

# NÍVEIS DE MATURIDADE E CAPACIDADE DO PENSAMENTO COMPUTACIONAL AO TÉRMINO DO ENSINO FUNDAMENTAL

Eduardo Cardoso de Souza<sup>1</sup>  
Wilson Massashiro Yonezawa<sup>2</sup>

## RESUMO

O conceito de Pensamento Computacional tem sido polêmico por não existir uma definição unificada, e estudos recentes mostram que esse tipo de pensamento não está bem estabelecido, sobretudo no currículo da Educação Básica. Este trabalho, contudo, teve como objetivo construir uma métrica que representa a capacidade de Pensamento Computacional dos estudantes ao término do Ensino Fundamental. Para isso, foi utilizado um instrumento espanhol de diagnóstico de Pensamento Computacional que, por meio de uma abordagem quantitativa de característica exploratória-descritiva, atendendo ao rigor das técnicas estatísticas de análise multivariada de dados, possibilitou construir e validar um modelo integrado por três dimensões estruturantes: algoritmo, reconhecimento de padrão e decomposição. Buscou-se responder à pergunta-chave: Quais são as habilidades que os estudantes, ao término do Ensino Fundamental de uma cidade do oeste paulista, mobilizam e articulam diante da resolução de problemas por meio de tarefas desplugadas de fundamentos da ciência da computação? Um total de 462 estudantes respondeu voluntariamente ao teste de Pensamento Computacional no período de outubro a dezembro de 2019, o que corresponde a 74,6% dos concluintes do Ensino Fundamental no município de Santa Cruz do Rio Pardo.

**Palavras-chave:** Pensamento computacional. Níveis de maturidade e capacidade. Análise multivariada de dados. Modelagem de equações estruturais.

## MATURITY LEVELS AND CAPACITY OF COMPUTATIONAL THINKING AT THE END OF ELEMENTARY SCHOOL

## ABSTRACT

The concept of Computational Thinking has been controversial because there is no unified definition and recent studies show that this type of thinking is not well established, especially in the basic education curriculum. However, this work aimed to build a metric that represents the computational thinking capacity of students at the end of elementary school. For this, a Spanish Computational Thinking diagnostic instrument was used, which, through an exploratory-descriptive quantitative approach, meeting the rigor of the statistical techniques of multivariate data analysis, made it possible to build and validate a model integrated by three structuring dimensions: algorithm, pattern recognition and decomposition. We sought to answer the key question of: what are the skills that students at the end of elementary school in a city in the west of São Paulo mobilize and articulate in the face of problem solving, through tasks unplugged from the fundamentals of information science? A total of 462 students who responded to the Computational Thinking test in the period from October to December 2019, which corresponds to 74.6% of the graduates of elementary school in the municipality of Santa Cruz do Rio Pardo.

**Keywords:** Computational thinking. Maturity and capacity levels. Multivariate data analysis. Structural equation modeling.

RECEBIDO EM: 29/10/2020

ACEITO EM: 26/1/2021

<sup>1</sup> Autor correspondente. Universidade Estadual Paulista Júlio de Mesquita Filho – Unesp. Av. Eng. Luiz Edmundo Carrijo Coube, 14-01 – Vargem Limpa. CEP 17033-360 – Bauru/SP, Brasil. <http://lattes.cnpq.br/5330125336931569>. <https://orcid.org/0000-0002-8707-615X>. [eduardo.cardoso@unesp.br](mailto:eduardo.cardoso@unesp.br)

<sup>2</sup> Universidade Estadual Paulista Júlio de Mesquita Filho, Faculdade de Ciências de Bauru, Departamento de Computação. Bauru/SP, Brasil. <http://lattes.cnpq.br/3806652471471086>. <https://orcid.org/0000-0003-0607-9787>.

Influenciado pelas ideias de Wing (2006), o Pensamento Computacional começou a ser amplamente discutido no mundo. Com diversas iniciativas no Brasil, vem se consolidando como área de interesse em diversos grupos de pesquisa, sob diferentes óticas, sobretudo na Educação Básica Brasileira (4 a 17 anos). Zanetti, Borges e Ricarte (2016) realizaram recentemente uma Revisão Sistemática da Literatura (RSL) para identificar quais são as práticas pedagógicas mais realizadas, em quais níveis de escolaridade as pesquisas estão sendo feitas e quais são as habilidades desenvolvidas por meio dessas iniciativas. Os autores categorizaram essas práticas, das quais foi possível destacar o uso de programação visual, computação desplugada e jogos digitais como as práticas mais recorrentes no processo de ensino e aprendizagem do Pensamento Computacional (PC), preponderantemente em estudos realizados no âmbito do Ensino Fundamental, com incidência de habilidades consideradas estruturantes do PC, como o pensamento algorítmico e a decomposição de problemas.

Os currículos possuem diversas missões nos diferentes níveis de ensino, em conformidade com a característica de cada um, e apresentam a concretização dos fins sociais e culturais a que se destina a educação escolarizada (SACRISTÁN, 2000). Nesse cenário, o uso intenso das tecnologias tem influenciado os modos de representar o pensamento, as relações sociais, as práticas de trabalho e a produção da ciência. Assim, as práticas pedagógicas são modificadas para além da sala de aula por meio de ambientes educacionais emergentes, ambientes estes que integram as Tecnologias da Informação e Comunicação (TICs) e os múltiplos letramentos e suas respectivas linguagens e pensamentos. Com isso, emergem novos contextos de aprendizagem em diferentes espaços legítimos de aprendizagem. O docente desse século tem de estar preparado para identificar as potencialidades de diferentes espaços. O professor é peça-chave nesse processo, que assim exigirá professores mais bem preparados.

Há um desafio pedagógico que consiste na mudança de um currículo que foi construído para a era do papel e do lápis, para um currículo voltado aos nativos digitais, para que abandonem o perfil de consumidores de tecnologia e passem a entender como são pensadas e construídas essas ferramentas, ampliando o conceito de letramento para o digital (multiletramentos), a saber, a cultura digital e o Pensamento Computacional.

Dessa forma, o ensino do Pensamento Computacional necessita respaldar-se em modelos em que a teoria de ensino se apresenta adequada (ou não) para o tipo de aprendizagem que se deseja gerar. Assim, reconhece-se o desafio de constituir um currículo progressivo de aprendizagem para o ensino do Pensamento Computacional, verificando-se quais habilidades devem ser avaliadas bem como aspectos relacionados à formação dos professores para que promovam, efetivamente, o ensino do Pensamento Computacional (GROVER; PEA, 2013).

De acordo com Kong (2016), para que haja progresso sistematizado dentro do currículo escolar, será necessário dividir o Pensamento Computacional em um conjunto de conhecimentos, habilidades e perspectivas bem-definidas.

O Pensamento Computacional praticamente não tem sido objeto de ensino explícito na Educação Básica, por isso a necessidade de um currículo que garanta progresso sistematizado por meio de modelos teóricos definidos (KONG, 2016).

Partindo do pressuposto que o Pensamento Computacional é passível de ensino, ele deve ser avaliado. As formas de avaliação têm se apresentado bastante diversificadas, seguindo distintos procedimentos metodológicos na tentativa de identificar se os estudantes apropriaram-se dessas habilidades.

Diante do contexto apresentado, emergiu o problema de pesquisa deste estudo: quais são as habilidades mobilizadas e articuladas pelos estudantes ao término do ensino fundamental em tarefas desplugadas que contemplam fundamentos da ciência da computação?

O objetivo deste trabalho centra-se na tentativa de construção de uma métrica para Pensamento Computacional (PC) que integre aspectos: (a) da capacidade de construir uma sequência finita de instruções para resolver um problema; (b) da capacidade de reconhecer padrões; e (c) da capacidade de dividir um problema complexo em partes menores e funcionais.

Dessa forma, este estudo tem como objetivo geral incluir aspectos relevantes a uma medida, considerando as dimensões da capacidade algorítmica, recursiva (reconhecimento de padrão) e decompositiva, enquanto elementos estruturantes que compõem uma escala/índice de proficiência que servirá de apoio a diferentes atores da escola, sobretudo aos gestores pedagógicos, interessados em avaliar a capacidade do Pensamento Computacional dos estudantes ao término do Ensino Fundamental, com a finalidade de construir diretrizes e elaborar políticas públicas educacionais objetivando a construção de um currículo que faça mais sentido aos estudantes nativos digitais, alinhado a quinta competência da BNCC.

Este estudo possui, ainda, o seguinte objetivo específico: validar um modelo para aferir a capacidade do Pensamento Computacional dos estudantes ao término do Ensino Fundamental. Como ineditismo, apresenta uma abreviação do instrumento espanhol no contexto brasileiro, analisado a partir de três eixos estruturantes do Pensamento Computacional: a capacidade algorítmica, recursiva e decompositiva.

Como não há uma definição unificada para o Pensamento Computacional e sua operacionalização na Educação Básica, o próximo tópico aborda algumas das propostas de Barr e Stephenson (2011) e Kong (2016).

## **CARACTERIZANDO O PENSAMENTO COMPUTACIONAL**

A ideia de Pensamento Computacional foi popularizada por Wing (2006), apresentada inicialmente como um conjunto de habilidades e atitudes que todos podem aprender e usar. A ênfase está na resolução de problemas, explorando os conceitos fundamentais da ciência da computação. Em suma, Wing equaciona o Pensamento Computacional com o pensar enquanto cientista da computação.

