

Uma Definição Operacional para Pensamento Computacional

Bianca L. Santana
UFBA – Universidade Federal
da Bahia
Salvador, Bahia, Brasil
biancasantana.ls@gmail.com

Christina F. G. Chavez
UFBA – Universidade Federal
da Bahia
Salvador, Bahia, Brasil
flach@ufba.br

Roberto A. Bittencourt
UEFS – Universidade Estadual
de Feira de Santana
Feira de Santana, Bahia, Brasil
roberto@uefs.br

RESUMO

O Pensamento Computacional é um conceito em destaque na área de educação em computação e faz parte de diversos currículos de computação na escola. É frequentemente descrito como um processo de resolução de problemas por meio de habilidades cognitivas. Apesar disso, não há consenso sobre quais habilidades fazem parte do Pensamento Computacional. Consequentemente, também não há consenso sobre como medi-lo, o que dificulta o estabelecimento de estratégias de ensino-aprendizagem e de avaliação da aquisição do PC. Nesse contexto, apresentamos uma definição operacional para Pensamento Computacional que relaciona sua aquisição e a aprendizagem de programação de computadores, com o objetivo de oferecer um framework para a construção de avaliações da aquisição do Pensamento Computacional. A definição apresentada é baseada em diversos trabalhos da área. Ela foi aprimorada por meio de um painel de especialistas, composto por dez pesquisadores brasileiros da área de educação em computação. Nossa concepção considera que as habilidades de Pensamento Computacional são fomentadas através do exercício de conceitos, práticas e perspectivas. Listamos quais são estes conceitos, práticas, perspectivas e apresentamos uma definição para cada um deles. A definição operacional apresentada pode ser a base de avaliações que têm como público-alvo estudantes do Ensino Fundamental II ou níveis posteriores de educação.

CCS CONCEPTS

• **Social and professional topics** → **Computational thinking.**

PALAVRAS-CHAVE

Pensamento Computacional, Avaliação, Framework.

1 INTRODUÇÃO

O Pensamento Computacional (PC) é considerado uma competência importante para o século XXI e está presente em diversas experiências de Educação em Computação apresentadas na literatura [13, 15, 22]. PC também aparece como um eixo em diversos currículos de Computação para escolas no mundo e no Brasil, como no *CSTA Computer Science Standards*, *AP Computer Science Principles Course and Exam Description*, *Referenciais Curriculares de Computação para a Educação Básica da Sociedade Brasileira de Computação*

Fica permitido ao(s) autor(es) ou a terceiros a reprodução ou distribuição, em parte ou no todo, do material extraído dessa obra, de forma verbatim, adaptada ou remixada, bem como a criação ou produção a partir do conteúdo dessa obra, para fins não comerciais, desde que sejam atribuídos os devidos créditos à criação original, sob os termos da licença CC BY-NC 4.0.

EduComp'21, Abril 27–30, 2021, Jataí, Goiás, Brasil (On-line)

© 2021 Copyright mantido pelo(s) autor(es). Direitos de publicação licenciados à Sociedade Brasileira de Computação (SBC).

(SBC) e o *Currículo de Tecnologia e Computação da Educação Infantil ao Ensino Fundamental* [5, 23, 24, 29].

Apesar do interesse crescente em ofertar experiências que fomentem a aquisição de PC, não há uma definição operacional amplamente aceita para PC na literatura [25]. PC é geralmente definido como um processo de resolução de problemas relacionado a um conjunto de atividades cognitivas e metacognitivas. No entanto, não há consenso quanto às definições formais de PC ou aos frameworks de PC [7]. A falta de definição afeta o modo como o PC pode ser medido, dificultando a avaliação das experiências de ensino e subsequente comparação entre experiências diferentes [7]. Em consequência disso, há uma escassez de avaliações amplamente aceitas para medir as habilidades de PC [32]. Avaliar a sua aquisição de forma eficaz é importante para integração bem sucedida de currículos e demais iniciativas [12].

Nesse contexto, apresentamos uma definição operacional de PC, que pode ser empregada como framework para elaboração de avaliações. A definição apresentada considera que as habilidades de PC são fomentadas através do exercício de conceitos, práticas e perspectivas. O objetivo deste trabalho é apresentar, avaliar e melhorar nossa definição operacional de pensamento computacional. As questões de pesquisa associadas são: Do ponto de vista de um painel de especialistas, qual a avaliação sobre i) os conceitos de PC e suas definições?; ii) as práticas de PC e suas definições?; e iii) as perspectivas de PC e suas definições?

Nosso trabalho baseia-se em definições apresentadas na literatura e foi submetido a um painel de especialistas composto por pesquisadores da área de Educação em Computação. Neste artigo, apresentamos a definição inicial, a avaliação do painel de especialistas e uma definição melhorada a partir dos resultados do painel. Este trabalho consiste em uma das etapas de um trabalho mais amplo, com o objetivo de propor uma avaliação de PC [21].

