de vista do PC (05).

Por sua vez, a maioria dos artigos da categoria PCEES (11, 12, 13, 14 e 16) sugerem diferentes propostas de reformas no ensino de ES, acrescentando diferentes abordagens do PC. Há também, nesta categoria, uma proposta de integração de conceitos do PC, da ES e competências STEM/STEAM em uma estrutura para resolução de problemas (02), assim como um trabalho que objetiva analisar a qualidade final de produtos gerados pela aquisição do PC guiado por conceitos da ES (10).

Já os artigos na categoria ESEPC fazem uso de pensamento visual (07) e métodos ágeis (06) – ambos fundamentos de ES – para o desenvolvimento de habilidades do PC.

Por fim, os artigos na categoria ESAPC utilizaram métricas de ES para realizar a avaliação do PC. As métricas foram utilizadas para avaliar a qualidade interna, externa e de uso (04) e a complexidade (05) de sistemas de software.

3.2. Fundamentos da ES

Na QP2 foram investigados os fundamentos (conceitos e técnicas) da ES considerados nos trabalhos. A Tabela 4 identifica conceitos/técnicas da ES abordados em cada um dos trabalhos.

Foi possível identificar que, devido a diversidade dos aspectos da ES abordados nos trabalhos analisados, os mesmos não adotaram o mesmo referencial teórico, não sendo possível destacar uma referência em particular.

Apesar dos artigos abordarem diferentes fundamentos de ES, é possível destacar as etapas do processo de desenvolvimento de software que apareceram com maior frequência: Projeto/design e desenvolvimento (7 ocorrências); Definição de requisitos (6 ocorrências); e Teste/avaliação e validação (6 ocorrências). O trabalho cooperativo também é relacionado em quatro artigos, sendo considerado importante para encontrar soluções mais adequadas e eficazes, conferindo qualidade ao produto final.

É importante salientar, que os fundamentos de ES que mais tiveram destaque estão

Tabela 4. Fundamentos da ES considerados nos trabalhos

	PCPDS					PCEES								ESAPC		ESEPC		Total
	01	03	05	08	09	02	10	11	12	13	14	16	04	15	06	07		
Projeto/design e desenvolvimento				x		x	x	x	x		x	x					7	
Definição de requisitos	x	x		x		x	x					x					6	
Teste/avaliação e validação				x		x	x	x	x							x	6	
Trabalho/ Pensamento cooperativo			x		x							x				x	4	
Manutenção/ Depuração							x	x	x								3	
Compreensão do problema/ viabilidade		x		x													2	
Integração				x												x	2	
Reutilização							x		x								2	
Orçamento e planejamento			x									x					2	
Método de desenvolvimento OO	x											x					2	
Software em grande escala								x				x					2	
Métricas de avaliação													x	x			2	
Métodos/Valores ágeis					x											x	2	
Uso de ferramentas e metodologias			x									x					2	
Modelo de processo de software		x										x					2	
Gerenciamento de risco								x									1	
Tolerância a falhas									x								1	
Gerenciamento de memória								x									1	
Pensamento de Engenharia										x							1	
Controle de fluxo não sequencial	x																1	
Modularidade	x																1	
Engenharia Semiótica							x										1	
Pensamento Visual															x		1	

relacionados com as atividades de processos de software: comunicação, requisitos, modelagem, construção, teste e suporte [Pressman and Maxim 2016].

3.3. Conceitos/habilidades do PC

A Tabela 5 apresenta os conceitos e habilidades do PC que estão sendo considerados nos trabalhos (QP3). Pode-se perceber que os mais discutidos são aqueles que estão relacionados com os quatro pilares do PC [BBC 2018]: algoritmos, abstração, decomposição e

reconhecimento de padrões. Outros conceitos, como loops, eventos, condicionais podem ser relacionados ao desenvolvimento de algoritmos, mesmo sendo abordados separadamente nos trabalhos. Os conceitos representação de dados, coleta de dados e paralelismo também são abordados com frequência, geralmente sendo considerados com relação ao projeto de sistemas mais complexos.

Neste levantamento, pode-se perceber as diferentes conceituações relacionadas ao termo PC. Enquanto alguns trabalhos consideram a resolução de problemas como o objeto fim do PC, outros incluem resolver problemas como uma das habilidades do PC.

3.4. Relações estabelecidas entre a ES e o PC

Os artigos apresentam diferentes relações entre os fundamentos da ES e/ou processo de desenvolvimento de software e os conceitos ou habilidades do PC (QP4). Alguns trabalhos relacionam habilidades do PC diretamente com conceitos da ES, e outros apenas abordam alguns aspectos mais gerais, como desenvolver o PC para uma melhor resolução de problemas no processo de desenvolvimento de software.

