

## Metodologias de avaliação do Pensamento Computacional: uma revisão sistemática\*

Christiano Avila, Simone Cavalheiro, Adriana Bordini,  
Mônica Marques, Maicon Cardoso, Gustavo Feijó

<sup>1</sup> Centro de Desenvolvimento Tecnológico  
Programa de Pós-Graduação em Computação  
Universidade Federal de Pelotas (UFPEL)  
Rua Gomes Carneiro, 1 - 96.010-610 - Pelotas - RS - Brasil

{christianoavila, simone.costa, abordini}@gmail.com

{moniicamarquees, maiconsc0, gusfeijo}@gmail.com

**Abstract.** *This article presents a survey of works that perform in some way the assessment of Computational Thinking (CT) in several teaching contexts and that were published in the main research bases of the academic community between 2011 and 2016. In order to report the state of the art in relation to the assessment of the CT, a systematic literature review (SLR) was developed with the focus in studies that carried out some intervention and evaluation. In this study, eight approaches have been identified that were focus for the evaluation of CT, of which two stand out: intervention with own assessment and intervention with own assessment based on existing methods.*

**Resumo.** *Este artigo apresenta um levantamento de trabalhos que realizam avaliação do Pensamento Computacional (PC) em vários contextos de ensino e que foram publicados nas principais bases de pesquisa entre 2011 e 2016. Com o objetivo de relatar o estado da arte em relação a avaliação do PC, foi desenvolvida uma revisão sistemática de literatura (RSL) com o foco em estudos que realizaram alguma intervenção com avaliação. Neste trabalho foram identificadas oito abordagens que têm sido foco para a avaliação do PC, das quais duas se destacam: intervenção com avaliação própria e intervenção com avaliação própria fundamentada em métodos existentes.*

### 1. Introdução

Pensamento Computacional (PC) é uma metodologia para a resolução de problemas baseada nos fundamentos e técnicas da Ciência da Computação. De forma geral, habilidades comumente utilizadas para a criação de programas computacionais para solucionar problemas específicos são utilizadas como uma metodologia para resolver problemas nas mais diversas áreas. É um processo capaz de desenvolver competências que podem ser utilizadas em qualquer área do conhecimento e para toda a vida [Wing 2006].

Em [CSTA 2011], define-se que este processo inclui (mas não está limitado) as seguintes características: formalizar problemas em uma estrutura que permita a utilização

---

\*Projeto com chancela da SBC e realizado com o apoio da PREC e PRPPG / UFPEL.

de um computador ou outra ferramenta para resolvê-los; organizar logicamente e analisar dados; representar dados por meio de abstrações como modelos e simulações; automatizar soluções por meio de algoritmos (série ordenada de passos); identificar, analisar e implementar possíveis soluções com o objetivo de atingir a mais eficiente e efetiva combinação de passos e recursos; generalizar e modificar a solução de um problema para resolver uma ampla variedade de outros problemas.

Simultaneamente, muito se discute sobre a inserção da computação como ciência na educação, incluindo-se o PC como um de seus pilares. Diversos trabalhos têm sido propostos, tanto com o objetivo de introduzir a computação na educação básica [Bordini et al. 2016a, Bastos et al. 2015, Kalelioğlu 2015], quanto de desenvolver as habilidades do PC [Bordini et al. 2016b, Rodrigues et al. 2015, Araújo et al. 2015, Farias et al. 2015, Brennan and Resnick 2012]. Ao se integrar o PC na educação, uma questão fundamental é a de conseguir identificar se o trabalho desenvolvido de fato contribui com a promoção do PC, ou seja, se é efetivo no atingimento dos objetivos que foram propostos. Este trabalho busca apresentar um panorama geral na área de avaliação do PC.

Identificou-se que existem dois levantamentos, um com foco nos trabalhos no Brasil [Araújo et al. 2016b] e outro mundial [Araújo et al. 2016a], que objetivam descrever os instrumentos utilizados para avaliação do PC bem como as habilidades consideradas. O presente trabalho, adicionalmente, preocupa-se em descrever os fundamentos teóricos (taxonomias, teorias de aprendizagem, entre outros) considerados nos diversos instrumentos, relacionando-os com as características que eles permitem avaliar, e também em analisar o potencial de generalização e reaplicação dos trabalhos considerados. O trabalho está organizado como segue. A Seção 2 detalha a metodologia utilizada e a Seção 3 responde as questões de pesquisa consideradas. As considerações finais estão na Seção 4.

