

<https://doi.org/10.18222/eae.v33.8938>

# APRIMORAMENTO DO CT PUZZLE TEST PARA AVALIAÇÃO DO PENSAMENTO COMPUTACIONAL

 LEONARDO FELIPE DE ÁVILA CALBUSCH<sup>I</sup>

 NATÁLIA ELLERY RIBEIRO COUTO<sup>II</sup>

 JULIA ZANETTI ROCCA<sup>III</sup>

 ANDRÉ LUÍS ALICE RAABE<sup>IV</sup>

<sup>I</sup> Universidade do Vale do Itajaí (Univali), Itajaí-SC; Instituto Federal Catarinense (IFC), Brusque-SC, Brasil; [leonardo.calbusch@ifc.edu.br](mailto:leonardo.calbusch@ifc.edu.br)

<sup>II</sup> Universidade do Vale do Itajaí (Univali), Itajaí-SC, Brasil; [natalia.couto@edu.univali.br](mailto:natalia.couto@edu.univali.br)

<sup>III</sup> Universidade Federal de Rondonópolis (UFR), Rondonópolis-MT, Brasil; [julia@ufr.edu.br](mailto:julia@ufr.edu.br)

<sup>IV</sup> Universidade do Vale do Itajaí (Univali), Itajaí-SC, Brasil; [raabe@univali.br](mailto:raabe@univali.br)

## RESUMO

O pensamento computacional é uma competência ligada à resolução de problemas que mobiliza o pensamento algorítmico, a identificação de padrões, a abstração e a modularização de problemas e suas soluções. Esse tema tem sido incorporado em currículos escolares de vários países, incluindo o Brasil, mas são poucos os instrumentos de mensuração validados. Este artigo trata da busca de evidências de validade do CT Puzzle Test, um instrumento de diagnóstico do pensamento computacional baseado em *puzzles*. Foi avaliada a validade convergente com um teste que já tem evidências de validade estabelecidas. Os resultados confirmam a convergência entre os instrumentos. Por fim, foi aprimorado o cálculo do escore do CT Puzzle Test e evidenciada a validade do uso de *puzzles* como instrumento de avaliação.

**PALAVRAS-CHAVE** RESOLUÇÃO DE PROBLEMAS • AVALIAÇÃO DA APRENDIZAGEM • INSTRUMENTOS DE AVALIAÇÃO • VALIDADE DE TESTES.

Como citar: CALBUSCH, Leonardo Felipe de Ávila; COUTO, Natália Ellery Ribeiro; ROCCA, Julia Zanetti; RAABE, André Luís Alice. Aprimoramento do CT Puzzle Test para avaliação do pensamento computacional. *Estudos em Avaliação Educacional*, São Paulo, v. 33, e08938, 2022. DOI: <https://doi.org/10.18222/eae.v33.8938>

# MEJORA DEL CT PUZZLE TEST PARA EVALUAR EL PENSAMIENTO COMPUTACIONAL

## RESUMEN

El pensamiento computacional es una competencia conectada a la resolución de problemas que moviliza el pensamiento algorítmico, la identificación de estándares, la abstracción y la modularización de problemas y sus soluciones. Este tema ha sido incorporado a currículos escolares de varios países, incluso de Brasil, pero hay pocos instrumentos válidos para medirlo. Este artículo trata de la búsqueda de evidencias de validez del CT Puzzle Test, un instrumento de diagnóstico del pensamiento computacional basado en *puzzles*. La validez convergente se evaluó con otro test que ya ha establecido evidencia de validez. Los resultados confirman la convergencia entre los instrumentos. Además, se mejoró el cálculo del score del CT Puzzle Test y se evidenció la validez del uso de *puzzles* como instrumento de evaluación.

**PALABRAS CLAVE** RESOLUCIÓN DE PROBLEMAS • EVALUACIÓN DEL APRENDIZAJE • INSTRUMENTOS DE EVALUACIÓN • VALIDEZ DE TESTS.

# IMPROVEMENT OF THE CT PUZZLE TEST FOR COMPUTATIONAL THINKING ASSESSMENT

## ABSTRACT

Computational thinking is a problem-solving skill that involves algorithmic thinking, pattern identification, abstraction, and the modularization of problems and their solutions. This subject has been incorporated into school curricula in several countries, including Brazil, but there are few instruments validated to measure it. The present article deals with searches for evidence of the validity of the CT Puzzle Test, a puzzle-based tool to diagnose computational thinking. Convergent validity was verified with another test that already has the evidence of its validity established. The results confirm the convergence between the instruments. In addition, the score calculation for the CT Puzzle Test items was improved and the validity of adopting puzzles as an assessment tool was evidenced.

**KEYWORDS** PROBLEM-SOLVING • LEARNING ASSESSMENT • ASSESSMENT INSTRUMENTS • TEST VALIDITY.

Recebido em: 12 AGOSTO 2021

Aprovado para publicação em: 5 SETEMBRO 2022



Este é um artigo de acesso aberto distribuído nos termos da licença Creative Commons do tipo BY-NC.

## INTRODUÇÃO

O pensamento computacional (PC) vem sendo estudado nos últimos anos sob diferentes perspectivas, e várias iniciativas têm surgido no intuito de implementar o PC em currículos escolares de diversos países. No entanto, a comunidade científica ainda não adotou uma definição formal congruente para o PC.

Nos últimos anos, o PC vem sendo apresentado como um conjunto de habilidades para solução de problemas que deve ser desenvolvido pelas novas gerações de estudantes (BOCCONI *et al.*, 2016), uma percepção já defendida na década de 1970 por Papert (1971). Jeanette Wing (2006) afirmou que o PC envolve a solução de problemas, o design de sistemas e a compreensão do comportamento humano, baseando-se nos conceitos fundamentais da ciência da computação. Desde então, apesar de os especialistas concordarem a respeito da relevância do desenvolvimento do PC para a formação do indivíduo, sua definição ainda não é consenso entre os pesquisadores da área (KALELIOGLU; GÜLBAHAR; KUKUL, 2016). A partir de todas essas discussões, neste artigo é adotada a definição de pensamento computacional como sendo uma extensão da capacidade humana de resolução de problemas, que se fundamenta nos conceitos e práticas relacionados à construção, ao uso e à avaliação de algoritmos, no reconhecimento de padrões, na decomposição de problemas e na abstração.

Com a popularização do termo “pensamento computacional”, as discussões sobre a maneira de introduzi-lo na educação básica se ampliaram. Segundo Raabe, Zorzo e Blikstein (2020), diversas iniciativas de introdução ao PC têm sido realizadas nos últimos anos, envolvendo pesquisadores de escolas e universidades em diferentes níveis educacionais. Para implementar com êxito o ensino de PC nas escolas, no entanto, é imprescindível projetar instrumentos e métodos de avaliação adequados para diagnosticar e acompanhar o desempenho dos estudantes (KONG; ABELSON; LAI, 2019). A validação de instrumentos de avaliação do PC é extremamente relevante e urgente porque, sem ferramentas de avaliação confiáveis e válidas, o PC pode perder seu potencial de ingressar nos currículos educacionais (ROMÁN-GONZÁLEZ; PÉREZ-GONZÁLEZ; JIMÉNEZ-FERNÁNDEZ, 2017).

Nos Estados Unidos, a Computer Science Teachers Association (CSTA) e a International Society for Technology in Education (ISTE) trabalharam de forma colaborativa para desenvolver materiais que possam auxiliar professores no ensino do PC. Apresentaram, então, uma definição operacional do PC com os seguintes conceitos computacionais: coleta, análise e representação de dados; decomposição de problemas; abstração; algoritmos e procedimentos; automação; simulação e paralelismo (INTERNATIONAL SOCIETY FOR TECHNOLOGY IN EDUCATION – ISTE; COMPUTER SCIENCE TEACHERS ASSOCIATION – CSTA, 2011). Os conceitos apresentados representam características que um indivíduo deve demonstrar no

processo de resolução de problemas. Brackmann (2017) apresenta uma pesquisa em que aponta organizações que são referência em computação no mundo, e estas concordam com a definição de quatro conceitos fundamentais considerados pilares para o PC:

- Decomposição: quebrar um problema ou sistema complexo em partes menores e mais propensas a serem solucionadas.
- Reconhecimento de padrão: procurar similaridades entre problemas, assim como dentro deles.
- Abstração: foco nas informações importantes, ignorando as irrelevantes.
- Algoritmos: desenvolver um conjunto de passos para a solução de um problema, ou regras para solucionar um problema.

O mesmo autor afirma que o PC é assimilado no dia a dia das pessoas, ainda que sua presença nem sempre seja percebida. Nesse sentido, desenvolver os pilares do PC durante a educação básica pode auxiliar na atitude do indivíduo ao se deparar com um problema. Ainda de acordo com Brackmann (2017), o PC é uma habilidade que qualquer pessoa deve ter, independentemente de sua área de conhecimento ou da atividade profissional.

Nas escolas básicas, a maioria das abordagens de desenvolvimento dos pilares do PC adota o ensino de programação, seja em disciplinas específicas do currículo escolar, seja por atividades extracurriculares, *workshops* ou minicursos. Para avaliar os pilares do PC desenvolvidos por essas abordagens, são utilizados instrumentos que realizam uma avaliação fechada por meio da análise estática de projetos desenvolvidos pelos estudantes, como o Dr. Scratch (MORENO-LEÓN; ROBLES; ROMÁN-GONZÁLEZ, 2015), Progression of Early Computational Thinking (PECT) (SEITER; FOREMAN, 2013), CodeMaster (WANGENHEIM *et al.*, 2018) e Quizly (MAIORANA; GIORDANO; MORELLI, 2015), entre outros. Esse tipo de avaliação apresenta algumas limitações. Para desenvolver seus projetos, os estudantes precisam de conhecimento prévio do ambiente ou da linguagem utilizada, com todas as particularidades técnicas. Além disso, esses instrumentos geralmente atuam no código-fonte dos projetos para avaliar o desempenho, normalmente somente depois de o estudante terminar o trabalho.