O artigo 01 relaciona abstração com todas as fases do desenvolvimento de software, desde a análise do problema até a concepção de uma solução e codificação. Representação de dados é associado a identificação dos dados necessários a serem armazenados nos programas. Decomposição está relacionado à divisão dos sistemas em módulos menores. Avaliação de soluções relacionadas a testes para correção, consistência e depuração de erros, de acordo com características de qualidade de software. Reconhecimento de padrões é associado à reutilização, não apenas no nível do código, mas também de artefatos de análise e design. Construção de algoritmos está associada ao uso de fluxogramas e pseudo-código, deste modo, focando em uma solução de alto nível, sem se preocupar com a linguagem de programação que será usada para implementar a solução final.

Os artigos 02, 11, 12, 13 e 14 referem-se ao desenvolvimento do PC como método para auxiliar a compreensão de soluções de problemas complexos no processo de desenvolvimento de software, conseqüentemente integrando o PC na projeção de soluções em conjunto com a ES. Em particular, o artigo 11 associa a integração das áreas como forma de desenvolver sistemas inteligentes, com gerenciamento, método e design eficientes. O artigo 13 apresenta também o conceito de Pensamento de Engenharia, abordado como um guia para tornar o desenvolvimento de software mais rápido e eficiente, sendo aplicado integrado ao PC.

O artigo 03 observa que a codificação de software está relacionada com o raciocínio lógico e a resolução de problemas; e, conseqüentemente, está diretamente relacionada com algoritmos. Além disso, associa coleta, análise e representação de dados com a etapa de entendimento do problema. Decomposição de problemas, abstração, algoritmos e procedimentos são associados à etapa de geração de ideias. Automação, simulação, paralelização são considerados na etapa da melhor escolha para o modelo de processo de software.

Os artigos 04 e 15 utilizam métricas de avaliação (qualidade e complexidade) de ES para analisar a aquisição do PC. Já, o artigo 05 apresenta relações de mais de alto nível, apresentado o conceito de decomposição (por tamanho, por tipo, por força de trabalho) diretamente relacionado com a produção de software. Também destaca que as diferentes

Tabela 5. Conceitos/habilidades do PC considerados nos trabalhos

	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	Total
Abstração	x		x	x	x		x	x		x			x	x	x	x	11
Pensamento Algorítmico	x		x	x	x	x	x	x	x							x	9
Reconhecimento de Padrões				x		x	x	x	x		x	x					7
Decomposição	x		x		x			x	x				x	x			7
Resolução de problemas			x							x	x	x	x	x		x	7
Paralelismo			x	x	x	x	x	x							x		7
Representação de Dados	x		x	x			x	x							x		6
Loops/recursividade				x	x	x	x							x		x	6
Raciocínio Lógico			x						x						x	x	4
Avaliação de solução	x								x							x	3
Simulação/Automação			x		x			x									3
Testes e depuração				x		x	x										3
Interações				x			x								x		3
Coleta de Dados			x					x									2
Eventos				x			x										2
Organização										x						x	2
Condicionais				x			x										2
Operadores				x			x										2
Modelagem/fluxograma		x														x	2
Reutilização				x			x										2
Generalização									x								1
Pensamento Sistêmico																x	1
Sincronização															x		1
Controle de fluxo															x		1
Viabilidade Mínima																x	1
Programação OO																x	1
Distribuição																x	1

metodologias de desenvolvimento de software, que empregam vários modelos, podem ser vistas como algoritmos.

O artigo 06 utiliza a abordagem de pensamento visual integrada ao PC para auxiliar no desenvolvimento de um aplicativo. O método do pensamento visual assemelha-se ao uso de diagramas para facilitar as visualizações e as trocas de ideias, que levam para a fase de desenvolvimento, fase que conta com o suporte do PC.

Tendo em vista que um indivíduo que domina habilidades do PC não é necessariamente capaz de cooperar com outros indivíduos para resolver problemas complexos, os artigos 07 e 09 relacionam a prática de métodos/valores ágeis nas atividades do PC. Desta maneira, trabalham com o desenvolvimento do PC e a cooperação para auxiliar na solução de problemas mais complexos.

O artigo 08 relaciona diretamente as fases de desenvolvimento de software com as habilidades do PC: Viabilidade com coleta de dados; Análise com coleta, análise e representação de dados e com decomposição de problemas; Design com análise e representação de dados, abstração, algoritmos e procedimentos; Desenvolvimento com representação de dados, algoritmos e procedimentos, automação e paralelização; Testes com simulação; e Integração com análise de dados, simulação e paralelização.