## 2. Metodologia da Revisão Sistemática

Este levantamento seguiu a metodologia de trabalho conhecida como Revisão Sistemática de Literatura (RSL). O objetivo foi o de identificar e analisar pesquisas relevantes que realizavam avaliação de habilidades ou dimensões do PC. Considerou-se os artigos publicados nos últimos 6 anos (de 2011 à agosto de 2016) em periódicos e anais de conferências internacionais, no idioma inglês. A busca dos trabalhos foi feita nas seguintes bases digitais on-line: IEEE, ACM, *ebsohost* e *Science Direct*. A *string* de busca utilizada foi a seguinte: “*computational thinking*” e as palavras *assessment* ou *assessing* ou *evaluation* ou *measuring* ou *performance* ou *test* em qualquer parte do texto.

Para selecionar os artigos que respondiam ao foco da pesquisa, definiu-se como critério de inclusão: I1 - trabalhos que incluem avaliação do PC como meio ou como fim do processo de ensino. Os critérios de exclusão considerados foram: E1 - trabalhos incompletos, tais como, artigos curtos, pôsteres, painéis, apresentações; E2 - outras revisões de literatura na área; E3 - trabalhos duplicados; E4 - trabalhos de um mesmo projeto, somente o mais recente será considerado; E5 - trabalhos que apresentam avaliações sem apresentar o método (detalhamento) utilizado. Por meio da leitura dos textos na íntegra e aplicando os critérios estabelecidos totalizaram 58 artigos.

### 3. Avaliação do Pensamento Computacional

Nesta seção são apresentadas as respostas às questões de pesquisa consideradas neste estudo. A relação dos estudos que fazem parte desta revisão sistemática estão detalhados na Tab. 4. No que segue, utiliza-se o campo ID da Tabela para referenciar os trabalhos.

**QP1 - Quais são as abordagens que incluem avaliação do PC?** Dentre os trabalhos analisados, identificaram-se 8 diferentes propostas que incluem avaliação de habilidades do PC, descritas na Tabela 1. As 3 primeiras abordagens são intervenções de ensino, tal que trabalhos em: A1 realizam a avaliação do PC com a criação de métodos e instrumentos próprios; A2 aplicam métodos ou padrões já propostos na literatura, sem alterações; A3 utilizam métodos existentes para complementar suas próprias avaliações. Outras 3 abordagens consideram ferramentas, onde trabalhos em: A4 avaliam uma ferramenta específica, com o objetivo de verificar se esta desenvolve habilidades do PC; A6 utilizam ferramentas que avaliam habilidades do PC; A7 propõem ferramentas que avaliem habilidades do PC. Em A5 são propostos métodos para avaliar o PC; e em A8 são analisadas habilidades do PC em métodos de avaliação existentes.

Conforme detalhado na Tabela 1, a maioria dos artigos enquadram-se na categoria A1. Nesta abordagem, são criados métodos e instrumentos de avaliação próprios para medir o desenvolvimento de habilidades do PC. Em geral, as avaliações qualitativas são realizadas por meio de observações, questionários de opinião, conversas informais e entrevistas com alunos e professores; e avaliações quantitativas, por meio de pré- e pós-testes, onde geralmente são realizados o mesmo teste no início e fim da intervenção. Em seguida, a maior quantidade de trabalhos está na abordagem A3, os quais baseiam-se em testes de inteligência, leitura, rubricas, frameworks, ou outras propostas já existentes para criarem métodos próprios de avaliação.

**Tabela 1. Classificação dos artigos quanto a sua abordagem.**

ID	Abordagem	Qtd.
A1	intervenção com avaliação própria	26
A2	intervenção com avaliação existente	2
A3	intervenção com avaliação própria fundamentada em métodos existentes	12
A4	avaliação de ferramentas	2
A5	proposta de método de avaliação	6
A6	uso de ferramenta(s) para avaliar o desenvolvimento do PC	3
A7	proposta de ferramenta para avaliar o PC	6
A8	análise de habilidades do PC em método de avaliação existente	1
<b>Total</b>		<b>58</b>

**QP2 - Quais são as habilidades do PC que têm sido avaliadas e quais são os conceitos de computação abordados?** As Figuras 1(a) e 1(b) relacionam, respectivamente, as abordagens com as habilidades e conceitos desenvolvidos. A maioria dos artigos se concentra nas abordagens A1 e A3, que são intervenções feitas, com professores e/ou alunos, nas salas de aulas ou cursos extraclasse. Nas duas abordagens, as habilidades do PC mais trabalhadas foram as mesmas: pensamento algorítmico, resolução de problemas e abstração. Os conceitos de computação mais trabalhados, também foram os mesmos nas duas abordagens: algoritmos e programação. Observou-se que tanto no desenvolvimento de algoritmos quanto na programação, as atividades geralmente incluem os conceitos de variáveis, sequência, condicional e repetição. Percebeu-se também que a introdução de tais conceitos envolve o desenvolvimento e avaliação das habilidades de abstração, resolução de problemas e pensamento algorítmico.