Outras iniciativas utilizam instrumentos na forma de questionários de múltipla escolha, como o Commutative Assessment Test (WEINTROP; WILENSKY, 2015), o Test for Measuring Basic Programming Abilities (MÜHLING; RUF; HUBWIESER, 2015), o Israel's Nationwide Exam (ZUR-BARGURY; PÂRV; LANZBERG, 2013) e o Bebras Tasks (DAGIENE; FUTSCHEK, 2008). Esse tipo de teste pontua somente as respostas dos estudantes a questões previamente definidas, sem avaliar o processo cognitivo necessário para a resolução de cada atividade.

Uma abordagem mais abrangente do que essas metodologias fechadas é a utilização de *puzzles*, um método de ensino baseado em projetos e problemas que desenvolve o pensamento crítico e o raciocínio lógico, independentemente de um domínio específico (FALKNER; SOORIAMURTHI; MICHALEWICZ, 2010; MERRICK, 2010; MICHALEWICZ; MICHALEWICZ, 2008). A pesquisa na literatura não apresentou instrumentos de avaliação do PC que utilizassem essa abordagem.

A abordagem de aprendizado baseada em *puzzles* (*puzzle-based learning* – PBL), segundo Falkner, Sooriamurthi e Michalewicz (2010), tem o objetivo de incentivar os estudantes a pensar em como conceber e resolver problemas não estruturados (aqueles que não são encontrados ao final de um capítulo de livro). O objetivo é motivá-los e aumentar a consciência matemática e as habilidades de resolução de problemas, induzindo-os a discutir estratégias de solução. De acordo com os mesmos autores, essa abordagem deve satisfazer alguns critérios:

- Independência de contexto: *puzzles* educacionais não devem estar especificamente vinculados a um único domínio de solução de problemas.
- Generalidade: *puzzles* educacionais desenvolvem princípios universais de solução de problemas, ou seja, o treinamento da habilidade de resolução de problemas deve permitir ao estudante resolver problemas ainda desconhecidos no futuro.
- Simplicidade: *puzzles* educacionais devem ser fáceis de declarar e lembrar, pois isso aumenta a probabilidade de os alunos se lembrarem do método de solução para aplicar em outros problemas no futuro.
- Fator *eureka*: *puzzles* educacionais devem inicialmente frustrar o solucionador de problemas, mas manter a promessa de resolução. Os solucionadores de problemas geralmente usam a intuição para iniciar sua busca pela solução e, eventualmente, alcançam um “momento *eureka*” – quando reconhecem o caminho correto para resolver o *puzzle* e experimentam uma sensação de recompensa por sua esperteza.
- Fator de entretenimento: *puzzles* educacionais devem ser divertidos e envolventes. O entretenimento costuma ser resultado da combinação da simplicidade, frustração, fator *eureka* e um cenário interessante, com algum contexto lúdico.

A abordagem por *puzzles* desenvolve habilidades de pensamento crítico e resolução de problemas, além de estabelecer uma conexão entre a capacidade de resolver *puzzles* e a capacidade de resolver problemas do mundo real. Muitos problemas do mundo real podem ser percebidos como *puzzles* em grande escala, pois, ao se deparar com um problema no trabalho ou no cotidiano, o indivíduo deve elaborar hipóteses e testar estratégias para resolvê-lo. Nessa perspectiva, ao analisar um

questionário objetivo, os estudantes deveriam ter a possibilidade de testar respostas com base em hipóteses experimentadas, ao invés de ter apenas uma chance de acerto que pune uma tentativa sem sucesso.

Nesse contexto, torna-se relevante a proposição de um método eficiente de avaliação do PC baseado em *puzzles*. A validação de diferentes métodos de avaliação e medição implicam definir operacionalmente o PC como construto. Portanto, avanços na avaliação contribuem para a consolidação do PC como um conceito sólido, pertinente a ser estudado e desenvolvido. A validade refere-se ao fato de um instrumento medir exatamente o que se propõe a medir (ROBERTS; PRIEST; TRAYNOR, 2006; PASQUALI, 2017). Rodrigues (2006) argumenta que deve haver um cuidado para que o teste inclua itens que cubram tão somente o conteúdo a ser avaliado e que revele os processos usados pelo educando para fazê-lo.

O presente artigo apresenta uma etapa de aprimoramento do Computational Thinking Puzzle Test – CT Puzzle Test (GONÇALVES, 2015; COUTO, 2018). Os diferenciais desse instrumento com relação aos instrumentos já existentes residem na avaliação por meio de *puzzles*, que estimulam o raciocínio lógico e o pensamento crítico, e se caracterizam pela independência de domínio e conhecimento prévio. O CT Puzzle Test avalia o PC por meio da captura das interações do estudante com o instrumento (número de tentativas, quantidade de cliques, tempo de resolução, quantidade de dicas utilizadas, etc.). A partir dessas interações, é realizado o diagnóstico sobre o desenvolvimento do PC no estudante, considerando suas tentativas, erros e acertos até chegar a uma resposta correta. Este trabalho apresenta a constituição de um modelo para análise dos dados de aplicação do CT Puzzle Test a partir da comparação com os resultados do teste Computational Thinking Test – CTt – (ROMÁN-GONZÁLEZ, 2015) em relação aos quatro pilares definidores do PC (Brackman, 2017). A pesquisa ora relatada investiga evidências da validade convergente entre o CT Puzzle Test e o CTt. As motivações em constatar essas evidências foram reafirmar a eficácia do CT Puzzle Test em avaliar o PC e demonstrar que os itens no formato de *puzzles* e a contemplação da experimentação e teste de hipóteses por meio da captura das interações do estudante com o instrumento constituem uma abordagem válida. Os resultados devem corroborar que o CT Puzzle Test converge com o CTt, que é um teste já em estado avançado de validação.

## **BACKGROUND - AVALIAÇÃO DO PENSAMENTO COMPUTACIONAL**

Ainda não há consenso sobre quais processos de pensamento específicos fazem parte do PC. Muitos autores apontam processos, habilidades e características que o indivíduo deve ter, mas falta um delineamento para se afirmar em que ponto o PC é ciência da computação ou uma habilidade matemática e/ou lógica. Por esse

motivo existe a dificuldade no desenvolvimento de um instrumento de avaliação que possa ser utilizado de forma abrangente para medir o desenvolvimento do PC de um indivíduo.

O ensino do PC na educação básica tem recebido atenção em vários países. Muitas das iniciativas existentes utilizam o ensino de programação como ferramenta de apoio ao desenvolvimento dos pilares do PC. Nesse contexto, normalmente são adotadas linguagens de programação baseadas em blocos visuais, como Scratch (<https://scratch.mit.edu>), Snap! (<http://snap.berkeley.edu>), App Inventor (<http://appinventor.mit.edu/explore>), entre outras. Existem, porém, poucas iniciativas para avaliar os pilares do PC no contexto da educação básica (WANGENHEIM *et al.*, 2018). Na maioria dos casos, a avaliação se dá por meio de tarefas de programação usando indicadores relacionados à verificação de assertividade e eficiência de código (SRIKANT; AGGARWAL, 2014; SMITH; CORDOVA, 2005) e/ou estilo de programação (SMITH; CORDOVA, 2005).

Alguns instrumentos de avaliação são adotados na forma de extensões das linguagens de programação, avaliando o código escrito pelo estudante. Alguns exemplos para a linguagem Scratch são o Dr. Scratch (MORENO-LEÓN; ROBLES; ROMÁN-GONZÁLEZ, 2015) ou o Ninja Code Village (OTA; MORIMOTO; KATO, 2016). Essas ferramentas adotam análise de código estático e avaliam tipo e número de blocos usados no programa, quantificando conceitos e práticas de PC, como abstração, lógica, fluxo de controle, etc. Outras ferramentas avaliam projetos de programação em outras linguagens baseadas em blocos, como o Snap! Autograder (BALL; GARCIA, 2016) ou App Inventor Quizly (MAIORANA; GIORDANO; MORELLI, 2015). O foco da maioria dessas abordagens está na avaliação de problemas fechados, com uma solução correta previamente conhecida. Esse tipo de avaliação apresenta algumas limitações, em primeiro lugar, porque as ferramentas geralmente funcionam no código-fonte, normalmente somente depois de o aluno finalizar a atividade. Esse foco no código-fonte também limita a avaliação, de forma que não cobre práticas essenciais do PC, como criatividade e colaboração, e em alguns casos não fornece suporte imediato para o estudante acompanhar seu desempenho durante a resolução das atividades (ROBLES *et al.*, 2018). Em segundo lugar, porque o estudante precisa de treinamento no ambiente de programação adotado, fundamentação técnica e familiaridade com a linguagem utilizada, conhecimentos prévios que são necessários para o acesso à avaliação do PC que influenciam no desempenho do estudante que realiza os testes.