O artigo 10 trabalha com a promoção do PC no desenvolvimento de jogos, guiado pelos princípios da Engenharia Semiótica e ES, analisando a qualidade do produto final e a aquisição do PC.

O artigo 16 relaciona pontos de conhecimento do PC com conteúdo em um curso de ES: solução de problemas com definição de ES; fluxograma com ferramentas de diagrama; pensamento lógico e modelagem com o modelo em cascata; princípio da viabilidade mínima com especificação de requisitos e protótipo de projeto; algoritmo de recursão com desenvolvimento incremental; programação orientada a objetos com método de desenvolvimento orientado a processos e orientado a objetos; abstração, modo de design, modelagem e pensamento sistemático com design de software; sistema de distribuição com desenvolvimento de software em grande escala; organização de código com implementação de software; e solução ótima com orçamento e plano do projeto.

3.5. Considerações finais

Realizou-se também análise dos principais trabalhos incluídos no referencial teórico dos artigos considerados. A autora Jeannette Wing teve destaque com seus artigos [Wing 2006], [Wing 2014] e [Wing 2008] citados, respectivamente, em 56,25%, 18,75% e 18,75% dos trabalhos. Também foram referenciados mais de uma vez os artigos [Brennan and Resnick 2012] e [Barr et al. 2011], aparecendo respectivamente em 25% e 12,5% dos trabalhos.

4. Conclusão

PC é uma metodologia para solucionar problemas que engloba diferentes conceitos e técnicas da Computação. Por sua vez, a ES integra processos, métodos e ferramentas para o desenvolvimento de software, tendo como propósito a qualidade e eficácia do produto final. Neste contexto, pode-se dizer que o PC e a ES trabalhados de maneira conjunta têm o potencial de desenvolver competências para encontrar melhores soluções para problemas reais e complexos. Neste sentido, este trabalho investigou as relações que estão sendo estabelecidas entre as áreas da ES e do PC. Foram analisados os objetivos dessas interações, bem como os fundamentos da ES e conceitos/habilidades do PC que estão sendo adotados.

A metodologia de pesquisa deste trabalho foi baseada na revisão sistemática da literatura [Kitchenham and Charters 2007]. Ao todo 16 artigos foram analisados para as discussões. Como resultado, pôde-se perceber que a maioria dos artigos integrou as áreas

no contexto educacional, utilizando o PC para auxiliar no ensino de ES, como também no processo de desenvolvimento de software. Alguns trabalhos utilizaram conceitos da ES para auxiliar no ensino e desenvolvimento do PC. Apenas dois artigos abordaram a ES para avaliar a aquisição do PC.

Pode-se concluir que o PC e a ES estão sendo desenvolvidos de forma integrada para desenvolver competências referente ao processo de resolução de problemas. Observa-se que a comunidade acadêmica já consegue estabelecer relações entre as áreas e explorá-las em conjunto. Em particular, alguns trabalhos estabelecem de forma direta a relação entre os fundamentos da ES com habilidades do PC, para serem desenvolvidos de maneira simultânea.

Em relação a quais fundamentos de ES estão sendo mais abordados nos trabalhos, apareceram diversos aspectos, devido a diferentes referências adotadas para a área de ES. Entretanto, os que mais tiveram destaque referem-se as atividades que devem ser consideradas em um processo de desenvolvimento de software [Pressman and Maxim 2016]. O PC também abrange diversos conceitos/habilidades a serem desenvolvidos, entretanto pode-se perceber que as técnicas de decomposição, reconhecimento de padrões, abstração e algoritmos tiveram destaque.

Essa investigação evidenciou que os métodos e processos para a elaboração de projetos de desenvolvimento de software podem servir como inspiração para a proposta de uma metodologia de desenvolvimento de projeto com fundamentação no PC, a qual pretende-se realizar como trabalho futuro.

Referências

- Apiola, M. and Sutinen, E. (2020). Design science research for learning software engineering and computational thinking: Four cases. *Computer Applications in Engineering Education*, pages 83–101.
- Barr, D., Harrison, J., and Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology*, 38(6):20–23.
- BBC (2018). Introduction to computational thinking. <https://www.bbc.com/bitesize/guides/zp92mp3/revision/>.
- Bordini, A., Avila, C., Marques, M., Foss, L., and Cavalheiro, S. (2017). Pensamento computacional nos ensinamentos fundamental e médio: uma revisão sistemática. In *Anais do Simpósio Brasileiro de Informática na Educação*, pages 123–132, Porto Alegre, RS, Brasil. SBC.
- Bordini, A., Foss, L., and Cavalheiro, S. (2020). Metodologia da decomposição e composição para resolução de problemas em LiVE. In *Anais do Simpósio Brasileiro de Informática na Educação*, pages 1453–1462, Porto Alegre, RS, Brasil. SBC.
- Brennan, K. and Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the American Educational Research Association*, pages 1–25.
- Ciancarini, P., Missiroli, M., and Russo, D. (2019). Cooperative thinking: Analyzing a new framework for software engineering education. *Journal of Systems and Software*, 157(2019):1–12.

