

## ***CT Puzzle Test: Em direção a uma avaliação interativa do pensamento computacional***

**André Raabe, Cassiano Viana, Leonardo Calbusch**

Mestrado em Computação Aplicada  
Laboratório de Inovação Tecnológica na Educação - LITE  
Universidade do Vale do Itajaí (Univali)  
Itajaí - SC - Brasil

[raabe@univali.br](mailto:raabe@univali.br), [cassiano.viana@univali.br](mailto:cassiano.viana@univali.br),  
[leonardo.calbusch@univali.br](mailto:leonardo.calbusch@univali.br)

**Abstract.** *Assessing Computational Thinking (CT) is essential to understand the benefits it can bring to Basic Education. However, most evaluative methods analyze static codes and not the thinking process that occurred during problem solving. This work seeks to fill this gap by presenting an assessment tool based on puzzles, called CT Puzzle Test. The process of assessing the validity of the instrument has already covered the stages of apparent validity, content validity and convergent validity, where it obtained a moderate correlation ( $p = 0.62$ ) with the Computational Thinking Test instrument. The CT Puzzle Test has evolved to become a platform that allows the creation and application of PC tests based on puzzles.*

**Resumo.** *Avaliar o Pensamento Computacional (PC) é fundamental para que se possa compreender os benefícios de ensiná-lo na Educação Básica. No entanto, a maioria dos métodos avaliativos, analisa códigos estáticos e não o processo de pensamento ocorrido durante a resolução do problema. Esse trabalho, procura preencher essa lacuna ao apresentar um instrumento de avaliação baseada em puzzles, chamado CT Puzzle Test. O processo de aferição da validade do instrumento já contemplou as etapas de validade aparente, validade de conteúdo e validade convergente, onde obteve correlação moderada ( $p=0,62$ ) com o instrumento Computational Thinking Test. O CT Puzzle Test foi evoluído para se tornar uma plataforma que possibilita a criação e aplicação de testes de PC baseados em puzzles.*

### **1. Introdução**

Em 2006, Jeannet Wing introduziu o termo e afirmou que o Pensamento Computacional (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" (WING, 2006). Desde então ainda não se estabeleceu um consenso entre os pesquisadores da área sobre a definição do PC (KALELIOGLU et al., 2016).

Brackmann (2017) apresenta uma pesquisa que aponta a definição de 4 conceitos fundamentais considerados como pilares para o PC: (i) decomposição - quebrar um problema ou sistema complexo em partes; (ii) reconhecimento de padrão - procurar similaridades entre problemas; (iii) abstração - foco nas informações importantes, ignorando detalhes irrelevantes e (iv) algoritmos - desenvolver passos ou regras para solucionar um problema. Brennan e Resnick (2012) avaliam que o PC se manifesta em dimensões que incluem a transferência de conceitos computacionais para outros

contextos, as práticas do processo de aprender e ensinar (pensamento incremental e iterativo, testagem e depuração, reutilização e combinação de conhecimentos) e as perspectivas dos indivíduos sobre o mundo à sua volta e sobre si mesmos (auto-expressão, interatividade, colaboração e interrogação). Blikstein (2008) argumenta que o PC está relacionado a “saber usar o computador como instrumento de aumento do poder cognitivo e operacional humano [...] para aumentar nossa produtividade, inventividade, e criatividade”. Estabelecer congruência entre essas abordagens é fundamental para a compreensão do PC e o planejamento de metodologias avaliativas eficientes. A partir disso, neste trabalho propomos uma 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, uso e avaliação de algoritmos, no reconhecimento de padrões, na decomposição de problemas e na abstração.

Avanços na definição de instrumentos de avaliação psicométrica contribuem para o estabelecimento de correlações entre o PC e outros construtos psicológicos. O resultado dessas correlações evidenciadas e validadas é o que possibilita a definição do PC como um construto psicológico concreto.

