

## Ferramentas para o Ensino-Aprendizagem do Pensamento Computacional: onde está Alan Turing?

James Roberto Bombasar<sup>1</sup>, Rafael de Santiago<sup>2</sup>  
Elisângela Maschio de Miranda<sup>2</sup>, André Luís Alice Raabe<sup>3,4</sup>

<sup>1</sup>Setor de Pesquisa e Desenvolvimento  
Faculdade Avantis – Balneário Camboriú, SC – Brasil

<sup>2</sup>Bacharelado em Ciência da Computação

<sup>3</sup>Mestrado em Computação Aplicada

<sup>4</sup>Programa de Pós-graduação em Educação

Universidade do Vale do Itajaí (UNIVALI) – Itajaí, SC – Brasil

james.bombasar@avantis.edu.br, {rsantiago,elis,raabe}@univali.br

**Abstract.** *Since 2006, when “computational thinking” term was popularized by Jeannette Wing as a new basic skill for the twenty-first century, teaching and learning of computational thinking in K-12 education has been the focus of much research in the scientific community. This research presents a systematic review of the literature on tools used for teaching computational thinking from 2006 through 2015, in order to contribute to gaps identifying. In the results, we have not found any tool that implements Alan Turing theories and formalisms.*

**Resumo.** *Desde o ano de 2006, quando Jeannette Wing popularizou o termo “computational thinking” como sendo uma habilidade essencial para o século XXI, o ensino-aprendizagem do pensamento computacional tem sido o foco de várias pesquisas no meio científico, principalmente na educação básica. Este trabalho apresenta uma revisão sistemática da literatura com o objetivo de identificar as principais ferramentas que foram utilizadas no ensino do pensamento computacional de 2006 até o início de 2015, e assim contribuir para a identificação de eventuais lacunas que possam ser preenchidas. Dentre os resultados, não foi identificada nenhuma ferramenta baseada nas teorias e formalismos desenvolvidos por Alan Turing, o pai da computação moderna.*

### 1. Introdução

Em 2006, Jeannette Wing despertou a atenção da comunidade científica quando propôs a equiparação do pensamento computacional (do inglês, *computational thinking*) às habilidades básicas de leitura, escrita e aritmética. Basicamente, Wing (2006) descreveu o pensamento computacional (PC) como sendo a habilidade de empregar de técnicas e conceitos da ciência da computação para resolver problemas do cotidiano, tais como colocar na mala os itens necessários para uma viagem (busca e *cache*), decidir entre comprar um carro e andar de táxi (algoritmo online) e refazer os passos para localizar um objeto perdido (*backtracking*).

O interesse da comunidade científica e dos governos de diversos países pelo ensino-aprendizagem do PC a partir da educação básica já está bem documentado em Groover e Pea (2013) e Mannila et al. (2014). Ao mesmo tempo que grande parte da pesquisa [Andrade et al. 2013; Ribeiro et al. 2013; Carvalho et al. 2013; Roscoe et al. 2014; Campos et al. 2014; França e Tedesco 2014; Hill 2014] encontra-se focada em metodologias e ferramentas para o ensino-aprendizagem do PC, outra [França e Amaral 2013; Gouws et al. 2013; Koh et al. 2014] busca instrumentos para avaliar os resultados da aprendizagem.

Este trabalho insere-se no primeiro contexto de pesquisa, e apresenta como principal contribuição uma revisão sistemática da literatura, ou SLR (*Systematic Literature Review*), com o objetivo de identificar as principais ferramentas que foram utilizadas no ensino-aprendizagem do PC de 2006 até o início de 2015, e com isto favorecer a identificação de eventuais lacunas que possam ser preenchidas. Nos resultados da SLR, não foi identificada nenhuma ferramenta inspirada nas teorias e formalismos desenvolvidos por Alan Turing, o pai da computação moderna.

Após a introdução (tópico 1), são apresentadas algumas definições do PC e os conceitos envolvidos pelo tema (tópico 2). O tópico seguinte (3) apresenta os trabalhos relacionados, identificados através de uma pesquisa exploratória. Após a descrição do planejamento e execução da SLR (tópico 4), são relatados os resultados obtidos (tópico 5). O tópico 6 aponta para uma lacuna que pode ser considerada no desenvolvimento de novas ferramentas para o ensino do PC na educação básica. O tópico 7 analisa estratégias para a concepção e avaliação dessas ferramentas. Ao final (tópico 8) são apresentadas as considerações finais e recomendações para trabalhos futuros.