Com isso, Wing (2006) abriu um guarda-chuva, e, sob sua sombra, fundamentou uma sucessão de pesquisas nessa mesma narrativa, e desde então tem sido influente nesse contexto. A precursora posteriormente refinou sua compreensão inicial sobre o Pensamento Computacional, apresentando como núcleo central a solução de problemas (WING, 2008, 2010, 2014, 2017).

Para Wing (2014), o Pensamento Computacional é o pensamento envolvido no processo de formulação de um problema e na solução do mesmo, de forma que um humano ou máquina pode realiza-lo, defendendo que as pessoas podem aprender Pensamento Computacional mesmo sem uma máquina. Assim, de acordo com a autora, o Pensamento Computacional descreve a atividade mental na formulação de um problema para admitir uma solução computacional.

A construção intelectual é permeada por vários tipos de pensamento, estudados por diferentes áreas. Diante da diversidade de pensamentos preconizados na literatura, será necessário outro tipo de pensamento? O Pensamento Computacional é de fato distinto e relevante no contexto atual? De antemão, pode-se inferir que os demais tipos de pensamentos foram propostos na era do lápis e do papel. A relevância social do Pensamento Computacional no contexto digital é inegável. A cultura digital tem contribuído para a criação de novas formas de interagir, de produzir, de aprender, de ser e de pensar (ALMEIDA *et al.*, 2017).

Diante disso, vários estudos defendem a integração do Pensamento Computacional no currículo da Educação Básica, como encontrado nos trabalhos de Barr e Stephenson (2011), Brennan e Resnick (2012), Grover e Pea (2013), Lye e Koh (2014), Weintrop *et al.* (2016), Kong (2016) e Shute, Sun e Asbell-Clarke (2017).

A integração do Pensamento Computacional no currículo da Educação Básica pode acontecer em componentes específicos para ensinar fundamentos da ciência da computação (KONG, 2016) ou por meio do ensino de habilidades do Pensamento Computacional em diversas áreas do conhecimento, mediante atividades de computação desplugadas<sup>3</sup> (BARR; STEPHENSON, 2011).

A abordagem proposta por Barr e Stephenson (2011) compreende o Pensamento Computacional como uma competência permeada em todo o currículo (transcurricular). Para Kong (2016), o Pensamento Computacional está fortemente associado à solução de problemas representados por artefatos computacionais. Nesse contexto, o ensino do PC está altamente relacionado com sua contribuição para a modelagem, programação e a robótica, alinhado com as ideias de Brennan e Resnick (2012) de promover um currículo da Educação Básica centrado na programação de computadores.

## **PENSAMENTO COMPUTACIONAL E DOCUMENTO NORTEADOR NO BRASIL**

No Brasil, um recente movimento da Sociedade Brasileira de Computação (SBC) enfatizou a importância de pensar numa Base Nacional Comum Curricular (BNCC) que introduzisse o Pensamento Computacional e a compreensão do mundo digital para a construção de uma nação em condições de igualdade com o cenário internacional. “A

<sup>3</sup> Computação desplugada tem o objetivo de ensinar os fundamentos da Ciência da Computação sem a necessidade de computadores. Uma grande vantagem dessa abordagem reside na sua independência de recursos de *hardware* ou *software*. Dessa forma, as atividades desplugadas são passíveis de aplicação em localidades remotas com acesso precário de infraestrutura, e podem até ser ministradas por não especialistas em computação (BELL; WITTEN; FELLOWS, 2015).

*SBC entende que ensinar Computação não apenas forma melhores cidadãos, mas pode ser o catalisador de mudanças substanciais na nação por ter, no futuro, pessoas melhor capacitadas e habilitadas para a geração de riqueza para o Brasil” (SBC, 2017).*

A BNCC foi homologada pelo Conselho Nacional de Educação no final de 2017 por meio da Resolução CNE/CP nº 2, de 22 de dezembro de 2017, e em seu texto traz quatro referências ao termo “Pensamento Computacional”. Nota-se, no documento, um tímido enfoque à temática, em que todas as referências são atreladas ao Ensino de Matemática na unidade temática da Álgebra.

A BNCC apresenta algumas habilidades do Pensamento Computacional, enfatizando o reconhecimento de padrões que têm relevância na aprendizagem da álgebra, e cita a importância de habilidades relacionadas ao pensamento algoritmo e à decomposição de problemas.

Apesar do tímido enfoque atribuído ao Pensamento Computacional no texto, a BNCC apresenta dez competências<sup>4</sup> gerais, e a quinta trata especificamente da tecnologia, intitulada “Cultura Digital”, que está relacionada à capacidade de os estudantes se apropriarem do conhecimento tecnológico não apenas como consumidores de tecnologia, mas compreendendo como essas tecnologias são programadas e como fazer o melhor uso delas, de forma crítica e reflexiva.

Dessa maneira, é importante que os estudantes saibam resolver os problemas da vida cotidiana, mobilizando também as competências do Pensamento Computacional, que não devem ser desenvolvidas de forma isolada, como uma disciplina complementar, mas de maneira pulverizada sobre o currículo, em que conhecimentos, habilidades e atitudes se “plugam” aos componentes curriculares.

## HABILIDADES ESTRUTURANTES DO PENSAMENTO COMPUTACIONAL

Instituições como a International Society of Technology in Education (Iste) e a Computer Science Teachers Association (CSTA) (2011), o Google (2018) e a British Broadcasting Corporation (BBC) (2018), entendem o Pensamento Computacional ancorado em quatro dimensões estruturantes: Algoritmo, Reconhecimento de Padrão, Decomposição e Abstração. Considerado um domínio amplo, o Pensamento Computacional não se restringe a essas quatro dimensões para a solução de problemas, mas se sustenta por meio delas. Dessa forma, balizar essas dimensões é importante, pois, ao representar e organizar uma solução para os problemas da vida cotidiana, mobilizando esses elementos, tornará a solução passível de produzir artefatos de *softwares*.

Na revisão da literatura diversos autores destacam a construção de algoritmos como uma habilidade fundamental do Pensamento Computacional, que pode ser evidenciada nos estudos de Barr e Stephenson (2011), Berland e Lee (2012), Gouws, Bradshaw e Wentworth (2013), Denner *et al.* (2014), Lee *et al.* (2014), Duncan e Bell (2015) e Mueller *et al.* (2017).

<sup>4</sup> Na BNCC (BRASIL, 2017, p. 8) entende-se como competência a “mobilização de conhecimentos (conceitos e procedimentos), habilidades (práticas, cognitivas e socioemocionais), atitudes e valores para resolver demandas complexas da vida cotidiana, do pleno exercício da cidadania e do mundo do trabalho”.

Uma solução genérica para uma classe de problemas, ao contrário de uma solução específica para um único problema, é um algoritmo (DOWNEY; ELKNER; MEYERS, 2002). De acordo com o Google (2018), os algoritmos são, geralmente, representações abstratas do problema, por isso um algoritmo está escrito por variáveis e não com valores específicos.

Na escola, o processo de projetar algoritmos é interessante, intelectualmente desafiante e imprescindível para o desenvolvimento do Pensamento Computacional. Para Denning (2009), o pano de fundo do Pensamento Computacional é o pensamento algorítmico.

Não menos importante, a capacidade de reconhecer um padrão é uma habilidade que possibilita encontrar semelhanças e diferenças, que será importante para categorizar o problema, ou seja, fragmentá-lo. Para Riley e Hunt (2014), um padrão indica se uma sequência específica faz ou não parte de uma categoria ou classe definida pelo padrão.

Os padrões estão em toda parte; basta olhar ao redor para reconhecer padrões geométricos, numéricos ou textuais. Da revisão da literatura foi possível levantar vários trabalhos que tratam o Reconhecimento de Padrão (*Pattern Recognition*) como habilidade estruturante do Pensamento Computacional, como nos estudos de Fraillon *et al.* (2019).

Lee *et al.* (2014) definem o reconhecimento de padrões como a capacidade de encontrar similaridades e regularidades para solucionar os problemas de forma eficaz. De acordo com Google (2018), a capacidade de reconhecer um padrão possibilita fazer encaminhamentos que podem direcionar ao núcleo de um problema, sendo considerada uma das habilidades centrais para o desenho algorítmico e a solução de problemas.

Na vida cotidiana, muitas vezes é necessário repetir uma sequência de ações para se alcançar um resultado melhor. Assim como muitas atividades da vida real exigem repetição, os computadores, muitas vezes, também devem executar uma sequência de ações repetidamente para solucionar um problema de forma eficiente. Dessa forma, um dos pontos fortes da capacidade computacional é a de repetir ações com rapidez, sem levar à exaustão, o que, provavelmente, aconteceria na atividade humana.

Diante disso, reconhecer padrões mostra-se uma das capacidades importantes do Pensamento Computacional. O reconhecimento de padrões na solução de problemas é fundamental para produzir as soluções adequadas a tais problemas, o que possibilitará decompor o problema e, ao mesmo tempo, a formação de uma visão sistêmica, em que as partes são interdependentes, produzindo soluções funcionais.