- Corral, L. and Fronza, I. (2019). A strategy for assessing the acquisition of computational thinking competences: A software engineering approach. In Fronza, I. and Pahl, C., editors, *Proceedings of the Systems of Assessments for Computational Thinking Learning Workshop*, volume 2434 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- Deng, Z., Huang, W., Dong, R., and Wen, P. (2009). Exploration of ability development of engineering and computational thinking skills in software engineering majors. In *International Conference on Computer Science & Education*, pages 1665–1668.
- Dong, X. and Jia, J. (2020). Teaching reform of software engineering course based on computational thinking. In *International Conference on Artificial Intelligence and Computer Engineering*, pages 399–402.
- Fronza, I., El Ioini, N., and Corral, L. (2015). Students want to create apps: leveraging computational thinking to teach mobile software development. In *Proceedings of the Conference on Information Technology Education*, pages 21–26.
- Fronza, I., El Ioini, N., and Corral, L. (2016). Teaching software design engineering across the K-12 curriculum: Using visual thinking and computational thinking. In *Proceedings of the Conference on Information Technology Education*, pages 97–101.
- Fronza, I., Ioini, N. E., and Corral, L. (2017). Teaching computational thinking using agile software engineering methods: A framework for middle schools. *ACM Transactions on Computing Education*, 17(4):1–28.
- Guarda, G. and Pinto, S. (2020). Dimensões do pensamento computacional: conceitos, práticas e novas perspectivas. In *Anais do Simpósio Brasileiro de Informática na Educação*, pages 1463–1472, Porto Alegre, RS, Brasil. SBC.
- Higgins, C., O’Leary, C., Hanratty, O., and Mtenzi, F. (2017). A conceptual framework for a software development process based on computational thinking. In *Proceedings of the International Technology, Education and Development Conference*, pages 455–464. Technological University Dublin.
- Hurson, A. R. and Sedigh, S. (2010). Transforming the instruction of introductory computing to engineering students. In *Transforming Engineering Education: Creating Interdisciplinary Skills for Complex Global Environments*, pages 1–19.
- ISTE and CSTA (2011). Computational thinking teacher resources. Second Edition. Disponível em https://cdn.iste.org/www-root/2020-10/ISTE_CT_Teacher_Resources_2ed.pdf. Acesso em: julho de 2021.
- Kitchenham, B. A. and Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE-2007-01, School of Computer Science and Mathematics, Keele University.
- Kwon, J. and Kim, J. (2018). A study on the design and effect of computational thinking and software education. *KSII Transactions on Internet and Information Systems*, 12(8):4057–4071.
- Lester, C. Y. (2008). An innovative approach to teaching an undergraduate software engineering course. In *International Conference on Software Engineering Advances*, pages 301–306.

- Monteiro, I. T., de Castro Salgado, L. C., Mota, M. P., Sampaio, A. L., and de Souza, C. S. (2017). Signifying software engineering to computational thinking learners with agentsheets and polifacets. *Journal of Visual Languages & Computing*, 40:91–112.
- Moreno-León, J., Robles, G., and Román-González, M. (2016). Comparing computational thinking development assessment scores with software complexity metrics. In *Global Engineering Education Conference*, pages 1040–1045.
- Pressman, R. and Maxim, B. (2016). *Engenharia de Software*. McGraw Hill Brasil, 8 edition.
- Souto, A. L., Araujo, O., Andrade, W., Guerrero, D., Melo, M., and de Souza, I. M. L. (2018). Análise de rede na identificação de habilidades relacionadas ao pensamento computacional. In *Anais do Simpósio Brasileiro de Informática na Educação*, pages 655–664.
- Trimble, J. and Keeling, H. (2013). Comparative paradigms in the examination of software production. In *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference*, pages 341–346.
- Wang, F. and Wang, H. (2019). Analysis of the curriculum reform method based on the combination of computational thinking and engineering thinking. In *International Conference on Information Technology in Medicine and Education*, pages 284–287.
- 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. (2014). Computational thinking benefits society. 40th Anniversary Blog of Social Issues in Computing. Disponível em <http://socialissues.cs.toronto.edu/index.html%3Fp=279.html>. Acesso em: julho de 2021.