## 2. Pensamento Computacional

A ideia central de Wing (2006) foi bem expressa pela *Royal Society* (2012, p. 29), que definiu o PC como “o processo de reconhecer os aspectos computacionais no mundo que nos rodeia, e aplicar ferramentas e técnicas da Ciência da Computação para compreender e raciocinar sobre os sistemas e processos naturais e artificiais”.

Definições nominativas como a apresentada pela *Royal Society* são boas fontes para o entendimento da essência do PC, porém, pouco nos dizem sobre que tipo de conteúdo pode ser adequado para o processo de ensino-aprendizagem do tema.

Assim, a definição operacional oferecida pela CSTA (2011) (*Computer Science Teachers Association*), em conjunto com a ISTE (*International Society for Technology in Education*), torna-se mais adequada para pesquisas envolvendo práticas de ensino-aprendizagem do PC. A CSTA (2011) define o PC para educação básica como sendo um **processo de resolução de problemas** que inclui características como: formulação de problemas computáveis; organização, análise e representação de dados através de modelos e simulações; implementação de soluções visando a otimização de passos e recursos, bem como a generalização dessas soluções para uma ampla gama de problemas.

Além da definição operacional, a CSTA (2011) também fornece um vocabulário com nove conceitos essenciais para o PC. Em uma análise de agrupamentos (*cluster analysis*) realizada por Mannila et al. (2014), os conceitos fornecidos pela CSTA foram

pertinentemente classificados em três grupos: (i) coleta de dados, análise de dados, representação de dados; (ii) decomposição de problemas, abstração, algoritmos; (iii) automação, simulação e paralelização.

Os conceitos do primeiro grupo envolvem atividades puramente relacionadas com dados: coletar dados (coleta); dar sentido, encontrar padrões e tirar conclusões sobre dados (análise); apresentar dados de forma apropriada através de gráficos, tabelas e imagens (representação) [CSTA 2011]. Os conceitos do segundo agrupamento estão diretamente relacionados com a formulação de problemas, envolvendo atividades como dividir em partes gerenciáveis (decomposição), reduzir a complexidade (abstração) e definir passos ordenados para se chegar a um objetivo (algoritmos) [CSTA 2011].

O último grupo compreende os conceitos relacionados com o processamento de problemas, e que envolvem atividades como utilizar computadores ou máquinas para tarefas repetitivas (automação), representar ou modelar processos (simular) e organizar recursos para realizar tarefas de forma simultânea (paralelização) [CSTA 2011].

Brennan e Resnick (2012) também empregam uma abordagem tripartida para a análise do PC, porém, com o foco voltado para práticas de programação. O framework desenvolvido pelos pesquisadores do *Media Lab* do MIT (*Massachusetts Institute of Technology*) é composto pelas seguintes dimensões: conceitos, práticas e perspectivas computacionais.

A primeira dimensão tem o foco voltado para conceitos comumente encontrados nas linguagens de programação (variáveis, condicionais, operadores, ...). A segunda contempla as práticas para resolução de problemas que ocorrem durante a atividade de programação (abstração, modularização, testes, ...), enquanto que a última se concentra na análise do entendimento dos estudantes sobre suas relações com os outros e com o mundo tecnológico em torno deles [Brennan e Resnick 2012].

### 3. Trabalhos Relacionados

A literatura recente apresenta estudos que buscam identificar as ferramentas utilizadas no ensino-aprendizagem do PC.

Mannila et al. (2014) apresenta os resultados de um questionário online voltado para professores de diversos países da Europa. Um dos objetivos do questionário foi identificar as tecnologias utilizadas pelos professores ao se trabalhar com os conceitos do PC em sala de aula. Com base na sintetização de 961 respostas, em grande parte de professores do ensino fundamental, o estudo apresenta uma análise estatística sobre os tipos de ferramentas utilizados: recursos da web, linguagens, robótica, simulação dentre outros.