Román-González, Moreno-León e Robles (2019) fazem uma análise dos instrumentos de avaliação do PC conforme sua abordagem avaliativa, e os classifica em sete modalidades:

- Instrumentos de diagnóstico: visam a medir o nível de aptidão para o PC do sujeito. Podem ser administrados em condição de pré-teste em indivíduos sem nenhuma experiência prévia em programação, e também na condição pós-teste, após uma intervenção educacional, para verificar se a aptidão para o PC foi desenvolvida.
- Instrumentos sumativos: aplicados depois de o indivíduo ser exposto a conteúdo e treinamento. Seu principal uso é na condição pós-teste e está vinculado ao ambiente de aprendizado utilizado (linguagem de programação ou conteúdo administrado).
- Instrumentos formativo-iterativos: visam a fornecer *feedback* imediato ao estudante, geralmente de maneira automática, a fim de desenvolver e melhorar suas habilidades. Essas ferramentas não avaliam os indivíduos, mas seus produtos de aprendizado, geralmente projetos de programação. Portanto, essas ferramentas são usadas principalmente durante o processo de aprendizado e são projetadas para um ambiente de programação específico.
- Instrumentos de mineração de dados: focados no processo de aprendizado, recuperam e registram a atividade do estudante durante a execução de uma tarefa. Fornecem dados e análises de aprendizado a partir dos quais os processos cognitivos do sujeito podem ser inferidos.
- Instrumentos de transferência de habilidades: seu objetivo é avaliar até que ponto os estudantes são capazes de transferir suas habilidades de PC para diferentes tipos de problemas, contextos e situações. Normalmente são avaliações textuais compostas de perguntas relacionadas a contextos do mundo real, e o sujeito deve utilizar conceitos do PC para respondê-las corretamente.
- Escalas de percepções e atitudes do PC: visam a avaliar as percepções (por exemplo, de autoeficácia) e atitudes dos sujeitos, não apenas sobre o PC, mas também acerca de questões relacionadas, como computadores, ciência da computação, programação de computadores ou mesmo alfabetização digital.
- Avaliação de vocabulário de PC: essas ferramentas pretendem medir vários elementos e dimensões do PC, quando expressas verbalmente pelos sujeitos.

No Brasil, observam-se algumas iniciativas de inclusão do PC no currículo escolar. Rodriguez, Reis e Isotani (2017) apresentam o projeto “Computação e Criatividade: criando jogos digitais usando o Scratch”. O projeto levou estudantes de escolas públicas ao laboratório com o objetivo de estimular o raciocínio lógico para a resolução de problemas por meio da programação de jogos digitais utilizando a ferramenta Scratch. Araujo, Primo e Pernas (2020) adotaram a ferramenta Dr. Scratch para realizar uma avaliação automatizada de projetos em Scratch. A partir dos resultados dessa ferramenta, os autores desenvolveram o OntoScratch,



uma representação de conhecimento baseada em ontologias. Esse método permite o acompanhamento da evolução do estudante, por meio da execução de análises temporais e da utilização de modelos de inferência.

França e Silva (2020) identificaram critérios pragmáticos para evidenciar as habilidades de PC. Por meio de uma síntese qualitativa dos resultados da pesquisa, apontaram um conjunto de 16 critérios de avaliação para cada uma das dimensões do PC. Raabe *et al.* (2017) desenvolveram o CT Puzzle Test, um instrumento de diagnóstico do PC. O instrumento dos autores foca nos elementos constituintes do pensamento computacional, os quais são mensurados pelo desempenho do estudante em atividades de resolução de problemas, planejadas para que não sejam necessários conhecimentos prévios de conceitos computacionais.

O esforço brasileiro pelo desenvolvimento de uma metodologia confiável de avaliação do PC também é evidenciado em trabalhos multinacionais envolvendo pesquisadores brasileiros como Brackmann *et al.* (2017). Os autores apontam melhoria significativa no desempenho de estudantes que vivenciaram atividades de pensamento computacional desplugado e utilizam o instrumento Computational Thinking Test como forma de avaliar comparativamente grupos de estudantes em uma abordagem quase-experimental. Cordenonzi, Del Pino e Cardoso (2021) utilizaram um modelo de referência de pensamento computacional e um método de avaliação. A partir desse método foi possível classificar os sujeitos em: (I) alfabetizados em código; ou (II) pensador computacional desplugado. A abordagem foi validada em uma turma de 20 estudantes brasileiros e uruguaios que participaram de um curso de 20 horas sobre o tema.

Vários pesquisadores têm conduzido revisões sistemáticas da literatura para investigar formas de avaliação do PC. O Quadro 1 sintetiza os instrumentos investigados em três pesquisas (COUTO, 2018; WANGENHEIM *et al.*, 2018; ROMÁN-GONZÁLEZ; MORENO-LEÓN; ROBLES, 2019):

**QUADRO 1 - Levantamento de instrumentos de avaliação do pensamento computacional**

INSTRUMENTO	MODALIDADE DO INSTRUMENTO	MODELO DE AVALIAÇÃO	CONHECIMENTO PRÉVIO
Computational Thinking Test (ROMÁN-GONZÁLEZ, 2015; ROMÁN-GONZÁLEZ <i>et al.</i> , 2017)	Diagnóstico	Questionário de múltipla escolha/ Perguntas compostas por desafios lógicos	Independente
CT Puzzle Test (GONÇALVES, 2015; COUTO, 2018)	Misto	<i>Puzzles</i> apoiados em pilares do PC, no estilo arrastar e soltar, graduação com base no registro de atividades	Independente
Scrape (WOLZ <i>et al.</i> , 2011)	Sumativo	Graduação de código-fonte concluído	Scratch

(continua)

(continuação)

INSTRUMENTO	MODALIDADE DO INSTRUMENTO	MODELO DE AVALIAÇÃO	CONHECIMENTO PRÉVIO
Fairy Assessment (WERNER <i>et al.</i> , 2012)	Sumativo	Gradação da <i>performance</i> em tarefas desenvolvidas com a ferramenta Alice após treinamento, questionário pré e pós-teste	Alice
Combined taxonomy tests (MEERBAUM-SALANT <i>et al.</i> , 2013)	Sumativo	Aplicação de questionário em pré, interim e pós-teste, após treinamento na ferramenta Scratch	Scratch
Israel's Nationwide Exam (ZUR-BARGURY; PÂRV; LANZBERG, 2013)	Sumativo	Questionário de múltipla escolha/ Perguntas compostas por trechos de código	Scratch
Hairball (FRANKLIN <i>et al.</i> , 2013; BOE <i>et al.</i> , 2013)	Sumativo	Gradação de código-fonte concluído	Scratch
Progression of Early Computational Thinking (SEITER; FOREMAN, 2013)	Sumativo	Gradação de código-fonte concluído	Scratch
Commutative Assessment Test (WEINTROP; WILENSKY, 2015)	Sumativo	Questionário de múltipla escolha/ Perguntas compostas por trechos de código	Snap!/Java
Test for Measuring Basic Programming Abilities (MÜHLING; RUF; HUBWIESER, 2015)	Sumativo	Questionário de Múltipla Escolha/ Perguntas compostas por trechos de código	Linguagem Karel
Quizly (MAIORANA; GIORDANO; MORELLI, 2015)	Sumativo	Plataforma de criação de perguntas ou tarefas no ambiente App Inventor com <i>feedback</i> para o estudante em tempo real e registro de atividades	App Inventor
Autograder (BALL; GARCIA, 2016; BALL, 2017)	Sumativo	Gradação de código-fonte concluído	Snap!
ITCH (JOHNSON, 2016)	Sumativo	Gradação de código-fonte concluído	Scratch
Computational Thinking Patterns CTP-Graph (KOH <i>et al.</i> , 2010)	Formativo-iterativo	<i>Feedback</i> para o estudante por avaliação de código-fonte concluído	AgentSheets
REACT (KOH <i>et al.</i> , 2014)	Formativo-iterativo	<i>Feedback</i> para o estudante por avaliação de código-fonte em tempo real	AgentSheets
Dr. Scratch (MORENO-LEÓN; ROBLES; ROMÁN-GONZÁLEZ, 2015)	Formativo-iterativo	<i>Feedback</i> para o estudante por avaliação de registro de atividade	Scratch
Ninja Code Village (OTA; MORIMOTO; KATO, 2016)	Formativo-iterativo	<i>Feedback</i> para o estudante por avaliação de código-fonte concluído	Scratch
Code Master (WANGENHEIM <i>et al.</i> , 2018)	Formativo-iterativo	<i>Feedback</i> para o estudante por avaliação de código-fonte concluído	App Inventor/Snap!
Shuchi Grover (GROVER <i>et al.</i> , 2017; GROVER <i>et al.</i> , 2016)	Mineração de dados	Estatística por avaliação de registro de atividade	Fairy Assessment/ Alice
Kodetu (EGUILUZ <i>et al.</i> , 2017)	Mineração de dados	Estatística por avaliação de registro de atividade	Kodetu/Blockly
Bebras Tasks (DAGIENE; FUTSCHEK, 2008)	Transferência de habilidades	Questionário de múltipla escolha/ Perguntas compostas por desafios lógicos	Independente

(continua)

(continuação)

INSTRUMENTO	MODALIDADE DO INSTRUMENTO	MODELO DE AVALIAÇÃO	CONHECIMENTO PRÉVIO
CTP-Quiz (BASAWAPATNA <i>et al.</i> , 2011)	Transferência de habilidades	Questionário com perguntas sobre a relação entre eventos reais e eventos programados em jogos predefinidos desenvolvidos pelos estudantes	AgentSheets
Dancing Alice (DAILY <i>et al.</i> , 2014)	Transferência de habilidades	Modelagem de movimentos de dança em personagem no ambiente Alice	Alice
Computational Thinking Scales – CTS (KORKMAZ <i>et al.</i> , 2017)	Escalas de percepções e atitudes	Escala de avaliação sobre a autopercepção do sujeito com relação a conceitos do PC	Independente
Computer Programming Self-Efficacy Scale – CPSES (KUKUL <i>et al.</i> , 2017)	Escalas de percepções e atitudes	Escala de avaliação sobre a autopercepção do sujeito com relação a conceitos de programação	Scratch/Smallbasic
Robotics and Engineering Workshop (GROVER, 2011)	Avaliação de vocabulário	Entrevistas em pré e pós-teste após treinamento	Linguagem Logo
CT Framework (BRENNAN; RESNICK, 2012)	Misto	Gradação de avaliação de código-fonte concluído complementada com entrevistas e tarefas de interpretação de código-fonte	Scratch

Fonte: Elaboração dos autores com base em Román-González, Moreno-León e Robles (2019), Wangenheim *et al.* (2018) e Couto (2018).