A disponibilidade de instrumentos de avaliação validados também permite que o PC seja investigado em contextos educativos variados. Nas etapas escolares da infância, investigar o PC pode auxiliar no desenvolvimento cognitivo, raciocínio lógico e capacidade de resolução de problemas das crianças. Na educação de jovens, pode-se desenvolver sua capacidade de transferir habilidades de PC para diferentes tipos de problemas, contextos e situações da vida real. No meio acadêmico, investigar o PC pode contribuir com a definição de currículos, projetos pedagógicos e políticas educacionais. Ainda é possível investigar o desenvolvimento do PC em indivíduos em contextos clínicos, como por exemplo os relacionados a públicos com necessidades especiais.

Um levantamento de Wangenheim et al. (2018) aponta que a avaliação do PC em estudantes, na maioria dos casos, se dá por meio de tarefas de programação que avaliam assertividade e eficiência de código. Outras iniciativas utilizam instrumentos na forma de questionários de múltipla escolha com uma única resposta correta previamente definida, como o *Commutative Assessment Test* (WEINTROP; WILENSKY, 2015), o *Test for Measuring Basic Programming Abilities* (MÜHLING et al., 2015) e o *Bebras Tasks* (DAGIENE; FUTSCHEK, 2008).

Percebe-se, portanto, uma carência de estratégias de avaliação mais amplas do PC. As abordagens baseadas em questionários e análise de código não capturam informações relevantes sobre como se dá o processo de resolução de problemas pelos estudantes tais como o número de tentativas, o tempo entre tentativas, o número de ações realizadas, número de erros cometidos e outras informações que são particularmente importantes para avaliação do PC tendo em vista que existem práticas que são tipicamente associadas ao seu desenvolvimento.

Acredita-se que abordagem de avaliação do PC por meio da resolução de *puzzles* possibilita considerar ao mesmo tempo os conceitos e as práticas relacionadas ao PC além de permitir a coleta de informações relevantes sobre o processo de resolução dos problemas pelos estudantes. Falkner et al. (2010), menciona que o uso de *puzzles* pode motivar e, ao mesmo tempo, aumentar a consciência matemática e habilidades de resolução de problemas dos alunos. Desta forma, este artigo apresenta o *Computational*

*Thinking Puzzle Test*, referido adiante como *CT Puzzle Test*. O instrumento adota o estilo de resolução de problema com uso de *puzzles* ao invés de questões de múltipla escolha, provendo um ambiente para experimentação e teste de hipóteses. Além de avaliar os resultados, são capturadas as interações do estudante durante o uso do instrumento para mensurar-se o nível de desenvolvimento dos quatro pilares do PC manifestados no processo de resolução dos *puzzles* pelo estudante.

O artigo está organizado como segue: Na seção 2 são analisadas as principais abordagens de avaliação do PC, a seção 3 apresenta o *CT Puzzle Test* e os resultados de etapas de validação já realizadas com o instrumento. A Seção 4 apresenta a evolução de escopo do teste que resulta na construção do *CT Puzzle Platform*. A seção 5 discute a contribuição da abordagem e as direções futuras da pesquisa.

## **2. Avaliação do pensamento computacional**

Muitas das iniciativas de avaliação do PC existentes se baseiam no ensino de programação como ferramenta de apoio ao desenvolvimento dos pilares do PC. Alguns exemplos são o *Dr. Scratch* (MORENO-LEÓN et al., 2015), *Progression of Early Computational Thinking* (PECT) (SEITER; FOREMAN, 2013), *CodeMaster* (WANGENHEIM et al., 2018) e *Quizly* (MAIORANA et al., 2015), entre outros. Esse tipo de avaliação apresenta algumas limitações. Para desenvolver seus projetos, os estudantes precisam de conhecimento prévio sobre o ambiente ou linguagem utilizada. Além disso, esses instrumentos geralmente funcionam no código-fonte dos projetos, normalmente somente após o estudante terminar seu trabalho.