Em outro trabalho recente, Lye e Koh (2014) apresentam uma resposta para a seguinte pergunta de pesquisa: como a programação foi incorporada nos currículos da educação básica? Para tal, uma revisão de 27 estudos primários foi realizada. Dentre os 9 estudos selecionados, destacam-se as ferramentas Logo e Scratch.

### 4. Metodologia

Para identificar as principais ferramentas que foram utilizadas no ensino-aprendizagem do PC de 2006 até o início de 2015, foi adotada a metodologia de revisão sistemática da

literatura ou SLR (*Systematic Literature Review*), um tipo de estudo secundário que segue um procedimento bem definido para identificar, analisar e interpretar evidências a respeito de uma pergunta de pesquisa específica de maneira repetível e, até certo grau, não tendenciosa [Kitchenham e Charters 2007].

Kitchenham e Charters (2007) agrupam as atividades de uma SLR em três fases sequenciais: planejamento, execução e relato. A primeira fase envolve, no mínimo, três atividades: identificação da necessidade da SLR, formulação da(s) pergunta(s) de pesquisa e desenvolvimento do protocolo de revisão. Como atividades da segunda fase, tem-se: identificação da pesquisa; seleção e avaliação dos estudos; extração e síntese de dados. A formatação e avaliação do relatório, bem como a especificação do mecanismo de divulgação, representam as atividades da terceira e última fase.

Os subtópicos seguintes descrevem as fases de planejamento e execução da SLR objeto deste trabalho. Em seguida (tópico 5), é apresentado o relatório.

#### 4.1. Planejamento

Através de uma pesquisa exploratória realizada em abril de 2015 nas bases ACM (dl.acm.org) e IEEE Xplore (ieeexplore.ieee.org) com o termo “*computational thinking*”, verificou-se a existência de um grande número de publicações que trata da utilização de alguma ferramenta como estratégia para o ensino do PC. Verificou-se, também, que o nome das ferramentas citadas nos títulos e resumos disponíveis varia muito entre uma e outra publicação, o que dificulta a identificação das ferramentas que foram utilizadas com maior frequência. Estes dados difusos e cientificamente disponíveis apontaram para a necessidade de uma SLR que respondesse à seguinte pergunta de pesquisa:

*Quais foram as principais ferramentas utilizadas no ensino-aprendizagem do pensamento computacional de 2006 até 2015?*

No protocolo da SLR, foram incluídas as fontes de dados ACM Digital Library, IEEE Xplore Digital Library, ScienceDirect, ERIC (*Education Resources Information Center*), SBIE (Simpósio Brasileiro de Informática na Educação), WIE (Workshop de Informática na Escola). Para cada uma das fontes, foi definida uma estratégia de busca específica (Quadro 1), considerando-se, para tal, os termos “*computational thinking*”, “*education*”, “*teaching*” e “*learning*”, bem como o idioma e as ferramentas de pesquisa disponíveis em cada caso.

**Quadro 1. Estratégias de busca**

Fonte de dados	Estratégia de busca
ACM (dl.acm.org)	<i>Abstract:("computational thinking") and Abstract:(education* or teach* or learn*)</i>
IEEE Xplore (ieeexplore.ieee.org)	<i>("computational thinking") AND (education* or teach* or learn*)</i>
ScienceDirect (sciencedirect.com)	<i>(pub-date &gt; 2005 and TITLE-ABSTR-KEY("computational thinking") and TITLE-ABSTR-KEY(education* or learn* or teach*))</i>
ERIC (eric.ed.gov)	<i>abstract:"computational thinking" AND abstract:(learn OR learning OR teach OR teaching)</i>

SBIE ( <a href="http://www.br-ie.org/pub/index.php/sbie">http://www.br-ie.org/pub/index.php/sbie</a> )	“ <i>Pensamento Computacional</i> ”
WIE ( <a href="http://www.br-ie.org/pub/index.php/wie">http://www.br-ie.org/pub/index.php/wie</a> )	“ <i>Pensamento Computacional</i> ”

Para a seleção das publicações, foram estabelecidos os seguintes critérios: estar escrita em inglês ou português; ter sido publicada entre 2006 (ano de popularização do termo PC) e 2015; conter os termos da string de busca no título, resumo ou palavras-chave. Os critérios de exclusão utilizados foram: não relatar alguma prática (aula, experimento ou workshop) com ênfase no ensino-aprendizagem do PC; não citar o nome da(s) ferramenta(s) utilizada(s) na prática. Para o processo de extração e síntese dos dados, foram estabelecidos os seguintes campos: referência da publicação (id, ano, autores e título), nome da(s) ferramenta(s) utilizada(s) e nível escolar alvo.