O restante deste artigo está organizado da seguinte forma. Primeiramente, apresentamos o referencial teórico que fundamenta esta pesquisa, o que é PC e quais as suas definições formais e frameworks operacionais. Em seguida, descrevemos a metodologia empregada para realizar o painel de especialistas, apresentamos os resultados e as respectivas discussões. O artigo termina com a apresentação da definição operacional revisada, descrevendo ainda as principais contribuições desse trabalho, suas limitações e recomendações para trabalhos futuros.

2 O QUE É PENSAMENTO COMPUTACIONAL

Segundo Wing [31], PC envolve a resolução de problemas, a concepção de sistemas e a compreensão do comportamento humano, baseando-se nos conceitos fundamentais da computação. Apesar de fomentar uma ampla discussão no campo da Educação em Computação, Wing não definiu em seu trabalho inaugural o que o termo

Pensamento Computacional significa exatamente. Algumas definições operacionais surgiram posteriormente na literatura. Uma definição operacional para PC é dada posteriormente por Barr et al. [3], afirmando que PC envolve uma combinação de habilidades que, quando usadas, fornecem a base de uma nova forma de solucionar problemas. Estas habilidades são: a capacidade de formular problemas, representar dados por meio de abstrações, automatizar soluções por meio do pensamento algorítmico, identificar analisar e implementar soluções, além da habilidade de generalização. Ainda na concepção de Barr et al. [3], PC é aprimorado por uma série de disposições ou atitudes como a confiança em lidar com a complexidade, persistência no trabalho com problemas difíceis, tolerância em relação a ambiguidades, capacidade de lidar com problemas abertos, e a capacidade de se comunicar e trabalhar com outras pessoas para alcançar um objetivo ou solução comum.

Mesmo hoje, apesar do amplo destaque que PC recebe, não há consenso na literatura sobre uma definição para o termo. Há muitas definições operacionais disponíveis na literatura, e isso dificulta o estabelecimento das abordagens que potencializam o PC [12]. De maneira geral, PC é visto como um processo de resolução de problemas, que pode ser decomposto em várias dimensões cognitivas e não cognitivas em um conjunto de habilidades, incluindo abstração, decomposição, avaliação, reconhecimento de padrões, lógica e design de algoritmos [11].

Um ponto de discussão pertinente ao tratar do ensino-aprendizagem de PC é o papel da programação de computadores. Nas discussões sobre a relação entre programação e PC, alguns acreditam que os desafios de design e as ferramentas de design da computação moderna envolvem pouca codificação e, portanto, a programação não precisa ser considerada [28]. A aquisição de habilidades para a resolução de problemas requer precisão de pensamento, exposição e consideração de detalhes, e as linguagens de programação incentivam o nível de precisão necessário para desenvolver soluções propostas, executá-las, analisá-las e compará-las com outras soluções [30]. No nível superior, a programação, além de ser um tema central nas disciplinas de introdução à Ciência da Computação, também é fundamental para potencializar o desenvolvimento das habilidades do Pensamento Computacional em um nível adequado para a graduação [6]. No entanto, apesar da importância da programação, ensiná-la por si só não implica necessariamente que o PC seja potencializado. Além disso, apesar de diversas definições apontarem que PC é mais do que codificar uma solução para execução por um computador por meio de programação, poucos contestam que a programação é um veículo importante para ensinar e avaliar PC [11].

2.1 Frameworks teóricos de Pensamento Computacional

Ao longo dos anos, diversos frameworks teóricos foram propostos para o PC. Brennan and Resnick [4] descrevem um framework de PC com três dimensões principais: Conceitos, Práticas e Perspectivas. De acordo com os autores, os conceitos de PC são aqueles que os estudantes utilizam ou aprendem enquanto programam. Os conceitos mais úteis são: sequências, loops, paralelismos, eventos, condicionais, operadores e dados. As práticas são aquelas que os

estudantes desenvolvem à medida que se envolvem com os conceitos. As principais práticas computacionais são: ser incremental e iterativo, testar e depurar, reutilizar e remixar, e abstrair e modularizar. As perspectivas são criadas pelos estudantes sobre o mundo e sobre si mesmos à medida em que se envolvem com os conceitos e práticas. O framework considera como perspectivas: expressão (autoexpressão), conexão (interação com outras pessoas) e questionamento.

As habilidades de PC e programação se relacionam com o domínio cognitivo da taxonomia de Bloom [27]. A ordem em que as habilidades de programação são ensinadas remete diretamente à ordem dos níveis no domínio cognitivo da taxonomia de Bloom. Sua pesquisa sugere uma sequência comum ao ensino de programação, independente de contexto, linguagem de programação ou ambiente. Esta sequência é composta por: 1) construções, fatos, tipos; 2) como funcionam os construtos individuais; 3) usar construções de programação em contextos artificiais; 4) discriminar, decompor, abstrair; 5) criar programas, design de algoritmos; 6) testar, avaliar. O modelo também sugere os níveis de ensino de programação de acordo com a Taxonomia de Bloom, onde avaliação equivale a testar e avaliar; síntese equivale a criar programas e design de algoritmos; análise e aplicação equivalem a resumir, decompor e discriminar; compreensão e conhecimento equivalem a abstrair, decompor e discriminar.