Cada instrumento aborda a avaliação do PC sob um aspecto diferente. Brennan e Resnick (2012) afirmaram que avaliar as competências computacionais apenas olhando para os programas criados pelos alunos poderia ser visivelmente insuficiente, por isso enfatizaram a necessidade de múltiplos meios de avaliação. Grover (2015) afirma que diferentes tipos de ferramentas complementares de avaliação devem ser sistematicamente combinados para alcançar uma compreensão total e abrangente do PC dos estudantes.

Nesse contexto, o CT Puzzle Test se destaca por combinar algumas das abordagens de avaliação indicadas por Román-González; Moreno-León e Robles (2019) – diagnóstico, formativo-iterativo, sumativo, mineração de dados – e por ser independente de linguagem/ferramenta. O Quadro 2 apresenta como cada abordagem se caracteriza no teste.

#### QUADRO 2 – Características das abordagens de avaliação no CT Puzzle Test

ABORDAGEM DE AVALIAÇÃO	CARACTERÍSTICAS
Diagnóstico	Mede o nível de aptidão do PC por meio de escores e pode ser aplicado em formato pré-teste pois independe de conhecimentos prévios.
Formativo-iterativo	Fornece <i>feedback</i> imediato e automático sobre acertos e erros; fornece dicas de resolução dos problemas.

(continua)

(continuação)

ABORDAGEM DE AVALIAÇÃO	CARACTERÍSTICAS
Sumativo	Pode ser aplicado após o indivíduo passar por instrução (uma disciplina de PC, por exemplo) para medir seu desenvolvimento.
Mineração de dados	Registra as interações do indivíduo com o instrumento e permite a análise do pesquisador sobre essas interações (como tempo, tentativas, utilização de dicas, etc.), e não somente os acertos e erros.

Fonte: Elaboração dos autores.

Além dessas características, o instrumento adota o estilo de pergunta em formato de *puzzle* ao invés da escolha entre múltiplas respostas apresentadas em questionários tradicionais.

Para estimar a validade do instrumento, Gonçalves (2015) conduziu as seguintes aplicações do teste: para a validade aparente e de conteúdo, o pesquisador utilizou amostras de especialistas do Grupo de Informática na Educação (GIE) da Universidade do Vale do Itajaí (Univali), composto por 22 integrantes, sendo: 11 deles doutores, mestres, doutorandos e mestrandos nas áreas da computação, educação, matemática e psicologia; e 11 estudantes do 1º e 2º ano do ensino médio do Colégio de Aplicação da Universidade e uma turma do 6º período do curso superior de ciência da computação da universidade. Para a validade concorrente, as amostras foram compostas por estudantes do 7º período do curso superior de ciência da computação, do 6º período do curso superior de engenharia da computação e do 1º período do curso superior de psicologia. Para a validade preditiva, 16 estudantes do 9º ano do ensino fundamental da rede municipal de ensino foram expostos ao instrumento. Para a estimativa de fidedignidade foi utilizado o método da consistência interna com o cálculo do coeficiente alpha Cronbach para todas as aplicações do teste. Os métodos de validade convergente e discriminante não foram abordados pelo fato de não haver outro instrumento à época para comparação de resultados. Os esforços de Gonçalves (2015) na investigação da validade e fidedignidade do CT Puzzle Test indicaram evidências fortes de validade do instrumento, no entanto, o próprio autor observa que, para fins estatísticos, a aplicação com mais indivíduos se faz necessária.

## MATERIAIS E MÉTODOS

Para verificar a validade convergente do CT Puzzle Test, foi adotado como critério o CTt (ROMÁN-GONZÁLEZ, 2015). Essa etapa não havia sido executada à época do primeiro processo de validação do CT Puzzle Test, por falta de outro instrumento semelhante para ser utilizado como critério, conforme apontou Gonçalves (2015). A escolha do CTt para essa nova etapa de validação se deu por duas razões principais. A primeira é que o CTt foi o teste com mais características em comum com o CT Puzzle Test – é um instrumento que não depende de conhecimento prévio do sujeito,

de linguagem de programação ou tecnologia, pode ser facilmente aplicado por ser um instrumento *on-line* e gera um escore que pode ser comparado com os resultados do CT Puzzle Test. A segunda razão é que o CTt já passou por testes para confirmar sua validade e fidedignidade (ROMÁN-GONZÁLEZ, 2015; ROMÁN-GONZÁLEZ; MORENO-LEÓN; ROBLES, 2017).

### CT Puzzle Test

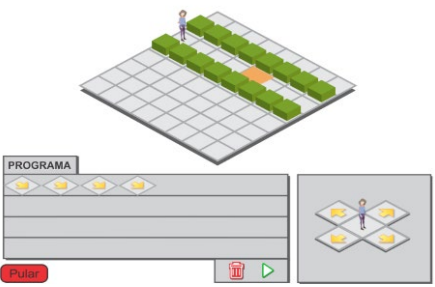
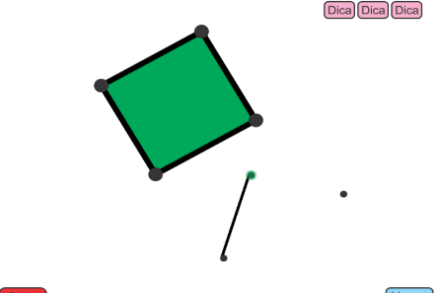
As questões criadas para o instrumento foram baseadas em atividades clássicas sobre resolução de problemas. Questões do Programa de Enriquecimento Instrumental de Feuerstein (2006) e exercícios encontrados em testes de QI foram adaptados para o formato de jogo *puzzle* para auxiliar a mensurar os pilares do PC.

O CT Puzzle Test é planejado para que não seja necessário nenhum conhecimento prévio para resolvê-lo. Os problemas estão organizados em fases que apresentam um mesmo tipo de mecânica/*puzzle*, porém com níveis diferentes. O instrumento foi projetado para ser aplicado *on-line* e utiliza as tecnologias HTML5 e Javascript.

### Mecânicas do CT Puzzle Test

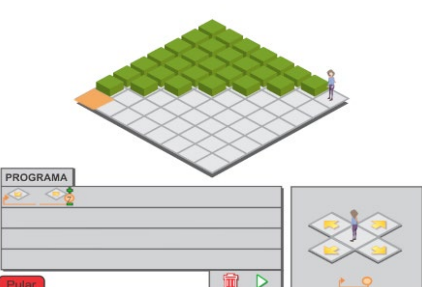
Em cada fase do teste, são capturadas variáveis de interação de uso. Na fase 1, por exemplo, são contadas quantas vezes um comando foi removido. A remoção de um comando nesse caso aponta que o aluno cometeu alguma falha e precisou corrigir. Isso tem influência na pontuação final. O Quadro 3 apresenta as fases do teste, com uma descrição de suas mecânicas e das interações avaliadas.

QUADRO 3 – Fases do CT Puzzle Test

ILUSTRAÇÃO	DESCRIÇÃO DA MECÂNICA
	<p><b>Fase 1:</b> o indivíduo deve programar o personagem para que este se movimente pelos quadros de um tabuleiro partindo de um ponto inicial até um ponto final.</p> <p><b>Interações avaliadas:</b> tempo despendido; quantidade de comandos utilizados; quantidade de utilizações da opção "remover um comando"; quantidade de utilizações da opção "remover todos os comandos"; e quantidade de execuções do programa.</p>
	<p><b>Fase 2:</b> baseada na atividade de conectar pontos de forma a criar o formato das figuras geométricas solicitadas.</p> <p><b>Interações avaliadas:</b> tempo despendido; quantidade de cliques nos pontos da tela; quantidade de utilizações da opção "limpar"; e quantidade de dicas utilizadas.</p>

(continua)

(continuação)

ILUSTRAÇÃO	DESCRIÇÃO DA MECÂNICA
	<p><b>Fase 3:</b> o indivíduo deve arrastar as peças menores para dentro da figura maior, encaixando-as nos espaços disponíveis, até que as peças se esgotem.</p> <p><b>Interações avaliadas:</b> tempo despendido; quantidade de cliques nas peças; quantidade de utilizações da opção "girar peça"; e quantidade de dicas utilizadas.</p>
	<p><b>Fase 4:</b> baseada no quebra-cabeça Tangram. O estudante deve selecionar uma peça, rotacioná-la se julgar necessário e arrastá-la para o lugar correto, repetindo esse processo até que a figura principal fique completa.</p> <p><b>Interações avaliadas:</b> tempo despendido; quantidade de cliques nas peças, quantidade de utilizações da opção "girar peça"; e quantidade de dicas utilizadas.</p>
	<p><b>Fase 5:</b> é necessário identificar qual das opções fornecidas corresponde à próxima figura da sequência apresentada, de acordo com o padrão observado nas figuras que já compõem a sequência.</p> <p><b>Interações avaliadas:</b> tempo despendido; quantidade de tentativas; e quantidade de dicas utilizadas.</p>
<p>Análise os objetos e os organize na tabela considerando suas características em comum. As linhas devem representar uma característica e as colunas outra.</p> 	<p><b>Fase 6:</b> o indivíduo deve encontrar a relação entre os objetos espalhados pela tela e organizá-los em linhas e colunas de uma tabela considerando suas características (cor e classe).</p> <p><b>Interações avaliadas:</b> tempo despendido; quantas vezes o estudante completou a tabela (quantidade de tentativas); quantidade de utilizações da opção "limpar"; e quantidade de dicas utilizadas.</p>
	<p><b>Fase 7:</b> muito similar à primeira, porém é acrescido o comando <i>loop</i>. Ao arrastar-se este comando para o bloco PROGRAMA, o estudante pode indicar quantas vezes um comando direcional deve ser repetido automaticamente.</p> <p><b>Interações avaliadas:</b> além dos itens já avaliados na fase 1, também são contabilizadas quantas vezes o comando <i>loop</i> foi inserido e a quantidade de comandos direcionais inseridos em cada <i>loop</i>.</p>

Fonte: Elaboração dos autores.