A capacidade de construir algoritmos e reconhecer padrões é um pilar importante do Pensamento Computacional. Não distante, a capacidade de decompor problemas integra as habilidades estruturantes do Pensamento Computacional. Decompor envolve dividir um problema em tarefas menores e mais gerenciáveis. Para muitos autores, a capacidade de decompor é considerada uma habilidade central do Pensamento Computacional, apoiando na solução de problemas complexos, conforme evidenciado nos estudos de Barr e Stephenson (2011), Seiter e Foreman (2013), Grover e Pea (2013), Grover (2015), Mueller *et al.* (2017) e Shute, Sun e Asbell-Clarke (2017).

Dessa forma, decompor um problema também exige analisá-lo como um todo. Essa capacidade requer a construção de partes gerenciáveis, e verificar como elas estão relacionadas entre si e com o problema todo. Nessa perspectiva, Weintrop *et al.* (2015) categorizaram o Pensamento Sistêmico como sendo uma prática importante que favorece a compreensão dos relacionamentos dentro de um sistema, em que os estudantes são capazes de identificar os elementos constituintes do mesmo, articular seus comportamentos e explicar como as interações entre elementos produzem os comportamentos característicos de um sistema completo e funcional.

Segundo Davis (2011, p. 118-119), a maior vantagem de dividir um problema complexo em partes menores tem relação com a quantidade de coisas que uma pessoa pode pensar ao mesmo tempo. Os problemas do mundo real e da vida cotidiana são complexos e não são modelados por uma única função. Assim, os programas são separados por módulos, chamados de funções.

Em computação, a decomposição é também utilizada na reutilização de funções, o que permite programas em módulos de código fonte separados. Assim, um programador pode dividir um programa único em arquivos fonte separados (módulos). Esses módulos são compilados separadamente e depois combinados durante o processo de montagem para gerar um programa único. Esse processo de combinar separadamente módulos compilados dentro de um programa único chama-se vinculação, ou, no inglês, *linking* (DAVIS, 2011).

Assim, há vários motivos para dividir um problema complexo em funções mais simples. Na ciência da computação, a principal delas está relacionada à ideia de reutilização do código, que poderia ser utilizado em vários programas, bem como por diferentes áreas dentro do mesmo programa.

## O INSTRUMENTO

O Pensamento Computacional ainda é um construto nebuloso. Sua avaliação permanece como uma questão espinhosa e não resolvida, por ser considerado um termo abrangente que se refere a um amplo conjunto de habilidades para resolver problemas, que deve ser apropriado pelas novas gerações com o objetivo de prosperar em um mundo mediado por algoritmos. Assim, o uso do termo PC evoluiu. Mesmo sem chegar a um consenso sobre sua definição, diversas ferramentas de avaliação foram desenvolvidas a partir de diferentes abordagens e definições (ROMÁN-GONZÁLEZ; MORENO-LEÓN; ROBLES, 2019).

Avaliar é uma atividade relevante e urgente nesse contexto, porque sem ferramentas de avaliação confiáveis e válidas o Pensamento Computacional poderá ter dificuldades para se consolidar no sistema educacional. Dessa forma, construir uma métrica para o Pensamento Computacional implica operacionalizar esse construto por meio de um instrumento avaliativo que deve ser validado.

Nessa perspectiva, torna-se relevante e urgente validar instrumentos para medir a capacidade do Pensamento Computacional. De acordo com Werner *et al.* (2012), umas das dificuldades para consolidação do Pensamento Computacional no currículo escolar

está na ausência de modelos e instrumentos de avaliação. Dessa forma, os avanços nas pesquisas em perspectivas avaliativas podem contribuir decisivamente para consolidar o conceito enquanto um construto robusto para ser estudado e aprimorado.

O teste de Román-González (2016) é uma ferramenta de avaliação de diagnóstico que consiste em um instrumento de múltipla escolha composto por 28 itens, apresentados em uma interface de labirinto ou figura. O teste demonstrou ser confiável e válido para avaliar o PC em estudantes entre 10 e 16 anos no contexto espanhol, em que cada item engloba um ou mais conceitos computacionais, que aparecem em crescente dificuldade e progressivamente ao longo do teste: sequências, condicionais, *loops* e funções (ROMÁN-GONZÁLEZ, 2016).

## UM METAMODELO DE MATURIDADE

Na década de 80 do século 20, o Departamento de Defesa dos Estados Unidos (DoD), em parceria com a *Carnegie-Mellon University*, financiou o *Software Engineering Institute* para pesquisar e desenvolver o *Capability Maturity Model* (CMM), que tem como objetivo avaliar e, conseqüentemente, melhorar o processo de desenvolvimento de *software* baseado em boas práticas bem definidas. Dessa forma, o CMM indica como deve ser produzido um *software*, com foco no processo, para se construir um bom artefato de *software*, alcançando o nível mais alto de qualidade. Há vários modelos CMM, por exemplo, para a engenharia de *software*, desenvolvimento de *software*, aquisição de *software*, etc. Como evolução desse modelo, o *Capability Maturity Model Integration* (CMMI) fornece um conjunto abrangente de diretrizes para o desenvolvimento de produtos e serviços, integrando todos esses modelos (SEI, 2010).

Esse modelo tem foco na melhoria contínua do processo e auxilia as organizações a potencializar seus esforços de melhoria. O CMM propõe cinco diferentes níveis de maturidade, e, para atingir um nível de maturidade, implica necessariamente aumento da capacidade de processo de uma organização (SEI, 2010).

O CMM identifica as características de um processo de *software* eficaz, e o nível de maturidade aborda todas as questões que são essenciais para um projeto de sucesso, incluindo pessoas, tecnologia e processo. Assim, cada nível de maturidade corresponde a um conjunto de práticas de desenvolvimento de *software* e de gestão específicas, denominadas áreas-chave do processo, no inglês *Key Process Areas* (KPAs), que são fundamentais para que a organização possa atingir o nível de maturidade desejado (SEI, 2010).

Dessa forma, a maturidade é avaliada a partir da capacidade que uma empresa tem de realizar um conjunto de atividades relacionadas com as KPAs. Cada KPA é definida por um conjunto de objetivos e práticas. A análise é realizada sobre cada KPA, verificando se os objetivos e práticas são atingidos ou realizados.

A partir da introdução do CMM na década de 90, diferentes tipos de modelos de maturidade derivativos foram constituídos e validados, incluindo modelos de maturidade voltados para a qualidade do ensino, currículos ou instituições, como encontrados nos trabalhos de Ling *et al.* (2012) e Solar, Sabattin e Parada (2013). Neste trabalho será constituído um modelo intitulado “Modelo de Maturidade do Pensamento Computacio-



nal ao término do Ensino Fundamental”, no contexto de uma cidade do oeste paulista, com base nos conceitos e habilidades mobilizadas, integradas e articuladas pelos estudantes, a partir dos resultados do teste de diagnóstico do Pensamento Computacional de Román-González (2016).

## **METODOLOGIA PARA CONSTRUÇÃO DA ESCALA DE CAPACIDADE DE PENSAMENTO COMPUTACIONAL**

Após classificados os itens do teste de PC em cada dimensão por meio de consulta a especialistas da Faculdade de Tecnologia de Ourinhos (SP), os dados foram tratados, transformadas as variáveis em dicotômicas-binárias e aplicados os testes de pressupostos para a Análise Fatorial. Como primeiro pressuposto, foi utilizada a correlação tetracórica que, de acordo com Little (2013, p. 13), é usada para estimar as correlações entre dados binários.

Para Hair Jr. *et al.* (2009), a Análise Fatorial é um termo global relacionado a uma classe de métodos estatísticos multivariados, que, de forma geral, possibilita analisar as correlações (inter-relações) entre as variáveis, definindo um conjunto de construtos (variáveis latentes) chamados fatores.

A Análise Fatorial, enquanto técnica de interdependência, tem se constituído em uma abordagem estatística amplamente utilizada no desenvolvimento, avaliação e refinamento de instrumentos de pesquisas em diversas áreas do conhecimento, incluindo a área de ciências sociais aplicada, conforme encontrado em trabalhos recentes de Araujo (2013), Santos (2014), Siqueira (2014) e Albino (2015).

Hair Jr. *et al.* (2009) sugerem que a Análise Fatorial Exploratória (AFE) preceda a Análise Fatorial Confirmatória (AFC). De acordo com Paranhos *et al.* (2014, p. 17), a AFC é utilizada para testar hipóteses em que o pesquisador, guiado por alguma teoria, investiga em que medida determinadas variáveis são representativas de um conceito/dimensão.

Para o desenho do modelo que comporá a escala de capacidade do PC, foi utilizada a técnica denominada Modelagem de Equações Estruturais (MEE). Conforme Hair Jr. *et al.* (2009), a MEE é uma técnica de Análise Multivariada que integra aspectos da regressão linear múltipla e da análise de fatores comuns, configurando-se como método confirmatório, indicado para testar a validade de um modelo.