Alguns frameworks de PC surgem através de revisões sistemáticas de literatura, considerando os termos mais empregados pelos pesquisadores. Selby and Woollard [25] propõem que PC é uma abordagem de solução de problemas que envolve as seguintes habilidades cognitivas: abstração, pensamento algorítmico, decomposição, avaliação e generalização. Este trabalho relaciona PC, principalmente, com práticas de programação, mas o framework apresentado também pode ser utilizado em abordagens que não envolvem programação. Kalelioglu et al. [14] propõem um framework baseado em uma revisão sistemática da área que aponta quinze habilidades mais utilizadas nas definições de PC. Dentre as habilidades listadas, cinco se destacam como habilidades principais: abstração, pensamento algorítmico, resolução de problemas, reconhecimento de padrões, pensamento baseado em design. Gouws et al. [9] examinaram definições de PC na literatura e agruparam as habilidades e práticas em seis classes distintas: processos e transformações, modelos e abstrações, padrões e algoritmos, ferramentas e recursos, inferência e lógica, e avaliações e melhorias.

Para Grover and Pea [11], PC é uma habilidade essencial do século XXI que ajuda os estudantes a entender e tirar proveito da computação em vários domínios. Os autores apresentam um framework que considera conceitos e práticas de PC. Os conceitos de PC incluem: lógica e pensamento lógico, algoritmos e pensamento algorítmico, padrões e reconhecimento de padrões, abstração e generalização, avaliação e, por fim, automação. As práticas de PC incluem: decomposição do problema, criação de artefatos computacionais, teste e depuração, desenvolvimento incremental e colaboração e criatividade. As práticas de colaboração e criatividade fazem parte das habilidades mais amplas do século XXI.

Araujo et al. [2] modelo de PC que engloba habilidades e competências, desenvolvido através de uma combinação de dados empíricos, análise do estado da arte no PC e elementos de avaliação cognitiva. Neste modelo, PC é uma abordagem para resolver problemas e

Tabela 1: Conceitos, práticas e perspectivas de PC.

Conceitos	Práticas	Perspectivas
Pensamento Lógico	Iterar	Expressão:
Sequências	Abstrair	E1: Eu posso criar artefatos computacionais.
Repetições	Decompor	E2: Eu posso expressar minhas ideias através da computação.
Condicionais	Modularizar	Conexão:
Operadores	Generalizar	C1: Eu posso criar com outras pessoas.
Dados	Testar e depurar	C2: Eu posso criar para outras pessoas.
Eventos		Questionamento:
Paralelismos		Q1: Eu posso questionar para entender artefatos e soluções computacionais.
Funções		Q2: Eu posso alterar artefatos e soluções computacionais existentes.

envolve quatro competências e doze habilidades. A primeira competência, intitulada “compreendendo a competência do problema”, tem como habilidades associadas: destacar os dados essenciais, reconhecer subproblemas e pensar como partes componentes. A segunda competência, intitulada “analisando a competência dos dados”, tem como habilidades associadas: compreender padrões de localização, identificar semelhanças/conexões e avaliação. A terceira competência, intitulada “pensando em competência de instruções”, inclui agir com base nos requisitos ou fazer um julgamento sobre uma situação depois de considerar as informações disponíveis. Esta competência tem como habilidades associadas: abranger a execução de um algoritmo, obedecer às regras e adequar para uma situação específica. A quarta categoria, “sistematizar os dados durante o processo de resolução”, tem como habilidades associadas: escolher uma representação de uma solução, usar uma solução anterior e reconhecer uma solução.

3 METODOLOGIA

Para o desenvolvimento deste trabalho, três etapas foram essenciais: (1) organização da definição operacional inicial, (2) realização do painel de especialistas e (3) compilar, resumir e relatar os dados. Os itens a seguir descrevem cada um desses estágios.

3.1 Definição operacional inicial

Em nosso trabalho anterior, apresentamos uma definição operacional para PC baseada na literatura [21]. Nossa abordagem considera que PC é uma abordagem para a resolução de problemas que envolve habilidades separadas entre três dimensões-chave: conceitos, práticas e perspectivas. Assim como no framework proposto por Brennan and Resnick [4], as práticas de PC se desenvolvem à medida em que os estudantes se envolvem com os conceitos, e a dimensão de perspectivas envolve habilidades relacionadas à visão do estudante sobre a computação e suas possibilidades.

Como definição operacional inicial, utilizamos o framework apresentado previamente, disponível na Tabela 1. Nossa concepção considera a programação de computadores como um fator que potencializa a aquisição de PC. Estabelecemos uma definição geral para PC e as três dimensões. Também definimos cada um dos conceitos, práticas e perspectivas listados.

3.2 Painel de especialistas

O painel de especialistas é normalmente empregado no processo de validação de conteúdo durante a elaboração de instrumentos de avaliação. Nesse processo, os especialistas ou juízes avaliam se os itens do instrumento são inteligíveis e se constituem uma representação adequada dos fatores que devem ser medidos[17].