Conforme detalhado na Figura 1(a), outras habilidades também tem sido avaliadas, tais como: colaboração, generalização, simulação e pensamento lógico. Igualmente, conforme detalhado na Figura 1(b), observa-se que diversos outros conceitos são considerados, entre eles: representação de dados, decomposição, paralelismo, simulação, modelagem, depuração, modularização, entre outros. Conceitos da computação que tiveram apenas uma ocorrência foram omitidos da Figura 1(b), sendo eles: complexidade de algorítmicos [10], reutilização [17], refinamento [18], ordenação [25], estrutura de arquivos [25], números binários [26] e recursão [14] todos na abordagem A1; internet [32] na A3; e computação móvel [47] na A5.

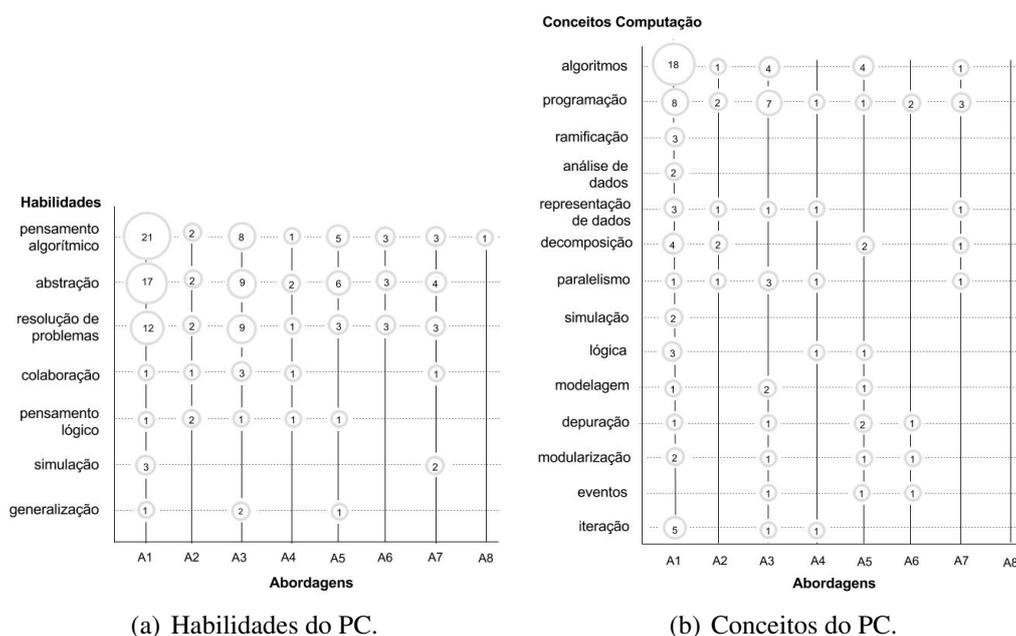


Figura 1. Habilidades e Conceitos do PC.

**QP3 - Quais fundamentos teóricos têm sido considerados nos instrumentos adotados?** Essa questão visa identificar possíveis tendências na utilização de fundamentos para embasar o processo de avaliação relacionadas ao desenvolvimento de habilidades do PC. Entretanto, em alguns estudos não foi possível identificar uma fundamentação, principalmente nos trabalhos que realizaram uma intervenção com avaliação própria.

A Tabela 2 relaciona os estudos com os fundamentos teóricos utilizados. Taxonomias de aprendizagem (Bloom [Bloom 1956, Conklin et al. 2005] e Solo [Biggs and Collis 2014]), identificadas em 6 estudos desta revisão, foram utilizadas para classificar os diferentes níveis de aprendizagem alcançados em uma intervenção. Em [45] a taxonomia Solo foi aplicada para avaliar o nível de aprendizagem de acordo com as respostas dadas a desafios de programação com *Scratch*.

Em [43], apresenta-se uma estratégia de avaliação do PC que combina níveis da taxonomia de Bloom (no domínio cognitivo) e altera sua ordem de aplicação, com o objetivo de melhor adequar a taxonomia ao escopo do PC. A estratégia de unir níveis da taxonomia também é utilizada pelos idealizadores de um teste nacional sobre PC aplicado pelo ministério da educação de Israel [36].