Justifica-se o emprego da MEE neste estudo, uma vez que, a partir de indicadores mensuráveis, foi possível delinear os construtos (variáveis latentes) que não poderiam ser medidos diretamente (HAIR *et al.*, 2009). Por meio da MEE, foi possível obter os escores fatorais para as variáveis latentes e os coeficientes de regressão. Dessa forma, ao se testar o modelo por intermédio do *software SmartPLS*, o mesmo automaticamente “plugou” os valores dos coeficientes de regressão às respostas de cada um dos itens, gerando uma métrica padronizada, que permitiu verificar cada participante individualmente. Dessa forma, foi possível obter diretamente o valor do indicador relativo a uma medida que representa a capacidade do PC e suas dimensões, constituído considerando todas as variáveis e casos utilizados. Esta pesquisa é baseado no diagrama de decisão proposto por Hair Jr. *et al.* (2009). O Quadro 1 mostra o percurso metodológico deste estudo para desenvolvimento desse produto.

Quadro 1 – Produto da pesquisa

<b>Produto – Desenvolvimento métrico</b>	<p>Neste produto, os resultados do instrumento proposto para Pensamento Computacional serão analisados. O instrumento é formado por 28 itens, com respostas zero ou 1 e sem dados faltantes ou incorretos. As seguintes análises foram realizadas:</p> <ol style="list-style-type: none"> <li>1. Análise de consistência interna dos dados mediante estimadores tradicionais e robustos.</li> <li>2. Análise de adequação fatorial confirmatória por meio de modelos de equações estruturais, utilizando estimadores baseados em <i>Partial Least Squares</i> (PLS).</li> <li>3. Ajuste/diagnóstico do modelo.</li> <li>4. Desenvolvimento de escores fatoriais para cada um dos participantes. Esses escores serão associados aos resultados dimensionais encontrados na análise confirmatória e servirão para posicionar o participante em relação ao seu grupo.</li> <li>5. Elaboração de uma escala contendo os níveis de maturidade e capacidade do Pensamento Computacional.</li> </ol>
--	---

Fonte: Elaborado pelo autor (2020).

## CONSTRUÇÃO DOS NÍVEIS DE MATURIDADE E CAPACIDADE DO PENSAMENTO COMPUTACIONAL

Como primeiro estágio para a construção do índice, as variáveis foram eleitas para cada dimensão *a priori* com base no arcabouço teórico, com o objetivo inicial de reunir itens que medem habilidades inerentes de cada dimensão. Para isso, cada item do teste foi validado por três pesquisadores com experiência em desenvolvimento de *software*. Nessa fase não se teve como objetivo procurar subdimensões, uma vez que essas emergirão da Análise fatorial *a posteriori*. Como primeiro estágio do desenvolvimento do índice, foi realizado um estudo exploratório em cada uma das dimensões propostas no modelo teórico.

Foram destacados do teste de PC os itens que visam a descrever a dimensão proposta pelo modelo teórico deste trabalho denominada “Reconhecimento de Padrão”. A Tabela 1 apresenta os resultados da Análise Fatorial (AF) para a referida dimensão.

Tabela 1 – Resultado da AF para a Dimensão Reconhecimento de Padrão

Construtos				
	Fator 1	Fator 2	Fator 3	Comunalidade
pc12	0,820	0,106	0,021	0,684
pc8	0,786	0,035	-0,022	0,620
pc22	0,779	-0,009	0,175	0,637
pc6	0,072	0,839	-0,022	0,710
pc9	0,026	0,767	0,235	0,644
pc7	0,011	0,048	0,793	0,632
pc11	0,113	0,138	0,780	0,640

Método de Extração: Análise de Componentes Principais.  
 Rotação: Varimax  
 Variância total explicada: 65,23%  
 Total de observações: 462

Fonte: Elaborada pelo autor.

Diante disso, os três fatores retidos representam 65,23% da variância, selecionado, assim, os fatores com autovalores maiores que 1, o que indubitavelmente mais bem explica o modelo. Dessa forma, a análise fatorial possibilitou uma solução fatorial acei-

tável para a dimensão “Reconhecimento de Padrão”, em que o Fator 1 é composto pelas variáveis pc12, pc8 e pc22, Fator 2 composto pelas variáveis pc6 e pc9 e o Fator 3 pelas variáveis pc7 e pc11.

Uma vez definidas as variáveis que representam as dimensões latentes, foi necessário rotulá-las. Cabe ressaltar que o rótulo é elaborado intuitivamente pelo pesquisador e sustentado pelo delineamento do modelo teórico. Apoiado nos resultados da AF, a dimensão “Reconhecimento de Padrão” foi decomposta em três subdimensões e descritas as respectivas habilidades no Quadro 2.

Quadro 2 – Subdimensões e descritores das habilidades da dimensão “Reconhecimento de Padrão”

<b>Estrutura de repetição por meio de instruções não escritas</b>	pc6	<b>rp1</b> – Reconhecer um padrão simples por meio de diagramas.
	pc9	
<b>Estrutura de repetição simples</b>	pc11	<b>rp2</b> – Reconhecer um padrão fazendo uso de estruturas de repetição simples.
	pc7	
<b>Estrutura de repetição composta pré ou pós-testada</b>	pc12	<b>rp3</b> – Reconhecer um padrão fazendo uso de estruturas de repetição compostas em diferentes contextos.
	pc8	
	pc22	

Fonte: Elaborado pelo autor.

Para a definição da dimensão denominada “Algoritmo” foi realizado um processo análogo ao feito na dimensão “Reconhecimento de Padrão”. Após gerada a matriz de correlação tetracórica foram desconsideradas as variáveis que apresentaram baixa comunalidade, mantendo aquelas que se aproximam ou ultrapassam 0,5. A ideia da consistência interna é de que os indicadores individuais devam medir o mesmo construto e serem altamente correlacionados entre si. Na prática, deseja-se obter uma comunalidade que exceda 0,5, e que as correlações entre os itens ultrapassem 0,3 (HAIR JR. *et al.*, 2009). Os três fatores retidos representam 67,40% da variância das oito variáveis, considerados suficientes em termos de variância total explicada. A Tabela 2 apresenta a matriz fatorial rotacionada com cargas significantes agrupadas em ordem decrescente em cada fator e sua respectiva comunalidade.

Tabela 2 – Resultado da AF para a Dimensão “Algoritmo”

Construtos				
	Fator 1	Fator 2	Fator 3	Comunalidade
<b>pc1</b>	0,864	-0,010	0,013	0,747
<b>pc2</b>	0,848	0,064	0,131	0,740
<b>pc4</b>	0,689	0,148	0,099	0,506
<b>pc24</b>	0,078	0,798	0,009	0,643
<b>pc14</b>	0,057	0,765	0,120	0,603
<b>pc13</b>	0,061	0,720	0,123	0,537
<b>pc17</b>	0,108	0,096	0,889	0,810
<b>pc18</b>	0,099	0,128	0,883	0,806

Método de Extração: Análise de Componentes Principais.

Rotação: Varimax

Variância total explicada: 67,40%

Total de observações: 462

Fonte: Elaborada pelo autor.

Em relação à comunalidade, nenhum critério estatístico diz o que é pequeno ou grande para a quantidade de variância numa variável que é explicada pelos três fatores tomados juntos. Assim, nesta análise foi considerado um nível mínimo de 0,5. Como nenhuma das variáveis demonstrou comunalidade abaixo dessa referência, possibilitou uma classificação mais precisa dos indicadores relacionados à dimensão “Algorítmica”. Dessa forma, foi possível concluir que o Fator 1 será estabelecido pelas variáveis pc1, pc2 e pc4, o Fator 2 pelas variáveis pc24, pc14 e pc13 e o Fator 3 integrado pelas variáveis pc17 e pc18. Diante dos resultados expostos pela AF, a dimensão “Algoritmo” foi decomposta em três subdimensões, apresentadas no Quadro 3.

Quadro 3 – Subdimensões e descritores das habilidades da dimensão “Algoritmo”

<b>Sequência simples</b>	pc1	<b>alg1</b> – Reconhecer o sequenciamento simples de instruções para solução de um problema em diferentes contextos.
	pc2	
	pc4	
<b>Sequência com uso de condicional simples</b>	pc24	<b>alg2</b> – Reconhecer o sequenciamento de instruções para solução de um problema fazendo uso de estruturas de decisão simples e repetição em diferentes contextos.
	pc14	
	pc13	
<b>Sequência com uso de condicional composta</b>	pc17	<b>alg3</b> – Reconhecer o sequenciamento de instruções para solução de um problema fazendo uso de estruturas de decisão composta e repetição num contexto visual.
	pc18	

Fonte: Elaborado pelos autores.

Por fim, foram selecionadas do Teste de Pensamento Computacional Espanhol que visam a descrever a dimensão proposta pelo modelo deste trabalho denominada “Decomposição”. Buscou-se facilitar a leitura dos fatores extraídos; então, a rotação ortogonal *Varimax* foi novamente aplicada na análise. A Tabela 3 mostra que foram mantidos dois fatores para interpretação que explicam, juntos, 63,39% da variância de todas as variáveis, considerados suficientes em termos de variância total explicada.

Tabela 3 – Resultado da AF para a Dimensão “Decomposição”

<b>Construtos</b>			
	<b>Fator 1</b>	<b>Fator 2</b>	<b>Comunalidade</b>
<b>pc28</b>	0,785	0,125	0,632
<b>pc27</b>	0,783	-0,036	0,615
<b>pc25</b>	-0,142	0,819	0,692
<b>pc26</b>	0,262	0,727	0,597

Método de Extração: Análise de Componentes Principais.  
 Rotação: Varimax  
 Variância total explicada: 67,40%  
 Total de observações: 462

Fonte: Elaborada pelos autores.