Neste trabalho, fazemos uso do painel de especialistas para validar a definição operacional de PC. Adaptamos as diretrizes apresentadas em Grant and Davis [10] para a validação de conteúdo conforme a natureza de nosso trabalho:

- Elaboramos uma lista de possíveis participantes, com base numa tabela prévia que lista pesquisadores da área de educação em computação;
- Elaboramos uma carta-convite, enviada por e-mail, explicando a natureza da pesquisa e solicitando manifestação de interesse em participar da pesquisa;
- Elaboramos um formulário online para coleta de informações sobre o perfil dos participantes e avaliação da definição operacional de PC.

3.2.1 Participantes. Convidamos 22 pesquisadores com experiência na área de educação em computação. Um total de dez especialistas aceitaram o convite no prazo estabelecido e receberam o link para preenchimento do formulário. Consideramos este número de participante satisfatório, uma vez que abordagens de painéis de especialistas apontam que o time de especialistas deve ter entre 3 e 10 membros [18]. Dentre os participantes, todos professores doutores, 70% tinham entre cinco e nove anos de experiência em pesquisas na área de Educação em Computação, enquanto 30% tinham 10 anos ou mais de experiência na área. Todos os participantes afirmaram ter muita proximidade com o tema Pensamento Computacional. Em relação ao tema Avaliação de PC, 50% dos participantes afirmaram ter muita proximidade.

3.2.2 Coleta de dados. O questionário de avaliação foi elaborado através da ferramenta online Google Forms¹. O conceito geral de PC foi apresentado em uma seção intitulada “O que é Pensamento Computacional” e os participantes avaliaram quantitativamente, em uma escala de um a seis, o quanto concordam ou discordam com a definição apresentada. Um espaço aberto para comentários e sugestões também foi apresentado. Para cada conceito, prática e perspectiva listado na Tabela 1, apresentamos a definição, e solicitamos que o especialista marcasse, em uma escala de um a seis, o grau de concordância quanto à inclusão desse fator como um fator de PC. Também solicitamos aos especialistas que marcassem, numa escala de um a seis, o grau de concordância em relação à definição apresentada. Os especialistas também podiam oferecer feedback e sugestões em

¹<https://www.google.com/forms/about/>

espaço aberto. Para fins de replicação, disponibilizamos uma cópia do formulário com o questionário².

3.3 Análise de dados

Neste trabalho, lidamos com um conjunto de dados qualiquantitativos. Os dados quantitativos foram tabulados e geramos um escore através da média dos valores. Como a escala utilizada nas questões quantitativas foi de um a seis, consideramos um escore positivo quando está acima de 3,5. Cada conceito, prática e perspectiva descrita possui dois escores associados: um escore refere-se à concordância dos especialistas quanto àquele item pertencer ao PC, e o segundo escore refere-se à concordância dos especialistas quanto à definição apresentada. Os dados qualitativos referem-se ao feedback aberto dos especialistas em relação às definições apresentadas.

Os dados qualitativos foram submetidos à análise de conteúdo, por meio do software NVivo³. Optamos por uma abordagem inicial exploratória, com codificação *in vivo* para os trechos mais expressivos que enfatizam a opinião dos especialistas. Em seguida, agrupamos esses códigos *in vivo* de acordo com a similaridade das opiniões. Posteriormente, agrupamos os códigos em categorias com a definição geral de PC, os nomes dos conceitos, práticas e perspectivas. Para cada categoria, geramos um memorando com a definição inicial apresentada, os escores gerados através da avaliação quantitativa, a análise qualitativa e as alterações na definição com base na triangulação dos dados. No texto deste artigo, extratos explícitos dos juizes/especialistas são apresentados como um código com a letra J seguida de um número sequencial do juiz.

4 RESULTADOS E DISCUSSÃO

Nesta seção apresentamos os resultados e discussões, agrupados de acordo com a definição geral de PC e cada uma das dimensões de conceitos, práticas e perspectivas. Para cada tópico, exibiremos a definição apresentada aos especialistas, seguida da análise dos resultados do painel.

4.1 Pensamento Computacional

Definição apresentada: *Pensamento Computacional (PC) é uma abordagem para a resolução de problemas e que envolve diversas habilidades que podem ser separadas entre três dimensões-chave: conceitos, práticas e perspectivas. Os conceitos de PC são habilidades relacionadas à programação de computadores. As práticas de PC são habilidades relacionadas às estratégias que os estudantes empregam enquanto criam seus programas, ou seja, enquanto manipulam os conceitos de PC. As perspectivas de PC envolvem habilidades relacionadas à visão do estudante sobre a computação e suas possibilidades. PC está intimamente relacionado à ciência da computação de modo que, ao utilizar o termo resolução de problemas, incluímos a formulação de problemas e expressão de soluções que possam ser computadas. Nesse contexto, o PC pode ser usado para projetar sistemas, criar novos conhecimentos e melhorar a compreensão do poder e das limitações da computação na era moderna. Além disso, em nossa concepção, a programação de computadores é um elemento fundamental para o desenvolvimento do PC.*

²<https://forms.gle/6muFZrnqkqjHenea7>

³www.qsrinternational.com/nvivo-qualitative-data-analysis-software/

Nossa definição inicial apresenta PC de maneira relacionada à programação de computadores. Nossa escolha por estabelecer uma relação estreita entre o ensino de programação como maneira de fomentar o desenvolvimento de PC se dá considerando o objetivo geral da pesquisa, que é a avaliação da aquisição de PC.