Já as iniciativas como o *Commutative Assessment Test* (WEINTROP; WILENSKY, 2015), o *Test for Measuring Basic Programming Abilities* (MÜHLING et al., 2015) e o *Bebras Tasks* (DAGIENE; FUTSCHEK, 2008), que adotam o formato de teste de múltipla escolha, avaliam somente as respostas finais dos estudantes para questões previamente definidas. Assim, acabam por não mensurar o processo cognitivo executado na resolução de cada questão.

Román-González et al. (2019) fazem uma análise dos instrumentos de avaliação do PC conforme sua abordagem avaliativa, e os classificam em 7 modalidades: diagnóstico, sumativo, formativo-iterativo, mineração de dados, transferência de habilidades, percepções e atitudes do PC e vocabulário de PC. Cada abordagem avalia o 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 insuficiente, e 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.

### **2.2 Trabalhos correlatos (Outros Instrumentos de Avaliação do PC)**

Ao analisar os exemplos de instrumentos de avaliação do PC existentes, percebe-se que cada instrumento acessa o PC sob diferentes perspectivas. Para investigar a combinação de abordagens avaliativas a fim de alcançar a mensuração do PC em sua maior completude, Román-González et al. (2019) pesquisou a validade convergente entre três instrumentos com abordagens diferentes: o *Bebras Tasks* (“transferência de habilidades”), o *CTt* (“diagnóstico”) e o *Dr. Scratch* (“formativo-iterativo”). O *CT Puzzle*

*Test* combina essas três abordagens de avaliação. A seguir, apresentamos as três ferramentas e as relacionamos com o *CT Puzzle Test*.

### 2.2.1 *Bebras Task*

O Concurso Internacional *Bebras* visa promover o interesse e a excelência de estudantes do ensino fundamental e médio em todo o mundo no campo da Ciência da Computação do ponto de vista do PC (CARTELLI et al., 2012; DAGIENE; FUTSCHEK, 2008; DAGIENE; STUPURIENE, 2014). A cada ano, o concurso propõe um conjunto de Tarefas (“*Bebras Tasks*”), independentes de qualquer *software* ou *hardware* específico e que, assim como o *CT Puzzle Test*, podem ser administradas a indivíduos sem qualquer experiência anterior em programação. Sua crescente expansão despertou o interesse de pesquisadores em psicometria, que começaram a investigar suas possíveis virtudes como instrumento de medida do PC (ROMÁN-GONZÁLEZ et al., 2017b).

### 2.2.2 *CTt*

O *Computational Thinking Test – CTt* (ROMÁN-GONZÁLEZ, 2015) é uma ferramenta de avaliação diagnóstica que consiste em um instrumento de múltipla escolha composto por 28 itens, administrados *on-line*. 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.

### 2.2.3 *Dr.Scratch*

O *Dr. Scratch* (MORENO-LEÓN et al., 2015) é um aplicativo *web* que analisa, de maneira automatizada, projetos programados em linguagem *Scratch*. A pontuação que o *Dr. Scratch* atribui a um projeto é baseada no grau de desenvolvimento de sete dimensões do PC: abstração e decomposição de problemas, pensamento lógico, sincronização, paralelismo, noções algorítmicas de controle de fluxo, interatividade do usuário e representação de dados. Essas dimensões são avaliadas estaticamente pela inspeção do código-fonte do projeto analisado, resultando em uma “pontuação de domínio” que varia de 0 a 21 quando todas as sete dimensões são agregadas.

Román-González et al. (2019) argumentam sobre a importância de combinar as abordagens avaliativas desses três instrumentos para uma compreensão mais completa sobre o desenvolvimento do PC nos estudantes. O *CT Puzzle Test* é apresentado como um instrumento que combina essas e outras abordagens, como apontado no Quadro 1:

**Quadro 1: Ferramentas de avaliação do pensamento computacional**

Instrumento	Abordagem	Conhec. prévio
<i>Computational Thinking Test</i> (Román-González, 2015; Román-González et al, 2017)	Diagnóstico (Questionário de Múltipla Escolha/ Perguntas compostas por desafios lógicos)	Não
<i>Dr. Scratch</i> (Moreno-León et al., 2015)	Formativo- iterativo (Feedback para o estudante por avaliação de registro de atividade)	Sim: Scratch
<i>Bebras Tasks</i> (Dagiene; Futschek, 2008)	Transferência de habilidades (Questionário de Múltipla Escolha/ Perguntas compostas por desafios lógicos)	Não