Em todos os itens das fases está disponível a opção “pular”, para que o estudante avance para o próximo item sem completar o atual, anulando a pontuação do item pulado.

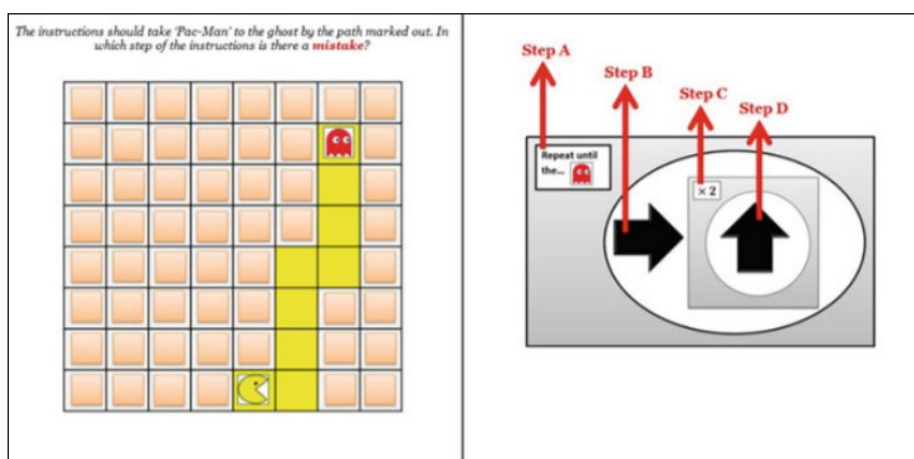
### Computational Thinking Test

O CTt (ROMÁN-GONZÁLEZ, 2015) é uma ferramenta de avaliação diagnóstica que consiste em um instrumento de múltipla escolha composto por 28 itens. Cada item aborda um ou mais dos seguintes conceitos computacionais, que aparecem em dificuldade crescente ao longo do teste: instruções e sequências básicas; loops (“repetir vezes”, “repetir até”, “repetir enquanto”); condicionais simples (“se”), condicionais compostas (“se/senão”) e funções simples.

Esses conceitos estão alinhados com os padrões de aprendizagem estabelecidos pela CSTA (2011) para o ensino da ciência da computação. Especificamente, em seu nível 2 (correspondente às séries 6 a 9 no sistema norte-americano), o padrão definido com relação aos conceitos e habilidades computacionais que devem ser dominados nessas idades se caracteriza pela implementação de soluções de problemas usando uma linguagem de programação, incluindo estruturas de repetição, estruturas condicionais, expressões lógicas, variáveis e funções (ISTE; CSTA, 2011).

Em cada item, as respostas são representadas em um dos dois estilos: “setas visuais” ou “blocos visuais”. Para ser resolvido, cada item exige ao sujeito uma das seguintes tarefas cognitivas: sequenciar um algoritmo, concluir um algoritmo incompleto ou depurar um algoritmo incorreto. A Figura 1 apresenta um exemplo de item do tipo “setas” que aborda os conceitos de “repetir até” e “repetir vezes”.

**FIGURA 1 - CTt, item #11: loops “repetir até + repetir vezes”; “setas visuais”**



Fonte: Román-González (2015).

Nesse item, o estudante precisa analisar no tabuleiro qual a sequência de passos que o personagem deve percorrer. Do lado direito, um aninhamento de comandos

de repetição determina a automatização do movimento do personagem. Porém uma das instruções está incorreta, e o estudante deve ser capaz de identificar o erro.

A Figura 2 apresenta um item do tipo “blocos” que aborda os conceitos “repetir até” e condicionais compostas “se/senão”.

**FIGURA 2 - CTt, item #18: loops “repetir até” + condicional “if/else”; “blocos visuais”**

Fonte: Román-González (2015).

Nesse item, novamente o estudante precisa interpretar o caminho que o personagem deve percorrer. Porém, nesse caso, as instruções são apresentadas em formato de bloco de comandos, e o movimento do personagem agora depende dos obstáculos no caminho, que devem ser evitados por meio das estruturas condicionais. Apenas um dos quatro algoritmos está correto, por isso o estudante deve depurar cada opção para identificar a correta.

A Figura 3 apresenta um exemplo de item do tipo “bloco” no qual é necessário concluir o algoritmo incompleto.

**FIGURA 3 - CTt, item #26: loops “repetir vezes” + funções simples; “blocos visuais”**

Fonte: Román-González (2015).



Nesse item, é proposto um padrão de desenho que o algoritmo deve produzir, porém falta a informação da quantidade de repetições em uma das etapas do programa. O estudante deve ser capaz de identificar o valor correto da repetição para garantir a execução do desenho, levando em consideração que esse algoritmo está estruturado de forma modular pela implementação do conceito de “funções”.

O teste é aplicado por meio de um formulário *on-line* e o escore final é obtido pelo cálculo da porcentagem do número de acertos em relação ao total de questões.

O CTt apresentou evidências de validade para avaliar o PC em indivíduos entre 10 e 16 anos. Seu processo de validação foi relatado por Román-González (2016), e algumas etapas desse processo foram descritas em publicações específicas, como a validação de conteúdo (ROMÁN-GONZÁLEZ, 2015), a validação de critério (ROMÁN-GONZÁLEZ; MORENO-LEÓN; ROBLES, 2017) e a validação preditiva (ROMÁN-GONZÁLEZ *et al.*, 2018).

### Aplicação dos testes

Para esta etapa da pesquisa, os dois testes foram aplicados em amostras de alunos do 8º a 9º ano do ensino fundamental e do 1º ao 3º ano do ensino médio do Colégio de Aplicação Univali, localizado no município de Itajaí, totalizando 86 estudantes em ambos os testes.

Os testes foram aplicados em laboratórios de informática, em horário regular de aula de cada turma com tempo disponível de 50 minutos para cada aplicação. O CT Puzzle Test foi aplicado no final do quarto bimestre de 2017, enquanto o CTt foi aplicado no final do segundo bimestre de 2018.

### Equiparação de critérios de avaliação

Em sua pesquisa, Brackmann (2017) também adotou o CTt como instrumento de avaliação do PC. Com o intuito de alinhar os resultados das aplicações do CTt com os objetivos da pesquisa, o autor relacionou cada item do CTt a pelo menos um dos pilares do PC, por meio da associação dos conceitos computacionais da CSTA, nos quais cada item está apoiado nos conceitos correspondentes no âmbito dos pilares. O Quadro 4 apresenta essa relação dos pilares com os itens do CTt.

**QUADRO 4 - Itens do CTt e sua relação com os pilares do PC**

ITEM	ABSTRAÇÃO	DECOMPOSIÇÃO	RECONHECIMENTO DE PADRÕES	ALGORITMOS
1	X			X
2	X			X
3	X			X

(continua)

(continuação)

ITEM	ABSTRAÇÃO	DECOMPOSIÇÃO	RECONHECIMENTO DE PADRÕES	ALGORITMOS
4		X	X	X
5		X	X	X
6		X	X	X
7	X	X		X
8			X	X
9			X	X
10		X	X	X
11	X	X	X	X
12	X	X	X	X
13	X	X		X
14	X		X	X
15	X	X	X	X
16				X
17			X	X
18			X	X
19				X
20			X	X
21	X	X		X
22	X	X	X	X
23	X	X	X	X
24				X
25	X	X	X	X
26	X	X	X	X
27	X	X	X	X
28	X	X	X	X

Fonte: Brackmann (2017).

Para comparar os dois testes na atual etapa de validação, os itens do CT Puzzle Test também foram mapeados para os pilares do PC. Para esse mapeamento foi aplicado um algoritmo que combina diversos valores atribuídos aos pesos das variáveis da equação do CT Puzzle Test. A partir dos resultados, foram selecionados os pesos que permitem alcançar a maior correlação possível entre os resultados dos dois testes. Para o elemento “algoritmo”, o maior coeficiente de correlação encontrado foi 0.55, para “abstração” foi 0.53, para “decomposição” foi 0.60, e para “reconhecimento de padrões” foi 0.58.

Finalmente, a distribuição de pesos em cada item, conforme a aderência do item a cada pilar, resultou nos valores apresentados na Tabela 1.