**Tabela 2. Fundamentos do processo de avaliação**

Fundamento	Estudos
Teste da Figura Complexa de <i>Rey-Osterrieth</i>	[52]
BIF ( <i>Behavioral Identification Form</i> )	[33]
Teste das Teorias de inteligência implícitas	[29]
Teste WASI	[14]
Teste GALT	[27]
Teste HAWIK	[31]
A Latent Trait Analysis (LTA)	[58]
Rubrica definida por Moreno-León e Robles	[28]
Rubrica fundamentada em “ <i>Backwards Design process</i> ”	[32]
Taxonomias (Bloom e Solo)	[39, 36, 12, 11, 43, 45]
REACT / CTPA / CTP	[40, 37, 38, 56, 53]
Framework 3D proposto por Brennan e Resnick	[30, 54]
Método de comparação: jogos × programação tradicional	[41]
Métricas de complexidade de software para avaliar produção em <i>Scratch</i>	[42]
Padrões de projeto (design patterns)	[42]

Em [11] a avaliação das habilidades dos alunos foi desenvolvida nos três níveis mais baixos da taxonomia de Bloom, isto é, sua capacidade de recordar, compreender e aplicar habilidades e técnicas específicas que são relevantes para o pensamento computacional. Já o estudo [12] cita a utilização da taxonomia de Bloom, porém aplicando apenas o nível mais básico, em uma etapa do estudo chamada de “*recall*”, onde os participantes eram convidados a explicar o propósito/significado de uma função e seus argumentos. Ainda no contexto das taxonomias, o estudo descrito em [39] utilizou uma “rubrica Bloom”, definida em [Fu et al. 2009], para analisar processos e resultados de aprendizagem. A rubrica permite ao avaliador classificar ou atribuir um conceito (falhou, aprovou, aceitável, bom e excelente) de acordo com o nível atingido pelo aluno.

Além da “rubrica Bloom”, descrita em [39], outros dois trabalhos utilizam rubricas para pautar a sua avaliação. Numa rubrica são definidos os níveis e uma descrição de referência que permite avaliar a aprendizagem. A rubrica definida por Moreno-León e Robles [Moreno-León and Robles 2015] foi utilizada em [28] para avaliar o desenvolvimento das habilidades do PC a partir dos jogos produzidos por estudantes que participaram de um Workshop de design de jogos destinado a adolescentes com autismo. Uma outra rubrica, baseada “*Backwards Design Process*” [Wiggins and McTighe 2005] é utilizada em [32] para avaliar planos de aula relacionados ao PC elaborados por professores participantes (enquanto alunos) de uma oficina sobre PC.

A ferramenta *Computational Thinking Pattern Analysis* (CTPA) [37, 38 e 40] e o sistema *cyberlearning Real-Time Evaluation and Assessment of Computational Thinking* (REACT) [53 e 56] estão sendo usados para avaliar a criação de jogos e simulações. Ambos trabalham os nove padrões do PC, *Computational Thinking Pattern* (CTP) [3]: controle de cursor, geração, absorção, colisão, transporte, empurrão, puxada, difusão e escalada. O CTP considera questões abstratas de programação, que podem ser facilmente analisadas na criação de jogos e simulações. Por exemplo, geração é utilizada quando um agente cria outro agente; absorção é representada, por exemplo, quando uma bala desaparece ao entrar no alvo. No REACT avaliaram o significado semântico dos personagens criados pelos alunos baseados no CTP, envolvendo na análise o número de mundos, agentes, formas, métodos, regras e instruções.

Na plataforma VIVA [54] (*the Vilnius collaboratively coded and Validated computer science questions/tasks for Assessment*), um ambiente de construção de conhecimento colaborativo entre professores e alunos, foram utilizados os fundamentos descritos no framework 3D proposto por [Brennan and Resnick 2012] para avaliar a capacidade do aluno

na resolução de problemas, incluindo ou não programação. Em [30] também se utilizou o mesmo framework para avaliar a efetividade na aprendizagem dos conceitos do PC, a partir da realização de tarefas no software Alice. Para isso, foi considerado a identificação e a resolução de problemas em programas prontos, a conclusão de tarefas semi-prontas e a construção de novas tarefas.

Alguns trabalhos [14, 27, 29, 31, 33, 52 e 58] adotam testes psicológicos e de QI para fundamentar a avaliação realizada. Em [52], um teste de PC que avalia a percepção visual e a função de memória visual de longo prazo, usando a tecnologia de tinta digital, foi baseado em testes psicológicos, no raciocínio espacial da inteligência e no teste da Figura Complexa de *Rey-Osterrieth*. O teste *Behavior Identification Form* (BIF) foi utilizado em [33] para avaliar pensamento abstrato e resolução de problemas. O *Latent Trait Analysis* (LTA) é um teste psicométrico que foi utilizado [58] para investigar os fatores psicométricos que determinam o sucesso no teste Bebras [Dagiené et al. 2016] (iniciativa internacional de promoção e avaliação do PC).