#### 4.2. Execução

Com o objetivo de encontrar o máximo de publicações relacionadas com a pergunta de pesquisa, e ao mesmo tempo realizar uma SLR sem viés, a estratégia apresentada no tópico 4.1 selecionou seis fontes de publicações relevantes e definiu termos-chave de busca suficientemente genéricos, a fim de que os resultados da busca não favorecessem nenhum tipo de ferramenta específico.

A execução das estratégias de busca apresentadas no Quadro 1 em 15/04/2015 forneceu um total de 288 resultados: 202, 34, 17, 29, 2 e 4 nas buscas em ACM, IEE, ScienceDirect, ERIC, SBIE e WIE, respectivamente. As 288 publicações atenderam aos critérios de seleção, porém, somente 106 delas relataram alguma prática (experimento, aula ou workshop) de ensino-aprendizagem do PC e, ao mesmo tempo, citaram o nome da(s) ferramenta(s) utilizada(s). Dentre os resultados selecionados, encontram-se cinco publicações da comunidade científica brasileira, duas em português e outras três em inglês.

Para a extração e sintetização dos dados relativos ao nível escolar, foi necessário estabelecer algumas regras de conversão entre os sistemas educacionais estrangeiros e o brasileiro, a saber: *Major* e *Undergraduate* = Graduação (G); *High School* = Ensino Médio (EM); *Middle School* = Ensino Fundamental (EF). Em várias publicações, o nível escolar não estava explícito ou o perfil do público era difuso. Para contemplar estas situações, estabeleceu-se a categoria de nível escolar Indefinido (I).

Os dados extraídos das 106 publicações selecionadas foram disponibilizados em: <https://goo.gl/At4WnJ>.

#### 5. Resultados

A Tabela 1 apresenta dados estatísticos das 10 ferramentas com maior frequência de utilização. Dentre as ferramentas não contempladas na tabela encontram-se: Logo, C#, JAVA, Excel, Minecraft (minecraft.net), Greenfoot (greenfoot.org), AgentWeb (agentweb.inf.usi.ch), BLAST (blast.ncbi.nlm.nih.gov), Simulation Creation Toolkit

(basawapatna.com) e CHERP (ase.tufts.edu), com ao menos 2 relatos de utilização, e outras 72 ferramentas com apenas 1 relato de utilização.

**Tabela 1. Quantidade de relatos de utilização de ferramenta por nível escolar**

Ferramenta	Nível escolar*				Total
	EF	EM	G	I	
Scratch	11	7	3	10	31
Alice	3	3	2	3	11
App Inventor (versões I e II)	2	4	2	3	11
LEGO Mindstorms	2	1	2	3	8
CS Unplugged	4	2	-	1	7
AgentSheet/AgentCubes	6	-	-	-	6
MATLAB	-	-	6	-	6
Scalable Game Design (projeto)	2	1	-	3	6
Python	-	2	1	2	5
Kodu	3	-	-	1	4
<b>Total</b>	<b>33</b>	<b>20</b>	<b>16</b>	<b>26</b>	<b>95</b>

\* EF = Ensino fundamental | EM = Ensino médio | G = Graduação | I = Indefinido

O Scratch ([scratch.mit.edu](http://scratch.mit.edu)) é uma linguagem de programação visual, ou VPL (*Visual Programming Language*), criada em 2003 pelo *Media Lab* do MIT e voltada para pessoas que estão começando a programar. O ambiente de programação Alice ([alice.org](http://alice.org)) permite criar animações 3D e também emprega o conceito de VPL, foi projetado para ser o primeiro contato de um estudante com a POO (Programação Orientada a Objetos). O App Inventor ([appinventor.mit.edu](http://appinventor.mit.edu)), também criado pelo MIT, é uma VPL que permite aos novatos em programação criar aplicativos para dispositivos Android. LEGO Mindstorms ([mindstorms.lego.com](http://mindstorms.lego.com)) é o resultado de uma parceria entre o MIT e o grupo LEGO, uma linha de brinquedos que permite criar robôs que executam funções básicas pré-programadas. O CS Unplugged ([csunplugged.org](http://csunplugged.org)), por sua vez, é um conjunto de atividades para o ensino de ciências da computação sem o uso do computador.