A avaliação quantitativa da definição apresentada para PC teve uma média de $3,9 \pm 1,3$. A maioria dos especialistas apontaram que é importante especificar que PC é algo que antecede a programação de computadores, sendo muito mais abrangente do que a própria programação, de modo que é possível fomentá-lo sem fazer uso dela. Alguns especialistas também apontaram a necessidade de tornar mais clara nossa decisão, especificando que a programação desempenha um papel importante na promoção de PC, uma vez que permite alavancar diversas habilidades de PC.

Um dos especialistas questionou a relação entre PC e Computação. Para ele, o conceito de PC pode ser estendido para diversos outros campos, de modo que o termo Pensamento Computacional possa até ser substituído pelo termo “Pensamento Humano”. Segundo ele, desenvolver artefatos computacionais não é uma obrigação para o exercício de PC. Essa visão parece ser isolada, uma vez que nenhum dos outros especialistas questionou a relação explícita entre PC e computação. No entanto, esta observação realçou a necessidade de aprimorar nossa definição, especificando que “soluções que possam ser computadas” não implica necessariamente na criação de artefatos computacionais. Por outro lado, para nosso objetivo de avaliação, a existência e criação destes são essenciais para avaliar o desenvolvimento do PC.

Alguns especialistas questionaram a divisão entre as dimensões de conceitos, práticas e perspectivas:

Não considero as práticas e perspectivas parte da definição do Pensamento Computacional. (J02)

Conceitos e práticas se misturam às vezes como, por exemplo, abstrair é um conceito e uma prática também. (...) entendo que as perspectivas têm mais relação [com] a forma do estudante aprender e se relacionar com o conhecimento. É mais do estudante do que do PC. (J08)

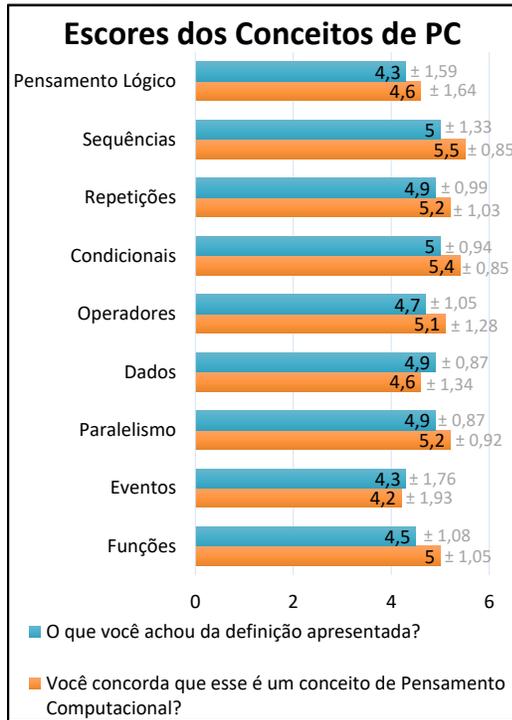
Essas observações apontam mais uma vez a necessidade de explicitar nossas motivações. Nos inspiramos diretamente na visão de Brennan and Resnick [4] que descrevem um framework operacional para PC a partir das experiências de aprendizagem com o ambiente Scratch. Por exemplo, classificamos abstração como uma prática pois, para abstrair num contexto de programação, é necessário reconhecer que determinadas especificações do problema são importantes para a construção da solução. Ou seja, resultam na aplicação dos conceitos de programação. Portanto, em nossa visão, abstrair é uma prática. Apesar dessas observações, os especialistas parecem ter entendido que conceitos, práticas e perspectivas fazem sentido ao considerar o contexto educacional:

Essa definição está vinculada ao contexto educacional, portanto, práticas e perspectivas são fatores importantes. (J06)

4.2 Conceitos

A Figura 1 apresenta os escores alcançados por cada conceito de PC apresentado aos especialistas. Pensamento Lógico, Dados e Eventos

Figura 1: Escores quantitativos para os Conceitos de PC.



obtiveram os menores escores. A seguir, apresentamos e discutimos individualmente os resultados para cada conceito.