O teste *Implicit Theories of Intelligence* foi utilizado em [29] para avaliar como os perfis dos alunos (estratégico, construção do conhecimento, apático, aprendizagem de superfície e desamparo aprendido) mudaram ao longo do semestre e como essas mudanças diferem em função da autodeterminação, da autorregulação, da aprendizagem e do curso em que os alunos estão matriculados. Outro teste, o *Whimbey Analytical Skills Inventory* (WASI), por meio de pré- e pós-teste, foi utilizado em [14] para testar a eficácia da incorporação dos conceitos do PC em alunos da ciência da computação em um curso “pré-curriculo”. Já os testes desenvolvidos pelo *Group Assessment of Logical Thinking* (GALT) realizados com papel e lápis foram utilizados para validar uma metodologia apresentada em [27] verificando se: a) há melhora no pensamento lógico dos alunos; b) ajuda a compreensão dos alunos sobre PC; e c) aumenta o interesse dos alunos em computação. Alguns subtestes da bateria de testes padronizados de HAWIK IV (matrizes e busca por termos genéricos) foram utilizados para avaliar a melhoria das habilidades gerais de aprendizagem em [31]. Neste mesmo estudo, para avaliar a compreensão de leitura, foram utilizados os testes *Salzburger Lese-Screening*.

O estudo relatado em [57] desenvolveu um framework chamado *Progression of Early Computational Thinking* (PECT) para compreensão e avaliação do PC no ensino básico (da 1º a 6º ano). O framework, baseado em padrões de projeto, permite avaliar o conhecimento, habilidades e competências relacionadas ao PC.

**QP4 - Quais instrumentos têm sido utilizados para o fim de avaliação?** A maioria dos artigos realizaram avaliações qualitativas e quantitativas. Geralmente os instrumentos utilizados foram confeccionados pelos próprios pesquisadores, tais como: pré- e pós-questionários de opinião, pré- e pós-testes de conhecimento, entrevistas semi-estruturadas, observações e produção dos alunos (códigos, planos de aula, *folders*, etc).

Alguns trabalhos utilizaram testes padronizados existentes para avaliarem as habilidades do PC, tais como: HAWIK IV (inteligência), BIF (comportamento), WASI (competência cognitivas), entre outros, descritos na QP3. As ferramentas *Hairball* e *Dr.Scratch* foram usadas respectivamente em [55] e [42] para avaliarem os códigos produzidos pelos alunos em *Scratch*. Ferramentas também foram criadas [37, 53, 54 e 56] para avaliar habilidades do PC nas produções dos alunos: CTPA para avaliar os 9 padrões em jogos

e simulações; REACT para avaliar a produção do aluno em tempo real; e a plataforma VIVA, um ambiente colaborativo para avaliar a habilidade de resolução de problemas.

**QP5 - Qual o potencial de (re)aplicação e generalização dos trabalhos?** Nesta questão os artigos investigados são classificados quanto ao seu potencial de reprodutibilidade, ou seja, a capacidade de se reproduzir a metodologia de desenvolvimento do PC aliado ao método de avaliação proposto. Estabeleceu-se que, para que uma proposta de avaliação de um artigo seja totalmente reaplicável, o mesmo deve descrever de forma clara, objetiva e sem margem para interpretações dúbias os seus métodos, sejam eles de ensino ou avaliativos. Inclui-se nessa definição os seguintes itens: disponibilizar com riqueza de detalhes os métodos de ensino do PC, incluindo a habilidade a ser desenvolvida e a forma como a mesma é abordada em uma intervenção; disponibilizar os materiais utilizados durante o curso de aprendizado (tais como vídeos, sistemas e ambientes para programação, exercícios de fixação, dentre outros); por fim, apresentar a metodologia de avaliação, incluindo seus fundamentos (ou seja, as correlações realizadas durante a experimentação) e materiais pormenorizados (questionários, entrevistas, provas, etc.).

Assim, definiram-se três principais categorias de avaliação dos artigos quanto a reaplicabilidade de seus métodos de ensino e avaliação: a primeira categoria (Tipo A) engloba artigos que possuem riqueza de detalhes nos quesitos descritos no parágrafo anterior; a segunda categoria (Tipo B) é composta por trabalhos que não disponibilizam detalhes pormenorizados das suas metodologias, ou seja, onde o pesquisador ou educador interessado em reproduzir o estudo teria de entrar em contato com o autor para maiores informações (repasso de material de aula ou questionários são exemplos de conteúdos deficitários nos artigos do Tipo B); finalmente, os artigos presentes na última categoria (Tipo C) foram classificadas como trabalhos onde há uma expressiva falta de detalhes quanto a sua metodologia (relativo ao desenvolvimento do PC ou, ainda, referente ao processo de avaliação das habilidades). Os artigos de Tipo C necessitam de várias inferências para que sejam compreendidos (o que levará a muitos erros de reprodução dos experimentos caso sejam realizados por pesquisadores independentes e sem o contato com o autor original).