Nessa dimensão não foi oportuna a eliminação de nenhuma variável do modelo, uma vez que todas elas apresentaram comunalidades acima de 0,5. Duas subdimensões foram observadas, sendo o Fator 1 composto pelas variáveis pc27 e pc28 e o Fator 2 pelas variáveis pc25 e pc26. Diante da Análise Fatorial, a dimensão “Decomposição” foi subdividida conforme apresentada no Quadro 4.

Quadro 4 – Subdimensões e descritores das habilidades da dimensão “Decomposição”

Invocar função em contexto visual sem passagem de parâmetros	pc28	<b>dec1</b> – Reconhecer a decomposição do problema em partes menores num contexto visual.
	pc27	
Invocar função em contexto geométrico sem passagem de parâmetro	pc25	<b>dec2</b> – Reconhecer a decomposição do problema em partes menores num contexto geométrico.
	pc26	

Fonte: Elaborado pelos autores.

A Análise Fatorial Exploratória mostrou-se, na prática, uma técnica em que o pesquisador tem pouco controle sobre a especificação da estrutura, como o número de fatores extraídos e cargas de cada variável. Para uma tentativa de confirmação desse modelo, irá exigir o emprego de métodos específicos de Modelagem de Equações Estruturais, que serão discutidos a seguir. Antes disso, optou-se pela realização de uma análise fatorial considerando todas as variáveis do modelo para, num viés confirmatório, verificar o agrupamento para os fatores latentes.

## MODELAGEM DE EQUAÇÕES ESTRUTURAIS PARA O DESENVOLVIMENTO DA ESCALA DE PROFICIÊNCIA

A aplicação da Modelagem de Equações Estruturais (MEE) apoiou-se nos estágios propostos por Hair Jr. *et al.* (2009), que se iniciou com o desenvolvimento de um modelo teórico seguido pela construção do diagrama de caminhos em que se definem os construtos endógenos e exógenos, procedido da avaliação das estimativas do modelo e da qualidade do ajuste no mesmo.

De acordo com Hair Jr. *et al.* (2009), as variáveis latentes não são perceptíveis diretamente, pois são representadas por múltiplas variáveis, porém podem ser mensuradas por meio de indicadores. Dessa forma, no domínio da MEE, as dimensões, subdimensões e o próprio índice da capacidade de Pensamento Computacional, são estabelecidos por meio de variáveis latentes. Assim, o modelo proposto neste estudo apresentada três níveis hierárquicos de variável latente: VL de primeira ordem (subdimensões), VL de segunda ordem (dimensões) e VL de terceira (índice da capacidade de Pensamento Computacional). Para a modelagem do índice de capacidade do Pensamento Computacional foi utilizado o *software* SmartPLS 3.0<sup>5</sup>.

Recentemente, a modelagem de equações estruturais, baseada em *Partial Least Squares* (PLS), tem ganho maior destaque na literatura acadêmica (ZWICKER; SOUZA, BIDO, 2008). A modelagem de equações estruturais com estimação por mínimos quadrados parciais (PLS-SEM) tem sido empregada nas mais variadas áreas de pesquisa, e o *software* SmartPLS tem facilitado o uso do PLS-SEM por meio de uma abordagem baseada na variância, em que as variáveis latentes (VLs) são combinações lineares dos indicadores, podendo existir modelos formativos ou reflexivos. Em relação à teoria,

<sup>5</sup> O SmartPLS pode ser obtido no *site* <https://www.smartpls.com/downloads>, por meio do pedido de uma licença de uso gratuita, válida por 30 dias. Para isso, é necessário fazer download do programa, instalar e, na tela de licenciamento, escolher a opção profissional com todos os recursos e informar a chave de licença que foi enviada por *e-mail*. O *software* também possui modo estudante, no qual não é necessária a utilização da chave de ativação, porém os recursos são limitados.

apresenta-se mais flexível, mais bem adequada para um contexto exploratório, com significância dos parâmetros estimada por métodos não paramétricos, como o *bootstrap*<sup>6</sup> (ZWICKER; SOUZA; BIDO, 2008).

A MEE, baseada em variância (VB-SEM), calcula as correlações entre os construtos e suas variáveis mensuradas ou itens (modelos de mensuração) e, em seguida, são realizadas regressões lineares entre construtos (modelos estruturais). Dessa forma, consegue-se conjecturar modelos complexos com número menor de dados (RINGLE; SILVA; BIDO, 2014).

No campo das Ciências Sociais, a MEE, com estimação por mínimos quadrados parciais (PLS-SEM), tem se mostrado uma excelente possibilidade para a avaliação de relações entre construtos e se adequa bem a situações em que as relações causais têm pouca sedimentação, o que permite ser usada de forma mais exploratória (BIDO; SILVA, 2019).

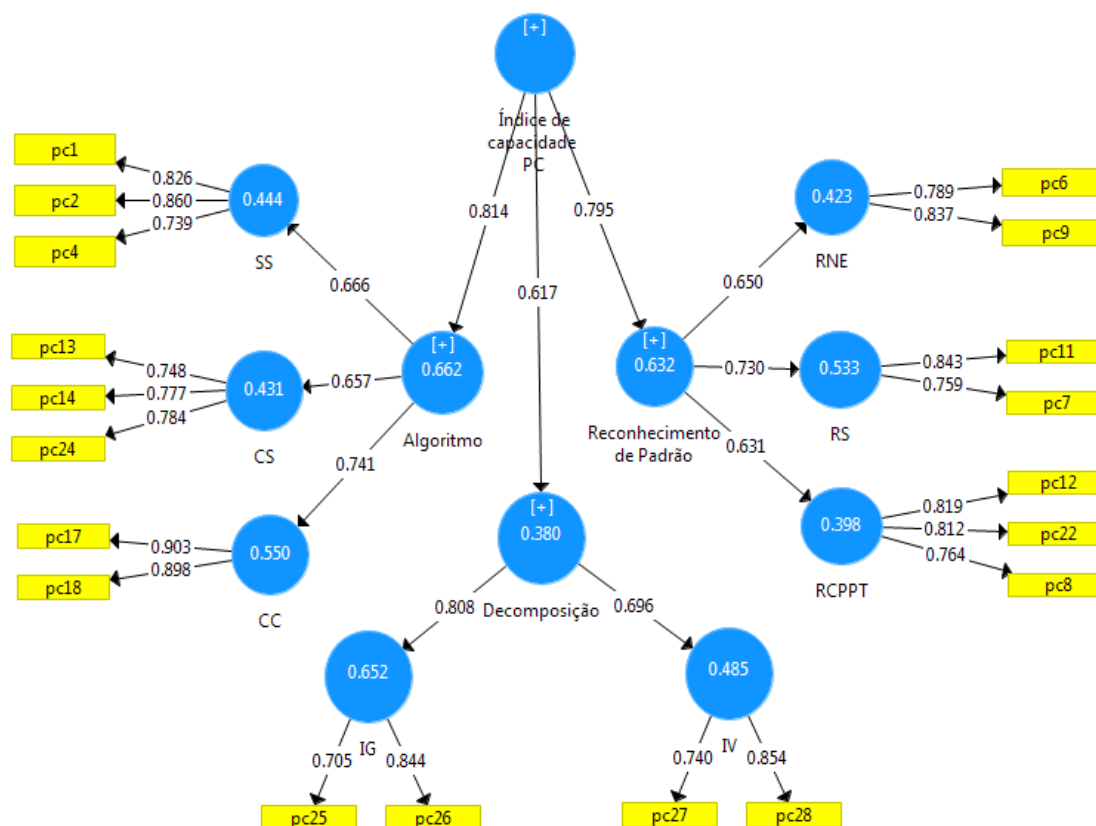
No modelo SEM há muitas formas para especificar as relações, inclusive em notação matemática, o que pode ser complexo. Desse modo, pesquisadores optam por um modelo visual conhecido como diagrama de caminhos, que mostra as convenções específicas tanto para construtos quanto para variáveis medidas e as relações entre elas. A Figura 1 apresenta o MEE inicial completo, depois de realizada todas as ligações necessárias, preparado para iniciar as análises.

Como observado na Figura 1, propõe-se um modelo no qual não há construtos exógenos, ou seja, variáveis latentes determinadas por fatores externos ao modelo. Diante disso, verifica-se a presença de construtos endógenos, teoricamente determinados por fatores dentro do modelo. Nessa construção, um construto é dependente de outros construtos, e esta dependência é visualmente representada por um caminho que chega a um construto endógeno a partir de um exógeno (HAIR JR. *et al.*, 2009). Dessa forma, o único construto exógeno do modelo é o próprio índice da capacidade do Pensamento Computacional.

O modelo de variáveis latentes hierárquico utilizando da modelagem de caminhos PLS proposto neste trabalho, seguiu as especificações apresentadas por Bido e Silva (2019), na qual um construto latente de segunda ordem é mensurado por construtos de primeira ordem e assim sucessivamente. Dessa maneira, uma possibilidade na modelagem foi reutilizar os indicadores dos construtos de primeira ordem nos construtos de ordem superior, formando, assim, um modelo hierárquico reflexivo. O algoritmo PLS aplicado ao modelo apresenta, em seu *output*, correlações entre os construtos de primeira e segunda ordem, conforme mostra a Figura 1.