<i>CT Puzzle Test</i>	Diagnóstico; formativo-iterativo; transferência de habilidades; mineração de dados (Jogo composto por puzzles apoiados em pilares do PC, graduação com base no registro de atividades)	Não
-----------------------	--	-----

### 3. *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.

#### 3.1 Mecânicas do *CT Puzzle Test*

O *CT Puzzle Test* é composto por problemas no formato de *puzzles*, planejados para que não seja necessário nenhum conhecimento prévio para resolvê-lo. Os problemas estão organizados em fases as quais apresentam um mesmo tipo de mecânica/*puzzle*, porém com níveis de dificuldade diferentes. O instrumento foi projetado para ser aplicado *on-line* e utiliza as tecnologias *HTML5* e *Javascript*.

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 2 apresenta as fases do teste, com uma descrição das suas mecânicas e das interações avaliadas.

**Quadro 2: 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>

	<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 seu devido lugar, repetindo este 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>Analisar os objetos e organizá-los 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; quantidade de vezes que 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 a quantidade de vezes que um comando direcional deve ser repetido automaticamente.</p> <p><b>Interações avaliadas:</b> além dos itens já avaliados na fase 1, também são contabilizadas a quantidade de vezes que o comando <i>loop</i> foi inserido e a quantidade de comandos direcionais inseridos em cada <i>loop</i>.</p>

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 deste item.

Cada tipo de interação avaliada possui um peso e gera descontos ou acréscimos na pontuação de cada item. Tanto as quantidades absolutas de cada tipo de interação, quanto os cálculos de escore de cada item são armazenados em banco de dados para permitir sua análise pelo aplicador do teste. Para cada fase, são atribuídos pesos aos quatro pilares do PC, e uma pontuação para cada pilar é mensurada. Ao final do teste, o aplicador dispõe de um relatório composto pelos dados brutos sobre as interações do estudante, dos escores de cada item e da pontuação geral de cada pilar do PC. No seu estado atual de desenvolvimento, o *CT Puzzle Test* ainda não apresenta uma interpretação

amigável dos resultados que facilite a análise do aplicador, recurso que será implementado na *CT Puzzle Platform*, detalhada na seção 4. A metodologia de aferição das pontuações e a abordagem de contemplação dos pilares do PC em cada fase são encontradas em Couto (2018), e serão aprofundados por Calbusch (2020, no prelo).

### 3.2 Etapas de validação já realizadas

A primeira versão do teste foi desenvolvida por Gonçalves (2015). Para estimar a validade do instrumento, Gonçalves conduziu aplicações na própria universidade onde o teste foi elaborado. Para a validade aparente e de conteúdo, o pesquisador utilizou amostras compostas por 11 especialistas e por estudantes de ensino médio. Para a validade concorrente, as amostras foram compostas por estudantes de cursos superiores. Para a validade preditiva, estudantes do 9º ano do ensino fundamental foram submetidos ao instrumento. Para estimativa de fidedignidade foi utilizado o método da consistência interna e o coeficiente alfa 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. De acordo com os relatos apresentados na pesquisa de Gonçalves, seus esforços na investigação da validade e fidedignidade do *CT Puzzle Test* indicaram evidências fortes de validade.