4.2.1 Pensamento Lógico. Definição apresentada: *O pensamento lógico envolve analisar situações para tomar uma decisão ou chegar a uma conclusão sobre uma situação. Como parte do desenvolvimento de habilidades de PC, os estudantes devem desenvolver habilidades de pensamento lógico trabalhando expressões booleanas com operações como AND, OR e NOT.*

Apesar de este conceito ter sido avaliado quantitativamente com um escore acima da média, a maioria dos juizes apontaram que a definição apresentada é restrita, uma vez que pensamento lógico pode ser fomentado de diversas maneiras. Apesar de expressarem que a definição de Pensamento lógico é restrita, nenhum deles apresentou uma sugestão alternativa para ampliar este conceito.

Empregamos este conceito considerando o framework de Grover and Pea [11] para PC. Uma série de experiências pode envolver o estudante na aquisição do pensamento lógico. Uma delas é através da lógica booleana formal, pela compreensão das operações booleanas e construção de expressões utilizando estas operações [11]. Esta concepção pareceu-nos bastante adequada, já que optamos por uma abordagem atrelada à programação de computadores para a aquisição de PC.

Analisando as observações dos especialistas e revisando a própria literatura, decidimos retirar este conceito de nosso framework por acreditar que, indiretamente, o Pensamento Lógico como um conceito mais amplo é contemplado à medida em que os estudantes são expostos a situações-problema, incluindo aquelas voltadas para as práticas de PC.

4.2.2 Sequências. Definição apresentada: *O conceito de sequências consiste na especificação de uma sequência de etapas ou instruções individuais que podem ser executadas de modo a produzir uma ação.*

Tanto os dados quantitativos quanto o feedback dos especialistas apoiam a permanência de sequências como um conceito de PC. As recomendações deles questionaram mais a forma: trocar o termo “sequência de etapas ou instruções” por “série/ordem de instruções”; especificar que uma sequência reproduz uma ou mais ações.

4.2.3 Repetições. Definição apresentada: *Repetições são estruturas empregadas para executar a mesma sequência várias vezes.*

As avaliações quantitativa e qualitativa apoiam a permanência de Repetições como um conceito de PC. As alterações sugeridas pela maioria dos especialistas relacionam-se com a forma do texto. Eles sugeriram melhorar a especificação sobre quantas vezes uma sequência é executada dentro de um loop:

Para executar a mesma sequência uma certa quantidade de vezes. (J01)

Uma estrutura de repetição pode executar apenas uma vez ou não executar sequer uma vez. (J02)

Blocos de instruções podem ser repetidos enquanto uma condição for verdadeira. (J10)

Optamos por combinar as diferentes sugestões dos especialistas.

4.2.4 Condicionais. Definição apresentada: *Condicionais são estruturas que permitem a execução de diferentes instruções a depender de determinadas circunstâncias. As condicionais oferecem aos programadores a capacidade de tomar decisões com base em determinadas condições, e apoiam a expressão de vários resultados.*

O conceito de Condicionais obteve boa avaliação quantitativa, tanto sobre ser considerado um conceito de PC quanto sobre a definição apresentada. Alguns especialistas apresentaram insatisfação com a parte final da definição: “apoia a expressão de vários resultados”. Também foi questionado o uso da palavra “programadores”, que pode limitar a aplicação das condicionais. Assim, alteramos a definição, aplicando as alterações sugeridas.

4.2.5 Operadores. Definição apresentada: *Os operadores fornecem suporte para expressões matemáticas e lógicas, permitindo que o programador execute manipulações.*

Tanto a avaliação quantitativa quanto a qualitativa corroboram para a manutenção de Operadores como conceito de PC. Nas avaliações qualitativas, alguns especialistas sugeriram substituir o termo “programadores” e acrescentar informação sobre que tipos de manipulação são realizados. Não consideramos o uso do termo “programadores” inadequado, uma vez que nossa perspectiva, na definição de PC apresentada, considera a programação como um meio importante de fomento para a aquisição do PC e os conceitos de PC também são conceitos de programação. Em relação à especificação das manipulações realizadas por meio do uso dos operadores, o comentário de um dos especialistas oferece uma síntese de todas as sugestões de mudança. Esta sugestão foi incorporada à definição:

Manipulações aritméticas, lógicas ou relacionais – talvez acrescentar esta informação. (J05)

4.2.6 Dados. Definição apresentada: *O conceito de dados relaciona-se com a habilidade de armazenar, acessar e atualizar valores.*

Os especialistas apontaram que a definição apresentada para o conceito de Dados foi reducionista, no sentido de que descreve apenas algumas manipulações possíveis para dados.

Acredito que “Manipulação de Dados” no lugar de “Dados” seja melhor como conceito. (J05)

De modo geral, os juízes sugeriram a adição de outros tipos de manipulações que podem ser feitas com dados, como exibir e excluir. Assim, optamos por manter a nomenclatura Dados, mas especificar que nossa definição tem foco na manipulação de dados.