**TABELA 1 - Pesos em cada fase e respectivos itens do CT Puzzle Test, segundo a aderência aos pilares do PC**

FASE	PILARES DO PC			
	ABSTRAÇÃO	RECONHECIMENTO DE PADRÕES	DECOMPOSIÇÃO	ALGORITMOS
1	0.1	0.1	0.45	0.4
2	0.4	0.4	0.2	0
3	0.2	0.4	0.4	0
4	0.2	0.4	0.4	0
5	0.2	0.4	0.4	0
6	0.4	0.3	0.3	0
7	0.1	0.1	0.4	0.4

Fonte: Elaboração dos autores.

Ao mapear os resultados do CT Puzzle Test para os pilares do PC, foi realizado o cálculo do coeficiente alpha de Cronbach para cada pilar nos dois testes, visando a validar a distribuição dos pesos de cada fase do CT Puzzle Test. A Tabela 2 apresenta os resultados.

**TABELA 2 - Coeficiente alpha de Cronbach para cada pilar do PC e teste**

PILAR DO PC	CT PUZZLE TEST	CTt
Algoritmos	0.66	0.80
Abstração	0.61	0.67
Decomposição	0.75	0.69
Reconhecimento de padrões	0.66	0.74
Média	0.67	0.72

Fonte: Elaboração dos autores.

Ao calcular a média entre os valores é possível afirmar que são válidos os pesos aplicados no cálculo dos escores CT Puzzle Test, de acordo com o coeficiente alpha de Cronbach de 0.67, próximo ao valor 0.72 alcançado pelo CTt.

Com todos os itens dos dois testes mapeados para os pilares do PC, foi possível realizar a comparação estatística dos escores obtidos nas aplicações de ambos os testes.

## ANÁLISE DOS RESULTADOS

Os resultados das aplicações do CT Puzzle Test foram analisados com base na metodologia pelo mapeamento dos quatro pilares do PC sugeridas por Brackmann (2017)

e relatadas na subseção “Equiparação de Critérios de Avaliação”. As médias dos escores da amostra são apresentadas na Tabela 3.

**TABELA 3 – Médias do escores dos estudantes no CT Puzzle Test**

PILAR DO PC	MÉDIA	DESVIO PADRÃO
Algoritmos	0.67	0.09
Abstração	0.69	0.08
Decomposição	0.68	0.08
Reconhecimento de padrões	0.72	0.09

Fonte: Elaboração dos autores.

O cálculo da média a partir dos resultados revela que os estudantes pontuaram escores altos, variando entre 0.67 e 0.72 com desvio-padrão baixo, evidenciando que não houve muita variação nos escores.

Os resultados das aplicações do CTt também utilizaram a metodologia pelo mapeamento dos quatro pilares do PC de Brackmann. A Tabela 4 apresenta os valores das médias do escores da amostra de estudantes que foram submetidos ao CTt.

**TABELA 4 – Médias dos escores dos estudantes no CTt**

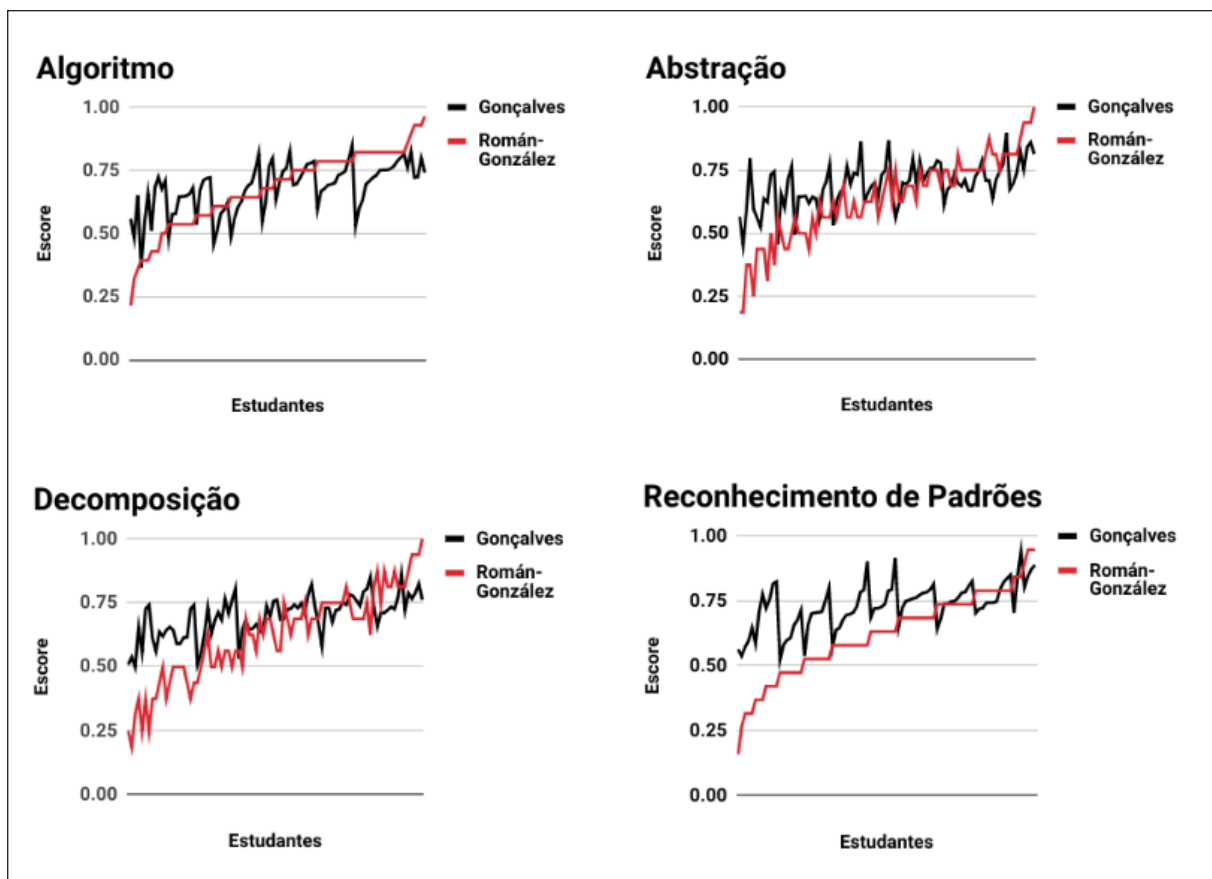
PILAR DO PC	MÉDIA	DESVIO PADRÃO
Algoritmos	0.66	0.17
Abstração	0.62	0.17
Decomposição	0.61	0.18
Reconhecimento de Padrões	0.61	0.18

Fonte: Elaboração dos autores.

As médias dos estudantes se situam entre 0.61 e 0.66, com desvio-padrão pequeno, mostrando pouca variação nas notas.

Para investigar evidências de validade convergente entre os testes, foi analisada a distribuição dos escores dos 86 estudantes que participaram de ambos os testes. A Figura 4 apresenta essa distribuição conforme cada pilar do PC.

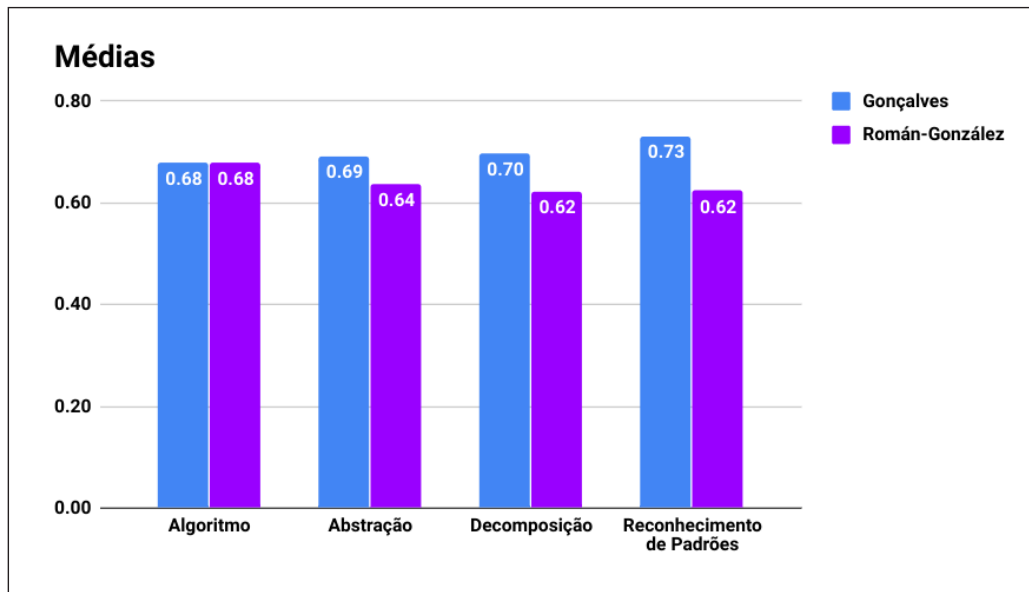
FIGURA 4 - Distribuição dos escores dos estudantes em cada teste conforme o pilar do PC



Fonte: Elaboração dos autores.

Nos gráficos da Figura 4, o eixo X representa os estudantes de maneira individual, e o eixo Y representa o escore que obtiveram no CT Puzzle Test (preto) e no CTt (vermelho). De acordo com a regra de Thumb, as correlações podem ser consideradas como moderadas positivas, pois os valores se posicionam entre 0.5 e 0.7 (HINKLE; WIERSMA; JURIS, 2003).

As médias da amostra de estudantes que realizaram os dois testes também foram calculadas. Conforme ilustra a Figura 5, é possível perceber que as médias são iguais na categoria “algoritmo”, e nas restantes elas são muito próximas, com diferença de apenas 0.05 ponto em “abstração”, 0.08 em “decomposição” e 0.11 em “reconhecimento de padrões”.