A Tabela 3 categoriza cada um dos trabalhos considerados neste levantamento. Como pode-se observar, 34,48% dos artigos estão inseridos na categoria de trabalhos Tipo A, enquanto 25,86% estão na categoria Tipo B e 39,66% compõem o Tipo C.

**Tabela 3. Categorização quanto a reprodutibilidade dos métodos apresentados.**

Categoria	Artigos (id.)	Qtd. (nº)	Qtd. (%)
Tipo A	[1, 2, 10, 16, 17, 19, 21, 22, 33, 34, 36, 37, 38, 40, 45, 46, 48, 54, 56, 57]	20	34,48
Tipo B	[3, 5, 8, 11, 20, 26, 27, 28, 31, 32, 35, 41, 42, 47, 52]	15	25,86
Tipo C	[4, 6, 7, 9, 12, 13, 14, 15, 18, 23, 24, 25, 29, 30, 39, 43, 44, 49, 50, 51, 53, 55, 58]	23	39,66
<b>Total</b>		<b>58</b>	<b>100</b>

#### 4. Considerações

Este artigo relata uma RSL, dos últimos 6 anos (de 2011 a agosto de 2016), do estado da arte em relação a avaliação do PC. Neste mapeamento foram encontradas oito abordagens que relacionam 58 artigos que avaliam habilidades do PC, seja nas experiências de ensino ou análises de ferramentas. As duas abordagens mais encontradas foram: intervenção com avaliação própria; e intervenção com avaliação própria fundamentada em métodos

existentes, onde as duas concentram mais de 65% dos artigos. Estas intervenções em sua maioria são experiências de ensino de algoritmos e programação com professores e alunos do ensino fundamental, trabalhando as seguintes habilidades do PC: pensamento algorítmico, resolução de problemas e abstração. Geralmente ambientes de programação visual são utilizados, tanto para as experiências de ensino, quanto para a implementação e análise de ferramentas. Destaca-se a utilização de instrumentos de avaliação qualitativos através de observações e quantitativos através de pré- e pós-testes. Ademais, ferramentas para avaliação de códigos provenientes de ambientes de programação visual são frequentemente utilizados, juntamente com plataformas de aprendizagem colaborativa.

Foi possível identificar que diversos estudos não reportam de forma explícita um fundamento teórico para suportar os instrumentos de avaliação e o processo avaliativo como um todo. Em relação aos estudos que descrevem um referencial teórico sobre avaliação, é possível verificar que não há um consenso ou consolidação de um determinado método de avaliação. Ou seja, ainda é necessário identificar ou desenvolver métodos (processos) e instrumentos mais genéricos que sejam capazes de mensurar com precisão a efetividade de intervenções ou ferramentas que se propõem a desenvolver ou promover conceitos, habilidades e competências relacionadas ao PC.

Como proposta futura pretende-se realizar uma pesquisa-ação com as abordagens mais utilizadas para validar as fundamentações identificadas neste levantamento. Procurando assim, propor métodos e/ou instrumentos/ferramentas que sejam efetivos na promoção das habilidades do PC.