<sup>6</sup> A abordagem *Bootstrapping*, com objetivo de validação, consiste em extrair um grande número de subamostras, estimando modelos para cada uma delas.

Figura 1 – Diagrama de Caminhos e suas respectivas cargas fatoriais geradas pelo método PLS-SEM



Fonte: Elaborada pelos autores, gerada pelo SmartPLS.

O modelo foi submetido à abordagem *Bootstrapping*, com objetivo de validação, extraíndo-se um grande número de subamostras e estimando modelos para cada uma delas, apresentando, dessa forma, coeficientes melhores ajustados.

Com isso, foi possível constatar a estabilidade do modelo apresentado por meio da técnica de reamostragem com reposição, a qual mostra que todos os valores das relações indicador-construto e dos construtos-construtos estão acima do valor de referência de 1,96. Dessa forma, em todos os casos rejeita-se a hipótese nula e pode-se afirmar que as correlações e os coeficientes de regressão são significantes, podendo inferir que o modelo sugere estabilidade, e pode ser considerado bom, uma vez que o número de variâncias que ele pode explicar é maior do que o número que ele não pode explicar.

## VALIDAÇÃO E CONFIABILIDADE DO MODELO PROPOSTO

Segundo Hair Jr. *et al.* (2009), o primeiro aspecto a ser observado dos modelos de mensuração são as Validades Convergentes, uma vez que os indicadores de um construto devem compartilhar elevada proporção da variância comum. Para isso, verifica-se o indicador resumido de convergência do percentual médio da variância extraída, no inglês *Average Variance Extracted* (AVEs).

Depois de garantir a validade convergente (AVE) do modelo, é indicado por Hair Jr. *et al.* (2009) estimar a confiabilidade dos construtos latentes por meio da *Composite Reliability* (CR). No SmartPLS os indicadores da AVE e CR para os construtos de segun-

da e terceira ordem foram calculados com base nos indicadores que foram repetidos e ocultados durante a construção do modelo. Assim, todos os valores para AVEs e CRs para construtos de segunda e terceira ordem foram calculados utilizando a carga fatorial apresentada no diagrama de caminho com o auxílio de uma planilha eletrônica, conforme mostra a Figura 2.

Figura 2 – Valores da AVE e CR dos construtos de segunda e terceira ordem

	A	B	C	D	E	F	G	H
1	<b>Algoritmo</b>		<b>Rec. de Padrão</b>		<b>Decomposição</b>		<b>Índice PC</b>	
2	<b>Carga</b>	<b>1 - carga<sup>2</sup></b>	<b>Carga</b>	<b>1 - carga<sup>2</sup></b>	<b>Carga</b>	<b>1 - carga<sup>2</sup></b>	<b>Carga</b>	<b>1 - carga<sup>2</sup></b>
3	0,666	0,556	0,650	0,578	0,808	0,347	0,814	0,337
4	0,657	0,568	0,730	0,467	0,696	0,516	0,617	0,619
5	0,741	0,451	0,631	0,602			0,795	0,368
6								
7	<b>AVE</b>	<b>0,475</b>	<b>AVE</b>	<b>0,451</b>	<b>AVE</b>	0,569	<b>AVE</b>	0,558
8	<b>CR (CC)</b>	0,730	<b>CR (CC)</b>	0,711	<b>CR (CC)</b>	0,724	<b>CR (CC)</b>	0,789

Fonte: Elaborada pelos autores.

Apenas dois construtos do MEE apresentaram valor da AVE < 0,50, e todos indicaram consistência interna adequada, uma vez que indicadores CR foram superiores a 0,7. Nessas situações sugere-se suprimir as variáveis mensuradas dos construtos que apresentam a AVE < 0,50 com o objetivo de aumentar o seu valor, eliminando as variáveis com cargas fatoriais de menor valor. Conforme preconizado por Hair Jr. *et al.* (2009, p. 592), porém, a confiabilidade composta também é um indicador de validade convergente e ambos devem ser analisados conjuntamente.

Apesar, contudo, de a AVE, para esses construtos, ter apresentado valores abaixo da referência, ao mesmo tempo mostrando indicadores próximos do aceitável (0,475 para o construto Algoritmo e 0,451 para o construto Reconhecimento de Padrão) e CC acima da referência (0,730 e 0,711 respectivamente), decide-se pela manutenção do modelo, inferindo que todas as medidas são consistentes e representam o mesmo construto latente.

A última etapa para a avaliação do MEE, proposta por Hair Jr. *et al.* (2009), é a validade discriminante (VD), que pode ser entendida como o indicador, em que um construto difere dos demais, ou seja, identifica o grau em que um construto é único. No SmartPLS, na seção Validade Discriminante, foi possível encontrar a matriz de correlação de Pearson entre os construtos pelo critério Fornell-Larcker que foram satisfatórios.

Findado o processo de validação do MEE, os itens estatisticamente comportaram-se como esperado e não foi necessária a eliminação de outras variáveis para elaboração e análise de um modelo derivativo. A Análise Fatorial possibilitou a redução instrumental, na qual foi possível eleger os melhores itens que comungavam de uma boa comunalidade dentro de cada construto.

Com isso, foi possível dar uma nova roupagem ao teste espanhol, organizando-o em três pilares do PC, de forma a não comprometer suas propriedades métricas, apresentando consistência interna aceitável para cada construto latente, bem como uma estrutura mais bem definida e equilibrada.



Os resultados dessa reorganização, além de selecionar os melhores itens votados à realidade dos estudantes e da escola brasileira ao término do Ensino Fundamental (14 e 15 anos), mostrou-se um instrumento abreviado válido e confiável para medir o Pensamento Computacional dos estudantes sob a ótica do Pensamento Algorítmico, Recursivo e Decompositivo.

O delineamento do modelo possibilitou testar a estabilidade do instrumento reduzido, ou seja, a sua capacidade de reproduzir os mesmos resultados em aplicações sucessivas por meio de técnica de reamostragem *bootstrap* que se apresentaram satisfatórias.

Partindo do pressuposto de que todo processo do mundo real, simplificado por meio de um modelo, apresenta limitações, os resultados conquistados no processo de validação do instrumento permitiram a abreviação do instrumento espanhol, que ancorou a mensuração da capacidade do Pensamento Computacional dos estudantes ao término do Ensino Fundamental no contexto de uma cidade do oeste paulista.

Após a Análise Fatorial Confirmatória fornecer um ajuste aceitável e mostrar evidência de validade dos construtos, pode-se inferir que a teoria de mensuração é sustentada. Dessa forma, foi possível associar, a cada observação da amostra, um escore fatorial padronizado para o índice da capacidade do Pensamento Computacional.

Em estatística, geralmente os escores ocorrem numa distribuição normal com a média zero e o desvio padrão de 1. Dessa forma, a normalização foi usada para dimensionar os dados entre zero e 1 (FIELD, 2009). Os resultados são apresentados na Tabela 4.

Tabela 4 – Estatísticas descritivas dos escores fatoriais normalizados

	Média	Mediana	Min	Máx	Desvio-padrão	Coef. Variação	N
<b>Índice de capacidade_PC</b>	0,4972	0,4895	0	1	0,2098	42,19%	462
<b>Algoritmo</b>	0,5888	0,5952	0	1	0,2584	43,88%	462
<b>Reconhecimento de Padrão</b>	0,5486	0,5172	0	1	0,2246	40,94%	462
<b>Decomposição</b>	0,4605	0,4408	0	1	0,2976	64,62%	462

Fonte: Elaborada pelos autores.

Nesse contexto, foi possível criar uma métrica que possibilitou verificar a capacidade do Pensamento Computacional demonstrada pelo participante. Assim, uma escala contempla os níveis de maturidade a partir de faixas criadas com base nos quartis, que permitiu uma avaliação qualitativa e a elaboração de expectativas das habilidades mobilizadas e articuladas pelos estudantes, dispostas numa escala progressiva e cumulativa. As faixas foram criadas baseadas nos quartis e dividiram a escala em quatro grupos, contendo aproximadamente igual número de observações. O Quadro 5 apresenta o nível de maturidade do Pensamento Computacional e as capacidades esperadas em cada um deles.

Quadro 5 – Níveis de maturidade e capacidade do PC ao término do Ensino Fundamental (MMPC\_EF)

Nível	Intervalo	Expectativas de operações cognitivas mobilizadas de Pensamento Computacional
<b>Nível 4 Avançado</b>	0,62851153 a 1	Além das habilidades anteriores, os alunos reconhecem a decomposição do problema e, por meio de uma visão sistêmica, são capazes de compor um algorítmico que mobilize estruturas de repetição compostas, além de construírem algoritmos para invocar funções em diferentes contextos sem passagem de parâmetros.
<b>Nível 3 Proficiente</b>	0,48951782 a 0,62851153	Além das habilidades anteriores, os estudantes reconhecem o padrão em diferentes contextos e conseguem compor um algoritmo, mobilizando estruturas de repetição pré ou pós-testadas.
<b>Nível 2 Básico</b>	0,35036688 a 0,48951782	Além das habilidades anteriores, os estudantes reconhecem um padrão em contextos não escritos e conseguem compor um algoritmo com decisões compostas e estruturas simples de repetição.
<b>Nível 1 Insuficiente</b>	0 até 0,35036688	Os estudantes com desempenho nessa faixa requerem atenção especial, pois demonstram habilidades muito elementares daquelas que deveriam apresentar ao término do Ensino Fundamental. Nesse nível, os estudantes são capazes de sequenciar uma instrução simples com decisão condicional simples.