Entre 2017 e 2018, o *CT Puzzle Test* foi aprimorado por Couto (2018), em continuidade ao trabalho de Gonçalves. A autora revisou e aprimorou as fórmulas de cálculo de escore de todos os itens do instrumento, e removeu alguns itens visando diminuir o tempo para sua resolução e viabilizar sua aplicação em um período regular de aula. Couto também investigou a validade convergente do instrumento modificado, e adotou como critério o *CTi* (ROMÁN-GONZÁLEZ, 2015) por ser um instrumento independente de conhecimento prévio do sujeito, ser aplicável de forma *on-line* e gerar um escore passível de comparação com os resultados do *CT Puzzle Test*. Além disso, o *CTi* já passou por testes para confirmar sua validade e fidedignidade (ROMÁN-GONZÁLEZ, 2015; ROMÁN-GONZÁLEZ et al., 2017b). Em sua pesquisa, disponível em Couto (2018), a autora relatou que o *CT Puzzle Test* possui consistência interna mensurada por um alfa de Cronbach de 0.67, e possui correlação moderada positiva com o *CTi* ( $p=0.62$ ), contribuindo com o aprimoramento da confiabilidade do *CT Puzzle Test*.

## 4. *CT Puzzle Platform*

O *CT Puzzle Test* é composto por fases de implementação fixa. Por isso, variações na forma de avaliar são possíveis apenas com a alteração no código fonte. Essa solução atende o requisito de um teste estável para avaliação do PC. Mas para situações onde é necessário variar os níveis, adicionar ou remover fases, a solução atual apresenta pouca flexibilidade. Por isso, como extensão do *CT Puzzle Test*, foi criada uma plataforma de avaliação do PC. Através dela, é possível instanciar testes contendo as mesmas fases (ou mecânicas) adotadas.

A configuração do teste se dá por meio de arquivos *JSON* (*JavaScript Object Notation*) específicos para cada fase do jogo. Por exemplo, a fase de programação recebe uma matriz de obstáculos (Figura 2), posição inicial do personagem e ponto de chegada:

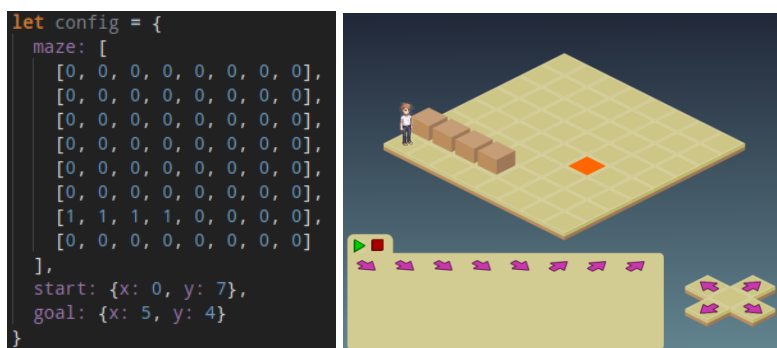


Figura 2: Exemplo de configuração de fase de programação

Para se caracterizar como uma plataforma, o *CT Puzzle Platform* pode ser estendido por meio de *plugins*. A vantagem de tornar a ferramenta extensível é que outras abordagens de *puzzles* podem ser criadas para avaliar os mesmos pilares do pensamento computacional. A estrutura para essa extensão é definida por um tipo padrão *CTPuzzlePhase* que deve ser implementado. Esse tipo recebe uma configuração e tem um objeto para registro dos diversos eventos *EventRegister* durante o uso. Essa classe (Figura 3) é responsável por capturar a interação do usuário e enviá-la para a camada de persistência.

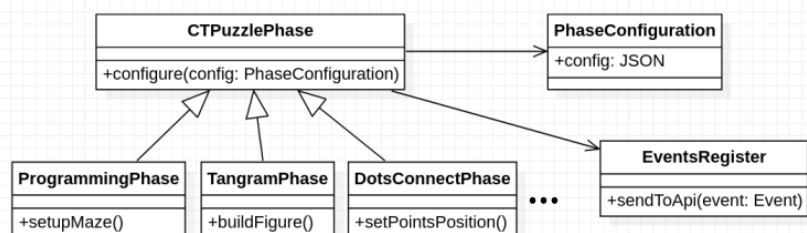


Figura 3: Diagrama de classes do CT Puzzle Platform

Independente do tipo de fase, os eventos capturados durante o uso são armazenados em uma tabela de dados relacionais durante o evento. Isso permite obter os dados mesmo que o participante não finalize o teste. A configuração das fases para aplicação em cada amostra gera dados para posterior consulta. O resumo dos resultados pode ser visualizado no ambiente por meio de um *dashboard*.