O MATLAB ([mathworks.com](http://mathworks.com)) é uma ferramenta mais utilizada nos níveis de graduação. Trata-se de um software interativo de alta performance voltado para cálculos numéricos. O SGD (Scalable Game Design) ([sgd.cs.colorado.edu/wiki](http://sgd.cs.colorado.edu/wiki)) é um projeto da Universidade do Colorado que visa reinventar o ensino de computação em escolas públicas nos EUA através do desenvolvimento de jogos. Uma versão brasileira do SGD foi implantada pela PUC-Rio ([www.serg.inf.puc-rio.br/wiki](http://www.serg.inf.puc-rio.br/wiki)). AgentSheets/AgentCubes ([agentsheets.com](http://agentsheets.com)) é uma ferramenta que permite criar jogos e simulações baseados em agentes e publicá-los na web. SGD e AgentSheets também utilizam o conceito de VPL. Python ([python.org](http://python.org)) é uma linguagem de programação que prioriza a legibilidade do

código através de uma sintaxe clara e concisa. A ferramenta Kodu (kodugamelab.com) permite que crianças criem jogos no PC e no Xbox através de uma VPL simples.

Os dados estatísticos apresentados na Tabela 1 confirmam o grande interesse da comunidade científica pelo ensino do PC na educação básica (K-12) e revelam que a abordagem através de VPLs (Scratch, App Inventor, Alice, SGD, AgentSheet, Kodu) foi a mais utilizada de 2006 até o início de 2015. Lye e Koh (2014) entendem que as VPLs são mais adequadas para o contexto da educação básica porque reduzem a carga cognitiva exigida pela sintaxe das linguagens tradicionais, permitindo aos estudantes se concentrar na lógica de resolução do problema.

## 6. Onde está Alan Turing?

Se o planejamento inicial deste trabalho fosse mantido, o título deste tópico seria algo como “Considerações Finais”. No entanto, após obter-se a resposta para a problemática inicial de pesquisa (tópico 4.1), um novo questionamento inusitado surgiu: já que o PC baseia-se em técnicas da Ciência da Computação [Royal Society 2012; Wing 2006], por que nenhum dos relatos apresentados na Tabela 1 utilizou uma ferramenta inspirada nas teorias e formalismos de Alan Turing, por exemplo, a máquina de Turing (MT)?

A MT é um dispositivo teórico, desenvolvido em 1936 pelo matemático Inglês Alan Turing (1912-1954), que realiza operações simples, mas que quando combinadas permitem resolver qualquer problema computável [Petzold 2008]. O modelo MT baseia-se na ideia de uma fita infinita e uma cabeça de leitura e escrita que se move ao longo da fita. A cada operação, a MT decide se escreve ou não um símbolo na fita e move-se para a esquerda ou direita, conforme o estado em que ela se encontra e a lógica programada para este estado. O processo de computação continua até que algum símbolo lido seja aceito ou rejeitado [Sipser 1997]. Mas por que pensar na MT como ferramenta para o ensino-aprendizagem do PC na educação básica?

A SLR apresentada neste trabalho aponta para a atividade de programação como sendo a principal estratégia utilizada no ensino do PC. As VPLs identificadas na Tabela 1 conduzem os alunos a uma forma de pensar computacionalmente baseada em laços de repetição, desvios condicionais e outras estruturas algorítmicas herdadas das linguagens de programação tradicionais. Diferente disto, a programação de uma MT conduz a uma forma de pensar baseada em trocas de estados, o que segundo Zylberberg (2011) et al., representa um modelo adequado do funcionamento do cérebro consciente.

Assim, uma VPL inspirada na MT pode representar uma nova estratégia para o ensino do PC na educação básica, pois além de reduzir a carga cognitiva exigida pela sintaxe e estruturas algorítmicas das linguagens de programação tradicionais, conduz a uma diferente maneira de pensar e programar. Além disso, a habilidade da MT em determinar os limites da computação [Petzold 2008] favorece a criação de desafios para explorar a noção de computabilidade (o que é computável?) presente na descrição do PC apresentada por Wing (2006, 2009).