Fonte: Elaborado pelos autores.

A escala ficou organizada em quatro níveis progressivos e cumulativos, da menor para a maior capacidade, ou seja, quando um estudante está posicionado em determinado nível da escala pressupõe-se que, além de ter desenvolvido as habilidades referentes ao nível atual, provavelmente desenvolveu as habilidades referente aos níveis anteriores.

## CONSIDERAÇÕES FINAIS

Apoiados por fatores relacionados às dimensões do Pensamento Computacional, definidos por um Modelo de Equações Estruturais resultante da Análise Fatorial Confirmatória, esses construtos foram considerados áreas-chave, análogos às KPAs do modelo CMM, denominados construtos de segunda ordem. Diante disso, duas formas de representação podem ser observadas: o nível de capacidade, quando apenas uma KPA pode ser avaliada isoladamente, ou uma representação em nível de maturidade, em que um grupo de KPAs associado a um nível de maturidade específico pode ser avaliado.

O modelo foi composto por 4 níveis de maturidade determinados dentro do intervalo (zero a 1), em que uma capacidade madura de Pensamento Computacional para a idade é representada quando se atinge o nível máximo de capacidade em todas as dimensões.

Dessa forma, esse modelo de maturidade do PC tem como objetivo ajudar professores a avaliar e melhorar o seu processo de ensino. Também pode ser usado pela instituição educacional para avaliar a qualidade do ensino de fundamentos da computação, e, quando necessário, traçar intervenções no processo de ensino e aprendizagem.

Partimos do pressuposto de que todo modelo é limitado, considerando o domínio amplo do PC, mas o modelo apresentado no Quadro 5 dará suporte para apoiar decisões, considerada a ausência de métricas para o PC no cenário brasileiro.

O modelo proposto foi intitulado MMPC-EF (Modelo de Maturidade em PC no Ensino Fundamental), em três domínios que sustentam o Pensamento Computacional: Algoritmo, Reconhecimento de Padrão e Decomposição, em que cada um dos domínios gera uma estrutura hierárquica mensurável, e estão relacionados a um terceiro nível hierárquico chamado de “escala/índice do Pensamento Computacional”, permitindo avaliar o modelo de forma quantitativa e qualitativamente.

Escolas mais maduras melhoraram seus níveis de desempenho significativamente e de maneira mais rápida daquelas que não são, pois conhecem as fragilidades e conseguem realizar uma intervenção pontual mais eficaz. O modelo de maturidade do PC é uma propriedade da rede de ensino de Santa Cruz do Rio Pardo – SP – que, certamente, apoiará os professores nas tomadas de decisão em sala de aula e uma possível contribuição na coerência pedagógica para o ensino dos fundamentos da Ciência da Computação.

## REFERÊNCIAS

- ALBINO, R. D. *Uma visão integrada sobre o nível de uso das Tecnologias da Informação e Comunicação em escolas brasileiras*. 2015. 167 f. Dissertação (Mestrado em Administração) – USP, Faculdade de Economia, Administração e Contabilidade, 2015.
- ALMEIDA, M. E. B.; VALENTE, J. A.; KUIN, S.; SILVA, J. M. O currículo na cultura digital e a integração currículo e tecnologias. In: CERNY, R. Z. et al. (org.). *Formação de educadores na cultura digital*. Florianópolis: UFSC; CED; NUP, 2017. Disponível em: [https://nupced.paginas.ufsc.br/files/2017/06/PDF\\_Formacao\\_de\\_Educadores\\_na\\_Cultura\\_Digital\\_a\\_construcao\\_coletiva\\_de\\_uma\\_proposta3.pdf](https://nupced.paginas.ufsc.br/files/2017/06/PDF_Formacao_de_Educadores_na_Cultura_Digital_a_construcao_coletiva_de_uma_proposta3.pdf). Acesso em: 25 ago. 2020.
- ARAÚJO, M. H. *Análise de fatores que influenciam o uso de serviços de governo eletrônico no Brasil*. 2013. 121 f. Dissertação (Mestrado) – Universidade de São Paulo, 2013.
- BARR, V.; STEPHENSON, C. Bringing computational thinking to K-12: what is involved and what is the role of the computer science education community? *Inroads*, n. 2(1), p. 48-54, 2011. Disponível em: <https://id.iste.org/docs/nets-refresh-toolkit/bringing-ct-to-k-12.pdf>. Acesso em: 12 ago. 2020.
- BBC. British Broadcasting Corporation. *Introduction to computational thinking*, 2018. Disponível em: <https://www.bbc.co.uk/education/guides/zp92mp3/revision/1>. Acesso em: 20 ago. 2020.
- BELL, T.; WITTEN, I. H.; FELLOWS, M. *Computer Science Unplugged*, 2015. Disponível em: [http://csunplugged.org/wpcontent/uploads/2015/03/CSUnplugged\\_OS\\_2015\\_v3.1.pdf](http://csunplugged.org/wpcontent/uploads/2015/03/CSUnplugged_OS_2015_v3.1.pdf). Acesso em: 8 jul. 2020.
- BERLAND, M.; LEE, V. R. Collaborative strategic board games as a site for distributed computational thinking. *International Journal of Game-Based Learning*, n. 1(2), p. 65-81, 2012.
- BIDO, D. S.; SILVA, D. SMARTPLS 3: especificação, estimação, avaliação e relato. *Raep – Administração: Ensino e Pesquisa*, v. 20, n. 2, 2019. Disponível em: <https://www.redalyc.org/jatsRepo/5335/533559666009/html/index.html>. Acesso em: 25 ago. 2020.
- BRASIL. *Base Nacional Comum Curricular*. Brasília: MEC, 2017. Disponível em: <http://basenacionalcomum.mec.gov.br/abase/#introducao>. Acesso em: 25 ago. 2020.
- BRENNAN, K.; RESNICK, M. New frameworks for studying and assessing the development of computational thinking. In: ANNUAL MEETING OF THE AMERICAN EDUCATIONAL RESEARCH ASSOCIATION, 2012, Vancouver, p. 1-25. Disponível em: [https://web.media.mit.edu/~kbrennan/files/Brennan\\_Resnick\\_AERA2012\\_CT.pdf](https://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf). Acesso em: 25 ago. 2020.
- DAVIS, S. R. *Começando a programar em C++ para leigos*. São Paulo: Alta Books, 2011.
- DENNER, J.; WERNER, L.; CAMPE, S.; ORTIZ, E. Pair programming: Under what conditions is it advantageous for middle school learners? *Journal of Research on Technology in Education*, n. 46(3), p. 277-296, 2014.