4.2.7 Paralelismos. Definição apresentada: *O paralelismo torna possível que sequências diferentes sejam executadas ao mesmo tempo.*

Para este conceito os especialistas concordaram com a inclusão como conceito de PC, mas sugeriram diversas alterações na forma do texto, as quais consideramos para aperfeiçoar a definição. Selecionamos três extratos que sintetizam as sugestões:

Incluir que as sequências que serão executadas não são dependentes entre si (J01)

O paralelismo torna possível que sequências diferentes sejam executadas ao mesmo tempo e simultaneamente. (J02)

Da forma como está, pode ter uma interpretação ambígua, como sendo sequências que obrigatoriamente são diferentes uma da outra. (J04)

4.2.8 Eventos. Definição apresentada: *Eventos são gatilhos que disparam ações. Um evento é “uma coisa que causa outra coisa acontecer”.*

Apesar de alcançar um escore médio acima da neutralidade (3.5), ao questionar se Eventos seriam um conceito de PC, na avaliação qualitativa, alguns especialistas fizeram observações sobre este conceito ter uso limitado:

Evento faz sentido apenas em IDEs de alto nível como o Scratch (...). Não é um conceito fundamental do PC, é uma forma de organizar um programa. (J08)

A minha dúvida foi em relação à possibilidade de considerar “Eventos” como conceito do PC, pois ele é um conceito encontrado em alguns ambientes de programação, como o Scratch, por exemplo, mas não é algo que se possa generalizar ou ser usado em outros ambientes/espacos. (J05)

Incluimos o conceito Eventos pois eles são um componente essencial da mídia interativa [4]. No entanto, é preciso reconhecer a relevância da limitação apontada pelos juízes, de que sua utilização na aprendizagem de programação em níveis mais básicos ocorre principalmente no Scratch, mas não é algo geral. É possível imaginar uma diversidade de utilizações dos eventos no Scratch e, apenas no contexto do Scratch, associar o emprego dos eventos à práticas de PC como abstração e modularização. Por essa razão, decidimos retirar o conceito de Eventos de nossa definição operacional.

4.2.9 Funções. Definição apresentada: *Funções são seções de programas que executam tarefas específicas. Funções podem ser simples ou ter parâmetros.*

Na avaliação qualitativa, alguns juízes sugeriram que funções não são elementos que definem PC. Apesar destas opiniões, optamos por manter o conceito de funções pois ele fomenta práticas

como decomposição e modularização em diferentes linguagens e situações. Neste sentido, a opinião dos juízes foi empregada para aprimorar a definição apresentada. Uma observação em específico concentrou as alterações apontadas:

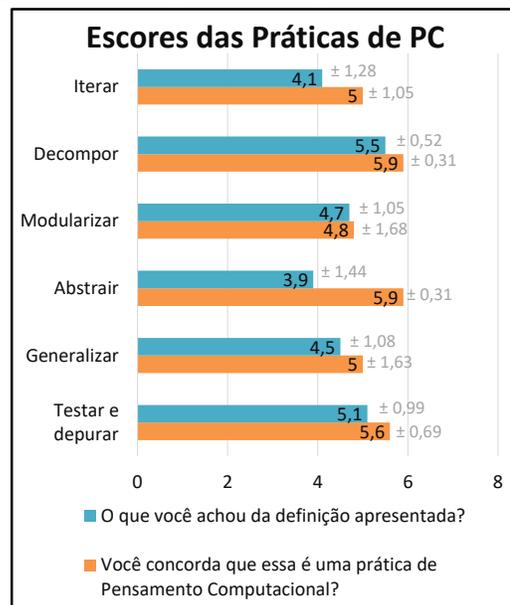
Funções são seções de uma sequência que executam tarefas específicas. Elas podem ser simples ou complexas, consumindo eventualmente dados de entrada (parâmetros) e gerando dados de saída (resultado). (J04)

Desse modo, alteramos a definição conforme esta sugestão.

4.3 Práticas

A Figura 2 apresenta os escores médios de cada prática de PC segundo a avaliação dos especialistas.

Figura 2: Escores quantitativos para as Práticas de PC.



4.3.1 Iterar. Definição apresentada: *Prática que consiste em aumentar a solução ou programa iterativamente, com testes frequentes para desenvolver melhorias. Desta forma, a iteração é um processo adaptativo, no qual os estudantes adquirem uma abordagem de solução em pequenas etapas.*

Tanto o escore quantitativo quanto o feedback qualitativo dos especialistas apontam que Iterar é uma prática de PC. Assim, focamos nossa análise em melhorar a definição da prática Iterar. As observações dos especialistas se referem, em quase sua totalidade, a sugestões de alteração na forma. O uso do termo “aumentar a solução” foi criticado por cinco especialistas, que sugeriram sua alteração para termos como “evoluir a solução”, “Desenvolver em partes”, “aperfeiçoar” e “melhorar”. O uso da palavra “iterativamente” também foi questionado:

Foi usada uma palavra de mesmo radical (iterativamente) para explicar o que é iterar. (J10)

A prática de Iterar em nossa definição equivale a prática *Being incremental and iterative* descrita no framework de Brennan and Resnick [4] e a prática de *Incremental development* descrita no framework de Grover and Pea [11]. A definição apresentada tem estes trabalhos como fonte. No entanto, ao apresentar as definições aos especialistas optamos por omitir as referências a fim de evitar vieses de julgamento.