## 7. Análise de estratégias para o desenvolvimento e avaliação de uma linguagem de programação visual inspirada na máquina de Turing

Os simuladores de MT disponíveis na internet, tal como o apresentado na página inicial da Google em comemoração ao 100º aniversário de Alan Turing (<https://goo.gl/rIFbYi>), comumente utilizam o alfabeto binário (0 e 1). No entanto, ao se desenvolver uma MT para os alunos da educação básica, pode-se considerar a utilização de notações menos abstratas, tais como figuras de objetos e elementos presentes no cotidiano dos alunos. Uma simbologia baseada em cores é outra possível estratégia a ser considerada, tendo em vista que a busca por imagens no Google com o termo “*puzzle game*” revela uma grande quantidade de jogos que utiliza esta abordagem.

A programação de uma MT consiste basicamente na definição da sua função de transição, onde para cada tupla *estado x símbolo lido* são definidas as seguintes regras: o novo estado, o símbolo a ser escrito (ou não) e a direção (esquerda ou direita) para a qual a cabeça de leitura deve se movimentar [Sipser 1997]. Uma possível maneira de programar uma MT utilizando o conceito de VPL é através da criação de uma interface composta de botões, caixas de seleção e outros componentes gráficos que permitam aos alunos descrever o programa utilizando apenas as funções do mouse. Dependendo da complexidade do desafio de programação e a faixa etária dos alunos, alguns parâmetros da função de transição podem estar pré-configurados, fazendo com que a atividade de programação resuma-se, por exemplo, à configuração da direção (esquerda ou direita) em cada regra.

Para a avaliação da ferramenta, há um aspecto importante a ser considerado. Ao analisar o *leadership toolkit* da CSTA (2011) para o ensino do PC na educação básica, percebe-se que as atividades que exigem um raciocínio procedimental mais formalizado (descrição de um algoritmo, por exemplo) são normalmente destinadas aos níveis escolares mais altos (a partir do sexto ano). Assim, por se tratar de uma ferramenta que envolve programação, entende-se que a MT discutida neste trabalho é mais adequada para os alunos a partir dos últimos anos do ensino fundamental.

## 8. Considerações Finais e Recomendações para Trabalhos Futuros

Este trabalho teve como principal objetivo realizar uma revisão sistemática da literatura para identificar as principais ferramentas que foram utilizadas no ensino-aprendizagem do PC de 2006 até o início de 2015. A meta-análise dos dados indica que as principais ferramentas são as linguagens de programação visuais, ou VPLs (*Visual Programming Languages*), e confirma um maior interesse da comunidade científica pelo ensino do PC na educação básica.

O emprego de atividades de programação para o ensino-aprendizagem do PC pode ter sua justificativa baseada na CSTA (2011), que define o PC como um processo de resolução de problemas e apresenta um vocabulário com nove conceitos a serem explorados: coleta de dados, análise de dados, representação de dados, decomposição de problemas, abstração, algoritmos, automação, simulação e paralelização. A atividade de programação, além de ser essencialmente destinada à resolução de problemas, permite que grande parte desses conceitos seja naturalmente ou ao menos facilmente explorada.



A escolha pelas VPLs, por sua vez, está diretamente relacionada com nível escolar para o qual as atividades de programação são destinadas (educação básica). Lye e Koh (2014) explicam que as VPLs reduzem a carga cognitiva exigida pela sintaxe das linguagens de programação tradicionais, permitindo aos estudantes se concentrar na lógica de resolução do problema.

Uma lacuna que pode ser identificada na SLR deste trabalho é a ausência de ferramentas inspiradas nas teorias e formalismos desenvolvidos por Alan Turing, o pai da computação moderna. Endente-se que uma VPL inspirada na MT pode representar uma nova estratégia para o ensino do PC na educação básica, pois conduz a uma forma de pensar e programar ainda pouco explorada com os alunos deste nível escolar. Além disso, uma VPL inspirada na MT favorece a criação de desafios para explorar a noção de computabilidade presente na descrição do PC apresentada por Wing (2006, 2009).