- DENNING, P. J. The profession of IT: Beyond computational thinking. *Communications of the ACM*, n. 52(6), p. 28-30, 2009. Disponível em: <http://denninginstitute.com/pjd/PUBS/CACMcols/cacmJun 09.pdf>. Acesso em: 25 ago. 2020.
- DOWNEY, A.; ELKNER, J.; MEYERS, C. *How to Think Like a Computer Scientist: Learning with Python*. Wellesley: Green Tea Press, 2002.
- DUNCAN, C.; BELL, T. A pilot computer science and programming course for primary school students. In: PROCEEDINGS OF THE 10TH WORKSHOP IN PRIMARY AND SECONDARY COMPUTING EDUCATION, 2015, Londres, 2015. p. 39-48.
- FORNELL, C.; LARCKER, D. F. Evaluating structural equation models with unobservable variables and measurement error. *Journal of Marketing Research*, v. 18, p. 39-50, 1981.
- FIELD, A. *Descobrimos a estatística usando o SPSS*. 2. ed. Porto Alegre: Artmed, 2009. p. 156-168.
- FRAILLON, J.; AINLEY, J.; SCHULZ, W.; DUCKWORTH, D.; FRIEDMAN, T. *IEA International Computer and Information Literacy Study 2018 Assessment Framework*. Amsterdam: Springer, 2019.
- GOOGLE. *Exploring computational thinking*, 2018. Disponível em: [https://computationalthinkingcourse.withgoogle.com/course?use\\_last\\_location=true](https://computationalthinkingcourse.withgoogle.com/course?use_last_location=true)[https://computationalthinkingcourse.withgoogle.com/course?use\\_last\\_location=true](https://computationalthinkingcourse.withgoogle.com/course?use_last_location=true). Acesso em: 20 jan. 2019.
- GOUWS, L.; BRADSHAW, K.; WENTWORTH, P. First year student performance in a test for computational thinking. In: SAICSIT '13: PROCEEDINGS OF THE SOUTH AFRICAN INSTITUTE FOR COMPUTER SCIENTISTS AND INFORMATION TECHNOLOGISTS CONFERENCE, 2013. p. 271-277.
- GROVER, S. Systems of Assessments for deeper learning of computational thinking in K-12. In: PROCEEDINGS OF THE 2015 ANNUAL MEETING OF THE AMERICAN EDUCATIONAL RESEARCH ASSOCIATION, 2015. Chicago, 2015. p. 15-20.
- GROVER, S.; PEA, R. Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, v. 42, n.1, p. 38-43, 2013. Disponível em: [http://multimedia.uoc.edu/carlos/chipro/wp-content/uploads/2013/10/38.full\\_.pdf](http://multimedia.uoc.edu/carlos/chipro/wp-content/uploads/2013/10/38.full_.pdf). Acesso em: 25 ago. 2020.
- HAIR JR.; J. F.; BLACK, W. C.; BABIN, B. J.; ANDERSON, R. E.; TATHAM, R. L. *Análise multivariada de dados*. 6. ed. Porto Alegre: Bookman, 2009.
- ISTE; CSTA. International Society of Technology in Education; Computer Science Teachers Association. *Computational thinking in K-12 education leadership toolkit*, 2011. Disponível em: <https://id.iste.org/docs/ct-documents/ct-leadership-toolkit.pdf?sfvrsn=4>. Acesso em: 25 ago. 2020.
- KONG, S. C. A framework of curriculum design for computational thinking development in K-12 education. *Journal of Computers in Education*, n. 3, p. 377-394, 2016. Disponível em: [https://www.researchgate.net/publication/311004023\\_A\\_framework\\_of\\_curriculum\\_design\\_for\\_computational\\_thinking\\_development\\_in\\_K-12\\_education](https://www.researchgate.net/publication/311004023_A_framework_of_curriculum_design_for_computational_thinking_development_in_K-12_education). Acesso em: 20 ago. 2020.
- LEE, T. Y.; MAURIELLO, M. L.; AHN, J.; BEDERSON, B. B. CTArcade: Computational thinking with games in school age children. *International Journal of Child-Computer Interaction*, n. 2(1), p. 26-33, 2014. Disponível em: <http://www.cs.umd.edu/hcil/trs/2012-22/2012-22.pdf>. Acesso em: 25 ago. 2020.
- LING, T. C.; JUSOH, Y. Y.; ABDULLAH, R.; ALWI, N. H. A Review Study: Applying Capability Maturity Model in Curriculum Design Process for Higher Education. *Journal for the advancement of science & arts*, n. 3(1), p. 46-55, 2012.
- LITTLE, T. D. *Longitudinal Structural Equation Modeling*. New York: The Guilford Press, 2013.
- LYE, S. Y.; KOH, J. H. L. Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, n. 41, p. 51-61, 2014.
- MUELLER, J.; BECKETT, D.; HENNESSEY, E.; SHODIEV, H. Assessing computational thinking across the curriculum. In: RICH, P. J.; HODGES, C. B. (org.). *Emerging research, practice, and policy on computational thinking*. Switzerland: Springer, 2017. p. 251-267.
- PARANHOS, R.; FIGUEIREDO FILHO, D. B.; ROCHA, E. C.; SILVA JUNIOR, J. A. Corra que o survey vem aí. Noções básicas para cientistas sociais. *Revista Latinoamericana de Metodología de la Investigación Social*, n. 6, p. 7-24, 2014. Disponível em: [https://edisciplinas.usp.br/pluginfile.php/304292/mod\\_resource/content/1/Paranhos%20e%20tal\\_Corra%20que%20o%20survey%20vem%20a%C3%AD.pdf](https://edisciplinas.usp.br/pluginfile.php/304292/mod_resource/content/1/Paranhos%20e%20tal_Corra%20que%20o%20survey%20vem%20a%C3%AD.pdf). Acesso em: 25 ago. 2020.
- RILEY, D. D.; HUNT, K. A. *Computational thinking for the modern problem solver*. Boca Raton: CRC Press, 2014.
- RINGLE, C. M.; SILVA, D.; BIDO, D. Modelagem de equações estruturais com utilização do SmartPLS. *Revista Brasileira de Marketing*, v. 13, n. 2, p. 56-73, 2014.

- ROMÁN-GONZÁLEZ, M. *Codigoalfabetización y Pensamiento Computacional en Educación Primaria y Secundaria: Validación de un Instrumento y Evaluación de Programas*, 2016. Tese (Programa de Doutorado em Educação) – Universidad Nacional de Educación a Distancia, Madrid, Spain, 2016.
- 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. (org.). *Computational Thinking Education*. Singapore: Springer, 2019.
- SACRISTÁN, G. *O currículo: uma reflexão sobre a prática*. 3. ed. Porto Alegre: Artmed, 2000.
- SANTOS, C. P. *Estudo dos fatores influenciadores da intenção de uso da informação dos sistemas de Business Intelligence em empresas brasileiras*. 2014. Tese (Doutorado em Administração) – Faculdade de Economia, Administração e Contabilidade – USP, 2014.
- SBC. Sociedade Brasileira de Computação. *A importância do ensino de computação na Educação Básica*. Audiência Pública da Base Nacional Comum Curricular, São Paulo: 2017. Disponível em: <https://www.sbc.org.br/noticias/2002-sbc-participa-da-audiencia-publica-da-bncc-em-sao-paulo>. Acesso em: 25 ago. 2020.
- SEI. Software Engineering Institute. *Relatório Técnico CMMI para Desenvolvimento Versão 1.3*. Universidade Carnegie Mellon, 2010. Disponível em: [https://resources.sei.cmu.edu/asset\\_files/WhitePaper/2006\\_019\\_001\\_28945.pdf](https://resources.sei.cmu.edu/asset_files/WhitePaper/2006_019_001_28945.pdf). Acesso em: 25 ago. 2020.
- SEITER, L.; FOREMAN, B. Modeling the learning progressions of computational thinking of primary grade students. In: PROCEEDINGS OF THE NINTH ANNUAL INTERNATIONAL ACM CONFERENCE ON INTERNATIONAL COMPUTING EDUCATION RESEARCH, 2013, San Diego, 2013. p. 59-66.
- SHUTE, V. J.; SUN, C.; ASBELL-CLARKE, J. Demystifying computational thinking. *Educational Research Review*, 2017. Disponível em: <http://myweb.fsu.edu/vshute/pdf/CT.pdf>. Acesso em: 12 ago. 2020.
- SIQUEIRA, E. S. *Exclusão digital de pequenas e médias empresas brasileiras e os fatores que influenciam o uso das TICs nessas organizações*. 2014. 156 f. Dissertação (Mestrado) – Universidade de São Paulo, São Paulo, 2014.
- SOLAR, M.; SABATTIN, J.; PARADA, V. A Maturity Model for Assessing the Use of ICT in School Education. *Journal of Ed. Tech. & Society*, n. 16(1), p. 206-218, 2013.
- WEINTROP, D.; BEHESHTI, E.; HORN, M.; ORTON, K.; JONA, K.; TROUILLE, L.; WILENSKY, U. Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, v. 25, n. 1, p. 127-147, 2016.
- WERNER, L.; DENNER, J.; CAMPE, S.; KAWAMOTO, D. C. *The fairy performance assessment: Measuring computational thinking in middle school*. In: SIGCSE '12: PROCEEDINGS OF THE 43RD ACM TECHNICAL SYMPOSIUM ON COMPUTER SCIENCE EDUCATION, p. 215–220, 2012.
- WING, J. M. Computational thinking – what and why? *The magazine of Carnegie Mellon*. University's School of Computer Science, 2010. Disponível em: <https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>. Acesso em: 25 ago. 2020.
- WING, J. M. Computational thinking and thinking about computing. *Philosophical transactions of the royal society a mathematical, physical and engineering sciences*, v. 366, n. 1.881, p. 3.717-3.725, 2008.
- WING, J. M. Computational thinking benefits Society. *Social Issues in Computing*, 2014. Disponível em: <http://socialissues.cs.toronto.edu/index.html>. Acesso em: 25 ago. 2020.
- WING, J. M. Computational thinking. *Communications of the ACM*, v. 49, n. 3, p. 33-35, 2006. Disponível em: <https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>. Acesso em: 12 ago. 2020.
- WING, J. M. Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology*, n. 25(2), p. 7-14, 2017. Disponível em: <http://www.cs.cmu.edu/~wing/publications/Wing17.pdf>. Acesso em: 25 ago. 2020.
- ZANETTI, H.; BORGES, M.; RICARTE, I. Pensamento Computacional no ensino de programação: Uma revisão sistemática da literatura brasileira. In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO. 5., SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 27., 2016. *Anais [...]*. 2016. Disponível em: <https://www.br-ie.org/pub/index.php/sbie/article/view/6677/4566>. Acesso em: 12 ago. 2020.
- ZWICKER, R.; SOUZA, C. A.; BIDO, D. S. Uma revisão do Modelo do Grau de Informatização de Empresas: novas propostas de estimação e modelagem usando PLS (Partial Least Squares). In: ENCONTRO DA ASSOCIAÇÃO NACIONAL DOS PROGRAMAS DE PÓS-GRADUAÇÃO EM ADMINISTRAÇÃO, 32., 2008. Rio de Janeiro. *Anais [...]*. Rio de Janeiro: Anpad, 2008.