O fluxo de uso da plataforma se divide em três etapas: preparação do teste, definição da amostra e análise estatística dos resultados. Na primeira etapa, o pesquisador deve selecionar as fases que farão parte da avaliação. Para isso será indicado qual dimensão do pensamento computacional cada fase analisa. Na definição da amostra, deve selecionar um identificador único e número de participantes. Na etapa de análise, poderá obter os dados de número de erros e acertos, momento de maior número de falhas por fase, desempenho por dimensão do PC, entre outras. Esses dados ajudam não somente a avaliar a turma, mas também verificar alterações necessárias nas fases dos jogos ao identificar puzzles que apresentam baixo potencial de discriminação.

## 5. Discussão

O *CT Puzzle Test* contempla a independência de domínio e conhecimento prévio, podendo ser aplicado nas condições de pré-teste e pós-teste, sem a necessidade de treinamento dos estudantes. O diferencial do *CT Puzzle Test* com relação aos demais instrumentos apresentados reside na avaliação por meio de *puzzles*, que estimula o



raciocínio lógico e o pensamento crítico. O instrumento considera a experimentação e teste de hipóteses por meio da captura das ações do estudante para o cálculo do escore, e mensura o nível de desenvolvimento dos quatro pilares do PC no indivíduo. Por fim, a abordagem avaliativa do *CT Puzzle Test* se enquadra nos quatro níveis de complexidade cognitiva do PC conforme Gows et al. (2013): reconhecer, entender, aplicar e assimilar.

A continuidade do processo de evolução do instrumento prevê uma nova etapa de validação das modificações que o instrumento sofreu em suas últimas etapas de aprimoramento com novas amostras de estudantes. O *CT Puzzle Test* será re-submetido a técnicas de investigação de validade de conteúdo, de critério e de construto. Especialistas julgarão a relevância e a representatividade do conteúdo do teste, e uma nova comparação com o teste de Román-González (*CTt*) será realizada para reafirmar a correlação dos instrumentos e atestar a validade do *CT Puzzle Test* no diagnóstico do PC para estudantes. Já o *CT Puzzle Platform*, após uma etapa de validação interna, será disponibilizado em conjunto com uma documentação para permitir que outros pesquisadores e educadores possam utilizá-lo.

## 6. Referências

- BLIKSTEIN, P. O pensamento computacional e a reinvenção do computador na educação. 2008. Disponível em: [http://www.blikstein.com/paulo/documents/online/ol\\_pensamento\\_computacional.html](http://www.blikstein.com/paulo/documents/online/ol_pensamento_computacional.html). Acesso em: 29 setembro 2020.
- BOCCONI, S. et al. Developing computational thinking in compulsory education-implications for policy and practice. Join Research Center (European Commission). Sevilha. 2016.
- BRACKMANN, Christian Puhmann. Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica. 2017. Tese (Programa de Pós-Graduação em Informática na Educação) – Universidade Federal do Rio Grande do Sul, Porto Alegre, Rio Grande do Sul.
- BRENNAN, K.; RESNICK, M. 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. 2012.
- CALBUSCH, L. *CT Puzzle Test: Instrumento para Avaliação do Pensamento Computacional de Jovens de 14 a 17 Anos*. Dissertação (Mestrado em Computação Aplicada) – Universidade do Vale do Itajaí. No prelo, 2020.
- CARTELLI, A.; DAGIENE, V.; FUTSCHEK, G. Bebras Contest and Digital Competence Assessment. Current Trends and Future Practices for Digital Literacy and Competence. [s.l.]: IGI Global, 2012.
- COUTO, N. E. R. Impacto da implantação de uma disciplina de pensamento computacional em estudantes. Dissertação (Mestrado em Computação Aplicada) – Universidade do Vale do Itajaí. 2018.
- 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
- DAGIENE, V.; STUPURIENE, G. Informatics education based on solving attractive tasks through a contest. In: KEY COMPETENCIES IN INFORMATICS AND ICT, 2014, Potsdam. Preliminary Proceedings. Potsdam: University Of Potsdam, 2014.
- FALKNER, N.; SOORIAMURTHI, R.; MICHALEWICZ, Z. Puzzle-Based Learning for Engineering and Computer Science. *Computer*, v. 43, n. 4, p. 20–28, abr. 2010.