**FIGURA 5 – Médias dos escores dos estudantes da amostra em cada teste, segundo os pilares do PC**

Fonte: Elaboração dos autores.

Apesar de os dois testes medirem os mesmos pilares do PC, a forma de mensurá-los é diferenciada em cada teste. O CTt apresenta todos os desafios por meio de perguntas de múltipla escolha com quatro opções, enquanto o CT Puzzle Test analisa diversos fatores da resolução, como tempo, número de tentativas e tipos de interação. A abordagem do conteúdo também é diferenciada: o CTt foca exclusivamente em situações relacionadas à lógica de programação (ROMÁN-GONZÁLEZ; MORENO-LEÓN; ROBLES, 2019); o CT Puzzle Test, por sua vez, apresenta situações independentes de domínio como identificação de formas geométricas, progressão de sequências e reconhecimento de padrões, entre outros. O alpha de Cronbach do CT Puzzle Test aprimorado (0.67) é considerado de consistência interna substancial (LANDIS; KOCH, 1977), o que indica a confiabilidade do teste, e, ao apresentar uma correlação moderada com o CTt, aponta forte evidência de sua validade convergente. Portanto os resultados reafirmam a eficácia do CT Puzzle Test em avaliar o PC e demonstram que os itens no formato de *puzzles* e a contemplação da experimentação e teste de hipóteses por meio da captura das interações do estudante com o instrumento é uma abordagem válida. Contudo, conforme Bittencourt *et al.* (2011), apesar da validação do instrumento ser uma importante iniciativa para seu aperfeiçoamento, deve-se considerar que o instrumento continue sendo aprimorado em estudos futuros. De acordo com os autores, “[...] não existem instrumentos definitivos e perfeitos, especialmente em se tratando de avaliação educacional” (p. 21).

## CONCLUSÕES E PESQUISAS FUTURAS

A consolidação do pensamento computacional como um construto científico tem evoluído com pesquisas em vários lugares do mundo. A ideia de implementar o PC nas escolas como um conjunto de habilidades que todo estudante precisa desenvolver tem recebido atenção de educadores e legisladores, e é fundamental dispor de metodologias de ensino e avaliação adequadas para que o PC se estabeleça com êxito nos currículos escolares. Analisar apenas códigos-fonte de programas finalizados ou adotar questionários compostos de perguntas fechadas com respostas objetivas é insuficiente para avaliar as competências computacionais dos estudantes. Por isso, combinar múltiplas formas de avaliação se faz necessário para alcançar um entendimento completo das habilidades de pensamento computacional de um sujeito. Nesse contexto, este artigo apresentou os resultados da investigação acerca das evidências de convergência entre os testes CT Puzzle Test e CTt. Os resultados demonstraram que são fortes as evidências da eficácia do CT Puzzle Test em avaliar o PC e corroboraram que a abordagem por *puzzles* e a contemplação da experimentação e teste de hipóteses por meio da captura das interações do estudante com o instrumento é uma abordagem válida para a avaliação do PC. O principal trabalho futuro proposto é a submissão do CT Puzzle Test a um novo e mais completo processo de validação. Novas pesquisas são necessárias a partir das modificações do instrumento em suas últimas etapas de aprimoramento, e uma nova etapa de validação apoiada em novas amostras e em técnicas mais rigorosas será a principal contribuição ao estado atual de evolução do instrumento. Além disso, a sistematização de aplicação do teste, com a definição de diretrizes e documentação de apoio também se faz pertinente para disponibilizar o teste a qualquer pesquisador que tenha interesse em investigar o PC em sua área de atuação.

Outra possibilidade para trabalhos futuros, com base nas melhorias desenvolvidas nesta pesquisa, é projetar novos níveis para as fases existentes no CT Puzzle Test para se obter maior variabilidade dos testes, o que permitirá aplicar o teste mais de uma vez a um mesmo estudante sem o risco do viés de testagem, quando o indivíduo memoriza as respostas do pré-teste para responder o pós-teste.

Por fim, a evolução do CT Puzzle Test representará uma colaboração com a evolução da rede nomológica de desenvolvimento do PC, pois a constatação da existência de correlação entre instrumentos de avaliação que se propõem a avaliar um mesmo construto contribui para explicar até que ponto e por que os construtos teóricos se relacionam. Essas correlações resultam na definição das variáveis moderadoras nas quais elas podem ou não ser observadas, e dependem do desenvolvimento de medidas eficientes e válidas.

## REFERÊNCIAS

- ARAUJO, N. de; PRIMO, T. T.; PERNAS, A. M. OntoScratch: ontologias para a avaliação do ensino de Pensamento Computacional através do Scratch. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 31., 2020, on-line. *Anais [...]*. Porto Alegre: Sociedade Brasileira de Computação, 2020. p. 1823-1832. DOI: <https://doi.org/10.5753/cbie.sbie.2020.1823>
- BALL, M. A.; GARCIA, D. D. Autograding and feedback for snap! In: TECHNICAL SYMPOSIUM ON COMPUTING SCIENCE EDUCATION, 47., 2016, Memphis. *Proceedings [...]*. New York: ACM, 2016.
- BITTENCOURT, H. R.; CREUTZBERG, M.; RODRIGUES, A. C. de M.; CASARTELLI, A. de O.; FREITAS, A. L. S de. Desenvolvimento e validação de um instrumento para avaliação de disciplinas na educação superior. *Estudos em Avaliação Educacional*, São Paulo, v. 22, n. 48, p. 91-113, jan./abr. 2011. DOI: <https://doi.org/10.18222/eaec224820111994>
- BOCCONI, S.; CHIOCCARIELLO, A.; DETTORI, G.; FERRARI, A.; ENGELHARDT, K.; KAMPYLIS, P.; PUNIE, Y. *Developing computational thinking in compulsory education: implications for policy and practice*. Luxembourg: Publications Office of the European Union, 2016.
- BRACKMANN, C. P. *Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica*. 2017. Tese (Doutorado em Informática na Educação) – Centro Interdisciplinar de Novas Tecnologias na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017.
- BRACKMANN, C.; BOUCINHA, R.; ROMÁN-GONZÁLEZ, M.; BARONE, D.; CASALI, A. Pensamento computacional desplugado: ensino e avaliação na educação primária espanhola. In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 6., 2017, Recife. *Anais [...]*. Porto Alegre: Sociedade Brasileira de Computação, 2017. DOI: <http://dx.doi.org/10.5753/cbie.wcbie.2017.982>
- BRENNAN, K.; RESNICK, M. *New frameworks for studying and assessing the development of computational thinking*. Vancouver: American Educational Research Association, 2012.
- CORDENONZI, W. H.; DEL PINO, J. C.; CARDOSO, V. M. Analisando o desenvolvimento do Pensamento Computacional na disciplina Matemática Discreta. *Revista Brasileira de Informática na Educação*, Porto Alegre, v. 29, p. 880-902, 2021. DOI: <http://dx.doi.org/10.5753/rbie.2021.29.0.880>
- COUTO, N. E. R. *Impacto da implantação de uma disciplina de pensamento computacional em estudantes*. 2018. Dissertação (Mestrado em Computação Aplicada) – Universidade do Vale de Itajaí, Itajaí, SC, 2018.
- CSTA. *K-12 Computer Science Standards*. New York: ACM, 2011.
- DAGIENE, V.; FUTSCHEK, G. *Bebras international contest on informatics and computer literacy: criteria for good tasks*. Lecture notes in computer science. [S. l.]: Springer Berlin Heidelberg, 2008. p. 19-30.
- FALKNER, N.; SOORIAMURTHI, R.; MICHALEWICZ, Z. Puzzle-based learning for engineering and computer science. *Computer*, [S. l.], v. 43, n. 4, p. 20-28, abr. 2010. DOI: <https://doi.org/10.1109/MC.2010.113>
- FEUERSTEIN, R. *The Feuerstein Instrumental Enrichment Program*. Jerusalem: ICELP Publications, 2006.