**Tabela 4: Artigos Analisados.**

ID	Cat.	Título	Autor	Ano
1	A1	DISSECT: An experiment in infusing computational thinking in a sixth grade classroom	Peel et al.	2015
2		Introducing Computational Thinking in Education Courses	Yadav et al.	2011
3		Recognizing Computational Thinking Patterns	Basawapatna et al.	2011
4		CS4Impact: Measuring Computational Thinking Concepts Present in CS4HS Participant Lesson Plans	Bort et al.	2013
5		Design Insights into the Creation and Evaluation of a Computer Science Educational Game	Horn et al.	2016
6		CS0 for Computer Science Majors at Ohio University	Marling et al.	2016
7		Improving learning of computational thinking using computational creativity exercises in a college CSI computer science course for engineers	Shell et al.	2014
8		Design and Preliminary Results From a Computational Thinking Course	Kafura et al.	2015
9		A Pilot Computer Science and Programming Course for Primary School Students	Duncan et al.	2015
10		Efficient Egg Drop Contests: How Middle School Girls Think About Algorithmic Efficiency	Friend et al.	2013
11		First Year Student Performance in a Test for Computational Thinking	Gouws et al.	2013
12		Teaching computational thinking to non-computing majors using spreadsheet functions	Yeh et al.	2011
13		Documentation Comes to Life in Computational Thinking Acquisition with Agentsheets	Mota et al.	2012
14		Effectiveness of a Computational Thinking (CS0) Course on Student Analytical Skills	Dyne et al.	2014
15		Exploring Students' Computational Thinking Skills in Modeling and Simulation Projects: A Pilot Study	Grgurina et al.	2015
16		DISSECT: Exploring the relationship between computational thinking and English literature in K-12 curricula	Nesiba et al.	2015
17		Code and Tell: Assessing Young Children's Learning of Computational Thinking Using Peer Video Interviews with ScratchJr	Portelance et al.	2015
18		A Study on the Impact of Multidisciplinary Collaboration on Computational Thinking	Pulimood et al.	2016
19		GK-12 DISSECT: Incorporating computational thinking with K-12 science without computer access	Folk et al.	2015
20		Can computational thinking predict academic performance?	Haddad et al.	2015
21		Computational Thinking in a Game Design Course	Settle, A.	2011
22		Dancing Alice: Exploring Embodied Pedagogical Strategies for Learning Computational Thinking	Daily et al.	2014
23		Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences	Atmatzido et al.	2016
24		DISSECT: Analysis of pedagogical techniques to integrate computational thinking into K-12 curricula	Burgett et al.	2015

Continua na próxima página...

Tabela 4 – continuação.

ID	Cat.	Título	Autor	Ano
25	A1	An Informatics Perspective on Computational Thinking	Walden et al.	2013
26		Serious Toys: Three Years of Teaching Computer Science Concepts in K-12 Classrooms	Feaster et al.	2014
27	A2	Paper-and-pencil programming strategy toward computational thinking for non-majors: Design your solution	Kim et al.	2013
28		Game design workshop to develop computational thinking skills in teenagers with Autism Spectrum Disorders	Munoz et al.	2016
29	A3	Exploring Changes in Computer Science Students' Implicit Theories of Intelligence Across the Semester	Flanigan et al.	2015
30		An Exploration of Three-Dimensional Integrated Assessment for Computational Thinking	Zhong et al.	2016
31		Modeling: A computer science concept for general education	Sabitzer et al.	2015
32		Using backwards design process for the design and implementation of computer science (CS) principles: A case study of a colombian elementary and secondary teacher development program	Vieira et al.	2013
33		Effects of gender and abstract thinking factors on adolescents' computer program learning	Park et al.	2015
34		Programming moves: Design and evaluation of applying embodied interaction in virtual environments to enhance computational thinking in middle school students	Parmar et al.	2016
35		A novel interdisciplinary course in gerontechnology for disseminating computational thinking	Yang et al.	2011
36		A Nationwide Exam As a Tool for Improving a New Curriculum	Zur-Bargury et al.	2013
37		Early Validation of Computational Thinking Pattern Analysis	Koh et al.	2014
38		Scalable Game Design: A Strategy to Bring Systemic Computer Science Education to Schools Through Game Design and Simulation Creation	Repenning et al.	2015
39		Visual programming languages integrated across the curriculum in elementary school: A two year case study using Scratch in five schools	Sáez-López et al.	2016
40		Computing Creativity: Divergence in Computational Thinking	Bennett et al.	2013
41		A4	Experimental Evaluation of BeadLoom Game: How Adding Game Elements to an Educational Tool Improves Motivation and Learning	Boyce et al.
42	Comparing computational thinking development assessment scores with software complexity metrics		Moreno-León et al.	2016
43	A5	Computational Thinking in Educational Activities: An Evaluation of the Educational Game Lightbot	Lindsey et al.	2013
44		Assessing Problem-Solving Process At Scale	Grover et al.	2016
45		Using SOLO to Classify the Programming Responses of Primary Grade Students	Seiter, L.	2015
46		The Fairy Performance Assessment: Measuring Computational Thinking in Middle School	Werner et al.	2012
47		The Assessment of Mobile Computational Thinking	Sherman et al.	2015
48		Computational Thinking in Elementary and Secondary Teacher Education	Yadav et al.	2014
49	A6	Introducing Computer Programming to Children Through Robotic and Wearable Devices	Merkouris et al.	2015
50		A Work in Progress Paper: Evaluating a Microworlds-based Learning Approach for Developing Literacy and Computational Thinking in Cross-curricular Contexts	Jenkins, C.	2015
51		Interfaces for Thinkers: Computer Input Capabilities That Support Inferential Reasoning	Oviatt, S.	2013
52	A7	Digital ink for cognitive assessment of computational thinking	Ambrósio et al.	2014
53		Closing The Cyberlearning Loop: Enabling Teachers To Formatively Assess Student Programming Projects	Basawapatna et al.	2015
54		New Horizons in the Assessment of Computer Science at School and Beyond: Leveraging on the ViVA Platform	Giordano et al.	2015
55		Automatic detection of bad programming habits in scratch: A preliminary study	Moreno et al.	2014
56		Real Time Assessment of Computational Thinking	Koh et al.	2014
57		Modeling the Learning Progressions of Computational Thinking of Primary Grade Students	Linda et al.	2013
58	A8	Investigating the Psychometric Structure of Bebras Contest: Towards Mesuring Computational Thinking Skills	Hubwieser et al.	2015