- FEUERSTEIN, R. The Feuerstein Instrumental Enrichment Program. Jerusalem: ICELP Publications, 2006.
- GONÇALVES, F. Um instrumento para diagnóstico do pensamento computacional. 2015. 128 f. Dissertação - Mestrado em Computação Aplicada, Universidade do Vale do Itajaí, 2015.
- GOUWS, L. A.; BRADSHAW, K.; WENTWORTH, P. Computational thinking in educational activities. Proceedings of the 18th ACM conference on Innovation and technology in computer science education - ITiCSE '13. Anais... In: THE 18TH ACM CONFERENCE. ACM Press, 2013.
- GROVER, S. Systems of assessments for deeper learning of computational thinking in k-12. In: Annual meeting of the American Educational Research Association, Chicago: 2015.
- HOGAN, T. Introdução À Prática de Testes Psicológicos. Rio de Janeiro: Ltc, 2006.
- KALELIOGLU, F.; GULBAHAR, Y.; KUKUL, V. A Framework for Computational Thinking Based on a Systematic Research Review. Baltic Journal Of Modern Computing. Letônia, 2016.
- MAIORANA, F.; GIORDANO, D.; MORELLI, R. Quizly: A live coding assessment platform for App Inventor. In: 2015 IEEE BLOCKS AND BEYOND WORKSHOP (BLOCKS AND BEYOND), out. 2015. Anais [...]. [s.l.]: IEEE, out. 2015.
- MORENO-LEÓN, J.; ROBLES, G.; ROMÁN-GONZÁLEZ, M. Dr. Scratch: Automatic Analysis of Scratch Projects to Assess and Foster Computational Thinking. Revista de Educación A Distancia, Espanha, v. 10, n. 46, p.1-23, set. 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. Anais [...]. [s.l.]: ACM Press, 2015.
- PAPERT, S. Teaching children thinking. MIT Artificial Intelligence Laboratory. Memo no. 2247, Logo Memo no. 2. Cambridge. 1971.
- ROMÁN-GONZÁLEZ, M. Computational thinking test: Design guidelines and content validation. In: Proceedings of the 7th Annual International Conference on Education and New Learning Technologies (EDULEARN), 2015, Barcelona. Proceedings.. Barcelona: IATED, 2015.
- ROMÁN-GONZÁLEZ, M.; MORENO-LEÓN, J.; ROBLES, G. Complementary Tools for Computational Thinking Assessment. Proceedings of International Conference on Computational Thinking Education 2017. Hong Kong: The Education University of Hong Kong, jun. 2017.
- 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, v. 72. 2017b.
- ROMÁN-GONZÁLEZ, M.; MORENO-LEÓN, J.; ROBLES, G. Combining Assessment Tools for a Comprehensive Evaluation of Computational Thinking Interventions. Computational Thinking Education. [s.l.]: Springer Singapore, 2019.
- SEITER, L.; FOREMAN, B. Modeling the learning progressions of computational thinking of primary grade students. In: THE NINTH ANNUAL INTERNATIONAL ACM CONFERENCE, 2013, San Diego. Anais [...]. San Diego: ACM Press, 2013.
- WANGENHEIM, C. et al. CodeMaster - Automatic Assessment and Grading of App Inventor and Snap! Programs. Informatics in Education, v. 17, n. 1, 14 abr. 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. Anais [...]. [s.l.]: ACM Press, 2015.
- WING, J. M. Computational thinking. Communications of the ACM, v. 49, n. 3, p. 33, 1 mar. 2006.