- FRANÇA, C.; SILVA, C. G. Identificação de critérios para avaliação do pensamento computacional aplicado. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 31., 2020, on-line. *Anais [...]*. Porto Alegre: Sociedade Brasileira de Computação, 2020. p. 1493-1502. DOI: <https://doi.org/10.5753/cbie.sbie.2020.1493>
- GONÇALVES, F. A. *Um instrumento para diagnóstico do pensamento computacional*. 2015. Dissertação (Mestrado em Computação Aplicada) – Universidade do Vale de Itajaí, Itajaí, SC, 2015.
- GROVER, S. Systems of assessments for deeper learning of computational thinking in k-12. In: ANNUAL MEETING OF THE AMERICAN EDUCATIONAL RESEARCH ASSOCIATION, 2015, Chicago, IL. *Proceedings [...]*. Washington, DC: American Educational Research Association, 2015.
- HINKLE, D.; WIERSMA, W.; JURIS, S. *Applied statistics for the behavioral sciences*. Boston, MA: Houghton Mifflin, 2003.
- INTERNATIONAL SOCIETY FOR TECHNOLOGY IN EDUCATION – ISTE; COMPUTER SCIENCE TEACHERS ASSOCIATION – CSTA. *Computational thinking in k-12 teachers resources*. 2nd. ed. [S. l]: ISTE: CSTA, 2011. Disponível em: [http://code.intef.es/wp-content/uploads/2017/10/ct-teacher-resources\\_2ed-pdf.pdf](http://code.intef.es/wp-content/uploads/2017/10/ct-teacher-resources_2ed-pdf.pdf). Acesso em: 2 ago. 2021.
- KALELIOGLU, F.; GÜLBAHAR, Y.; KUKUL, V. A framework for computational thinking based on a systematic research review. *Baltic Journal of Modern Computing*, Riga, v. 4, n. 3, p. 583-596, 2016.
- KONG, S. C.; ABELSON, H.; LAI, M. Introduction to computational thinking education. In: KONG, S. C.; ABELSON, H. (ed.). *Computational Thinking Education*. [S. l]: Springer Singapore, 2019. p. 1-10.
- LANDIS, J. R.; KOCH, G. G. The measurement of observer agreement for categorical data. *Biometrics*, Bethesda, v. 33, n. 1, p. 159-174, 1977.
- MAIORANA, F.; GIORDANO, D.; MORELLI, R. *Quizly: a live coding assessment platform for app inventor*. In: IEEE BLOCKS AND BEYOND WORKSHOP, 2015, [S. l]. *Proceedings [...]*. [S. l]: IEEE, out. 2015. p. 25-30. DOI: <https://doi.org/10.1109/BLOCKS.2015.7368995>
- MERRICK, K. E. *An empirical evaluation of puzzle-based learning as an interest approach for teaching introductory computer science*. [S. l]: IEEE Transactions on Education, 2010.
- MICHALEWICZ, Z.; MICHALEWICZ, M. *Puzzle-based learning: introduction to critical thinking, Mathematics, and problem solving*. Melbourne: Hybrid Publishers, 2008.
- MORENO-LEÓN, J.; ROBLES, G.; ROMÁN-GONZÁLEZ, M. Dr. Scratch: automatic analysis of scratch projects to assess and foster computational thinking. *RED-Revista de Educación a Distancia*, Murcia, n. 46, p. 1-23, 2015.
- MÜHLING, A.; RUF, A.; HUBWIESER, P. Design and first results of a psychometric test for measuring basic programming abilities. In: THE WORKSHOP IN PRIMARY AND SECONDARY COMPUTING EDUCATION, 2015, London. *Proceedings [...]*. New York: ACM, 2015.
- OTA, G.; MORIMOTO, Y.; KATO, H. Ninja code village for scratch: Function samples/function analyser and automatic assessment of computational thinking concepts. In: IEEE SYMPOSIUM ON VISUAL LANGUAGES AND HUMAN-CENTRIC COMPUTING (VL/HCC), set. 2016, Cambridge, UK. *Proceedings [...]*. Cambridge, UK: IEEE, 2016.

PAPERT, S. *Teaching children thinking*. Cambridge, MA: Massachusetts Institute of Technology, 1971. Disponível em: <https://dspace.mit.edu/handle/1721.1/5835>. Acesso em: 2 ago. 2021.

PASQUALI, L. Validade dos Testes. *Examen: Política, Gestão e Avaliação da Educação*, v. 1, n. 1, p. 36, 1 set. 2017.

RAABE, A.; SANTANA, A. L. M.; ELLERY, N.; GONÇALVES, F. Um instrumento para diagnóstico do pensamento computacional. In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 6., 2017, Recife. *Anais [...]*. Porto Alegre: Sociedade Brasileira de Computação, 2017. DOI: <http://dx.doi.org/10.5753/cbie.wcbie.2017.1172>

RAABE, A.; ZORZO, A.; BLIKSTEIN, P. *Computação na educação básica: fundamentos e experiências*. Porto Alegre: Penso, 2020.

ROBERTS, P.; PRIEST, H.; TRAYNOR, M. Reliability and validity in research. *Nursing Standard*, v. 20, n. 44, p. 41-45, 12 jul. 2006.

ROBLES, G.; HAUCK, J.; ROMÁN-GONZÁLEZ, M.; NOMBELA ALONSO, R.; GRESSE VON WANGENHEIM, C.; MORENO-LEÓN, J. On tools that support the development of computational thinking skills: Some thoughts and future vision. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL THINKING EDUCATION, 2018, Hong Kong. *Proceedings [...]*. Hong Kong: The Education University of Hong Kong, 2018. p. 125-129.

RODRIGUES, M. M. M. Proposta de análise de itens das provas do Saeb sob a perspectiva pedagógica e psicométrica. *Estudos em Avaliação Educacional*, São Paulo, v. 17, n. 34, p. 43-78, maio/ago. 2006. DOI: <https://doi.org/10.18222/ae173420062117>

RODRIGUEZ, C.; REIS, R. C. D.; ISOTANI, S. Recursos e estratégias para desenvolvimento e avaliação do Pensamento Computacional na escola. *Tecnologias, Sociedade e Conhecimento*, Campinas, SP, v. 4, n. 1, p. 59-81, dez. 2017. DOI: <https://doi.org/10.20396/tsc.v4i1.14485>

ROMÁN-GONZÁLEZ, M. Computational thinking test: Design guidelines and content validation. In: ANNUAL INTERNATIONAL CONFERENCE ON EDUCATION AND NEW LEARNING TECHNOLOGIES, 7., 2015, Barcelona. *Proceedings [...]*. Barcelona: IATED, 2015.

ROMÁN-GONZÁLEZ, M. *Códigoalfabetización y pensamiento computacional en educación primaria y secundaria: validación de un instrumento y evaluación de programas*. 2016. Tese (Programa de Doctorado en Educación) – Universidad Nacional de Educación a Distancia, Madrid, 2016.

ROMÁN-GONZÁLEZ, M., MORENO-LEÓN, J., ROBLES, G. Complementary tools for computational thinking assessment. In: INTERNACIONAL CONFERENCE ON COMPUTATIONAL THINKING EDUCATION, 2017, Hong Kong. *Proceedings [...]*. Hong Kong: The Education University of Hong Kong, p. 154-159, 2017.

ROMÁN-GONZÁLEZ, M.; MORENO-LEÓN, J.; ROBLES, G. Combining assessment tools for a comprehensive evaluation of computational thinking interventions. In: KONG, S. C.; Abelson, H. (ed.). *Computational Thinking Education*. [S. l]: Springer Singapore, 2019. p. 79-98.

ROMÁN-GONZÁLEZ, M.; PÉREZ-GONZÁLEZ, J.-C.; JIMÉNEZ-FERNÁNDEZ, C. Which cognitive abilities underlie computational thinking? Criterion validity of the computational thinking test. *Computers in Human Behavior*, Amsterdam, v. 72, p. 678-691, 2017.

ROMÁN-GONZÁLEZ, M.; PÉREZ-GONZÁLEZ, J.-C.; MORENO-LEÓN, J.; ROBLES, G. Can computational talent be detected? Predictive validity of the Computational Thinking Test. *International Journal of Child-Computer Interaction*, v. 18, p. 47-58, nov. 2018. DOI: <https://doi.org/10.1016/j.ijcci.2018.06.004>.

SEITER, L.; FOREMAN, B. Modeling the learning progressions of computational thinking of primary grade students. In: ANNUAL INTERNATIONAL ACM CONFERENCE ON INTERNATIONAL COMPUTING EDUCATION RESEARCH, 9., 2013, San Diego. *Proceedings* [...]. New York: ACM, 2013. p. 59-66.

SMITH, L.; CORDOVA, J. Weighted primary trait analysis for computer program evaluation. *Journal of Computing Sciences in Colleges*, Evansville, v. 20, n. 6, p. 14-19, June 2005.

SRIKANT, S.; AGGARWAL, V. A system to grade computer programming skills using machine learning. In: ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, 20., 2014, New York. *Proceedings* [...]. New York: ACM, 2014. p. 1887-1896. DOI: <https://doi.org/10.1145/2623330.2623377>

WANGENHEIM, C. G. V.; HAUCK, J. C. R.; DEMETRIO, M. F.; PELLE, R.; ALVES, N. D. C.; BARBOSA, H.; AZEVEDO, L. F. CodeMaster – Automatic assessment and grading of app inventor and snap! programs. *Informatics in Education*, Vilnius, v. 17, n. 1, p. 117-150, 2018.

WEINTROP, D.; WILENSKY, U. Using commutative assessments to compare conceptual understanding in blocks-based and text-based programs. In: THE ELEVENTH ANNUAL INTERNATIONAL CONFERENCE, 2015, Omaha. *Proceedings* [...]. New York: ACM, 2015. p. 101-110. DOI: <https://doi.org/10.1145/2787622.2787721>

WING, J. M. Computational thinking. *Communications of the ACM*, New York, v. 49, n. 3, p. 33-35, 2006.

ZUR-BARGURY, I.; PÂRV, B.; LANZBERG, D. A nationwide exam as a tool for improving a new curriculum. In: INNOVATION AND TECHNOLOGY IN COMPUTER SCIENCE EDUCATION CONFERENCE, 18., 2013, Canterbury. *Proceedings* [...]. New York: ACM, 2013. p. 267-272. DOI: <https://doi.org/10.1145/2462476.2462479>

**NOTA:** Os autores participaram de todo o processo de elaboração do artigo, da concepção à finalização. A pesquisa relatada foi fruto da dissertação de mestrado da autora Natália Ellery Ribeiro Couto. O autor Leonardo Felipe de Ávila Calbusch realizou o levantamento de referencial teórico, a elaboração das justificativas para o trabalho e a organização e redação do texto final. A autora Julia Zanetti Rocca colaborou com sua *expertise* na área da psicometria, com sugestões para o conteúdo e a organização do texto. O autor André Luís Alice Raabe orientou o trabalho, promoveu discussões e sugeriu metodologias de análise dos dados. Todos os autores participaram da revisão e colaboraram com trechos do texto do presente artigo.