## Referências

- Araújo, A. L. S. O. d., Andrade, W. L., and Guerrero, D. D. S. (2015). Pensamento computacional sob a visão dos profissionais da computação: uma discussão sobre conceitos e habilidades. In *Anais do WAIGProg/CBIE 2015*, pages 1454–1463.
- Araújo, A. L. S. O. d., Andrade, W. L., and Guerrero, D. D. S. (2016a). A systematic mapping study on assessing computational thinking abilities. In *Frontiers in Education Conference (FIE), 2016 IEEE*, pages 1–9. IEEE.
- Araújo, A. L. S. O. d., Andrade, W. L., and Guerrero, D. D. S. (2016b). Um mapeamento sistemático sobre a avaliação do pensamento computacional no Brasil. In *Anais dos Workshops do CBIE 2016*, pages 1147–1158.

- Bastos, N. S., Adamatti, D. F., and Carvalho, F. A. H. (2015). Ensino de lógica de programação no ensino médio e suas implicações na neurociências. In *Anais do SBIE 2015*, pages 459–468.
- Biggs, J. B. and Collis, K. F. (2014). *Evaluating the quality of learning: The SOLO taxonomy (Structure of the Observed Learning Outcome)*. Academic Press.
- Bloom, B. S. (1956). *Taxonomy of educational objectives: The classification of educational goals: Cognitive Domain*. Longman.
- Bordini, A., Avila, C., Marques, M., Cavalheiro, S., and Foss, L. (2016a). Desdobramentos do pensamento computacional no Brasil. In *Anais do SBIE 2016*, pages 200–209.
- Bordini, A., Avila, C. M. O., Weissahhn, Y., Cunha, M. M. d., Cavalheiro, S. A. d. C., Foss, L., Aguiar, M. S., and Reiser, R. H. S. (2016b). Computação na educação básica no Brasil: o estado da arte. *Revista de Informática Teórica e Aplicada*, 23(2):210–238.
- Brennan, K. and Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada*, pages 1–25.
- Conklin, J., Anderson, L. W., Krathwohl, D., Airasian, P., Cruikshank, K. A., Mayer, R. E., Pintrich, P., Raths, J., and Wittrock, M. C. (2005). *A taxonomy for learning, teaching, and assessing: A revision of bloom's taxonomy of educational objectives complete edition*.
- CSTA (2011). Computational thinking - leadership toolkit. <https://goo.gl/KnwWVw>.
- Dagienė, V. et al. (2016). Bebras-a sustainable community building model for the concept based learning of informatics and computational thinking. *Informatics in Education- An International Journal*, (Vol15\_1):25–44.
- Farias, A., Andrade, W., and Alencar, R. (2015). Pensamento computacional em sala de aula: Desafios, possibilidades e a formação docente. In *Anais do WAlgProg/CBIE 2015*, pages 1226–1235.
- Fu, F.-L., Wu, Y.-L., and Ho, H.-C. (2009). An investigation of cooperative pedagogic design for knowledge creation in web-based learning. *Computers & Education*, 53(3):550–562.
- Kalelioğlu, F. (2015). A new way of teaching programming skills to k-12 students: Code.org. *Computers in Human Behavior*, 52:200–210.
- Moreno-León, J. and Robles, G. (2015). Dr. scratch: A web tool to automatically evaluate scratch projects. In *Proceedings of the Workshop in Primary and Secondary Computing Education*, pages 132–133. ACM.
- Rodrigues, R. d. S., Andrade, W., Guerrero, D., and Sampaio, L. (2015). Análise dos efeitos do pensamento computacional nas habilidades de estudantes no ensino básico: um estudo sob a perspectiva da programação de computadores. In *Anais do SBIE 2015*, pages 121–130.
- Wiggins, G. P. and McTighe, J. (2005). *Understanding by design*. Ascd.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35.