Assim, este trabalho levanta duas novas perguntas de pesquisa que deverão ser respondidas em trabalhos futuros: (i) o uso de uma VPL inspirada na MT pelos alunos da educação básica desperta o interesse destes alunos pela atividade de programação? (ii) uma VPL inspirada na MT permite explorar noções de computabilidade com os alunos da educação básica?

## Referências

- Andrade, D. et al. (2013). “Proposta de Atividades para o Desenvolvimento do Pensamento Computacional no Ensino Fundamental”. In *Anais do XIX Workshop de Informática na Escola (WIE 2013)*.
- Brennan, K. and Resnick, M. (2012). “New frameworks for studying and assessing the development of computational thinking”. Annual Meeting of the American Educational Research Association (AERA'12), Vancouver, Canada.
- Campos, G. M. de et al. (2014). “Organização de Informações via Pensamento Computacional: Relato de Atividade Aplicada no Ensino Fundamental”. In *Anais do XX Workshop de Informática na Escola (WIE 2014)*.
- Carvalho, T. et al. (2013). “Discussing the Challenges Related to Deployment of Computational Thinking in Brazilian Basic Education”. 2013 2nd Workshop-School on Theoretical Computer Science.
- CSTA. (2011). “Computational thinking teacher resources”. 2<sup>th</sup> edition. Disponível: < >. Acesso: maio/2015.
- Gouws, L., Bradshaw, K. and Wentworth, P. (2013). “Computational thinking in educational activities: an evaluation of the educational game light-bot”. In *Proceedings of the 18th ACM conference on Innovation and technology in computer science education - ITiCSE '13*.
- França, R. S. de and Amaral, H. J. C. do. (2013). “Proposta Metodológica de Ensino e Avaliação para o Desenvolvimento do Pensamento Computacional com o Uso do Scratch”. In *Anais do XIX Workshop de Informática na Escola (WIE 2013)*.
- França, R. S. de and Tedesco, P. C. de A. R.. (2014). “Um modelo colaborativo para a aprendizagem do pensamento computacional aliado à autorregulação”. In *Anais do III Congresso Brasileiro de Informática na Educação (CBIE 2014)*.

- Hill, C. (2014). "Computational thinking curriculum development for upper elementary school classes". In *Proceedings of the tenth annual conference on International computing education research - ICER '14, 2014*.
- Kitchenham, B. and Charters, S. (2007). "Guidelines for performing systematic literature reviews in software engineering". Technical Report EBSE 2007-001, Keele University and Durham University Joint Report.
- Koh, K. H. et al. (2014). "Early validation of computational thinking pattern analysis". In *Proceedings of the 2014 conference on Innovation & technology in computer science education - ITiCSE '14, 2014*.
- Lye, S. Y. and Koh, J. H. L. (2014). "Review on teaching and learning of computational thinking through programming: What is next for K-12?". In *Computers in Human Behavior*, v. 41, p. 51-61.
- Mannila, L. et al. (2014). "Computational Thinking in K-9 Education". In *Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference - ITiCSE-WGR '14*.
- Petzold, C. (2008). "The Annotated Turing: A Guided Tour Through Alan Turing's Historic Paper on Computability and the Turing Machine". Wiley Publishing.
- Ribeiro, L. et al. (2013). "Computational Thinking: Possibilities and Challenges". In *2nd Workshop-School on Theoretical Computer Science*.
- Roscoe, J. F., Fearn, S. and Posey, E. (2014). "Teaching Computational Thinking by Playing Games and Building Robots". In *2014 International Conference on Interactive Technologies and Games*.
- Royal Society. (2012). "Shut down or restart: The way forward for computing in UK schools". Disponível: <https://royalsociety.org/~media/education/computing-in-schools/2012-01-12-computing-in-schools.pdf>. Acesso: maio/2015.
- Sipser, M. (1997). "Introduction to the theory of computation". Boston: PWS Pub. Co.
- Wing, J. M. (2006). "Computational thinking". In *Communications of the ACM*, v. 49, n. 3, p. 33.
- Wing, J. M. (2009). "Computational thinking". In *Journal of Computing Sciences in Colleges*, v. 24, Issue 6.
- Zylberberg, A. et al. (2011). "The human Turing machine: a neural framework for mental programs". Trends in Cognitive Sciences.