4.3.2 Decompor. Definição apresentada: *Prática que consiste em dividir o problema em subproblemas menores, tornando-o mais tratável.*

A prática de decomposição recebeu escores altos. Além disso, poucas observações foram feitas por parte dos especialistas. Esperávamos que essa prática recebesse boa avaliação, uma vez que ela parece ser consenso entre diversos trabalhos de fomento e avaliação da aquisição do PC [8, 16, 19, 20].

Três especialistas sugeriram a substituição do termo “tratável”. Um dos especialistas sugeriu uma alteração explícita, e nós resolvemos acatá-la:

Em vez de tratável, eu usaria “de mais simples resolução”. (J10)

4.3.3 Modularizar. Definição apresentada: *Prática de desenvolver processos que encapsulam um conjunto de comandos frequentemente usados para a execução de uma função específica.*

Ambos os escores para essa prática estão acima da média de 3,5, mas recebemos pouco feedback dos especialistas sobre essa prática. Um dos especialistas avaliou a definição apresentada como inconsistente:

Muitas palavras que são vagas e não mantém a consistência com o termo sequencias usado antes, como processos, encapsulam, comandos. (J08)

Concordamos, em parte, com essa visão. Incluímos modularizar como uma prática, porque ela aparece no framework de Brennan and Resnick [4] como uma prática associada à abstração, e é caracterizada como “construir algo grande, colocando juntos coleções de peças menores”. Apesar de encontrar trabalhos de avaliação do PC que avaliam modularização, a utilização da prática de modularizar não é algo que aparece expressivamente na literatura. No trabalho de Brennan and Resnick [4], esta prática é exemplificada no contexto do Scratch, quando um programador separa os diferentes comportamentos ou ações de um ator em diferentes pilhas de código, usando diretamente a noção de eventos. Uma vez que já discutimos o uso de eventos e seu uso limitado a organização de programas no Scratch, optamos por remover a prática de Modularização.

4.3.4 Abstrair. Definição apresentada: *Prática relacionada à capacidade de esconder complexidade, negligenciando detalhes desnecessários.*

Essa prática obteve um escore alto quanto a ser uma prática de PC, contrastando com o escore que representa a avaliação da definição apresentada. Corroborando, o feedback fornecido pelos especialistas foi inteiramente voltado para a melhoria da definição. Quatro especialistas reclamaram do uso da palavra “negligenciando” e três especialistas reclamaram do uso do termo “esconder complexidade”. Tentamos oferecer uma definição simplificada para a prática de abstração, assim como Grover and Pea [11], que descrevem abstração como uma maneira de “ocultação de informações” para simplificar e

gerenciar a complexidade. Em face ao feedback recebido, reconhecemos que utilizamos termos demasiadamente informais, descaracterizando a definição da prática. Promovemos uma mudança baseada na própria sugestão de reescrita oferecida por um dos especialistas:

Abstração não é “esconder” complexidade, e sim ter a habilidade de selecionar, na representação do problema, as informações que são relevantes naquele contexto específico. (J10)

4.3.5 Generalizar. Definição apresentada: *Prática de reconhecer “padrões comuns”. É a capacidade de passar de uma aplicabilidade específica para uma mais ampla.*

Os escores quantitativos resultantes sugerem que Generalizar é uma prática de PC. Metade dos especialistas sugeriram algum tipo de ênfase na ideia de reconhecimento de padrões. Generalização é uma das habilidades de PC onde a falta de consenso sobre uma definição específica fica em evidência. O framework de Brennan and Resnick [4] não faz menção direta ao termo generalização. Incorporamos Generalização pois esta é uma habilidade frequentemente associada ao PC, seja como prática ou perspectiva [7]. O trabalho de Grover and Pea [11], por exemplo, considera padrões e reconhecimento de padrões como um conceito e abstração e generalização, juntos, como outro conceito de PC, embora os autores reconheçam que o reconhecimento de padrões pode levar à definição de soluções generalizáveis. Para Selby [26], uma de nossas principais fontes, Generalização é a habilidade de passar de uma aplicabilidade mais específica para uma mais ampla. Neste contexto, generalização associa-se intimamente à capacidade de reconhecer padrões para aplicar soluções que já foram utilizadas em situações prévias, ou aplicar a solução atual em futuros problemas de natureza similar. Os especialistas demonstraram compartilhar, mesmo que em parte, da nossa concepção. Por isso, alteramos a definição de generalização, apenas com a finalidade de tornar mais clara esta relação.

4.3.6 Testar e depurar. Definição apresentada: *Prática de identificação e correção de erros nos projetos. Testar e depurar faz parte do processo de avaliação de uma solução.*

Essa prática recebeu escores acima da média, indicando que é uma prática de PC e que a definição apresentada está em concordância com os especialistas. Um dos especialistas sugeriu alteração na forma e nós acatamos:

Os erros não existem nos “projetos”, mas “nas soluções desenvolvidas”. (J04)

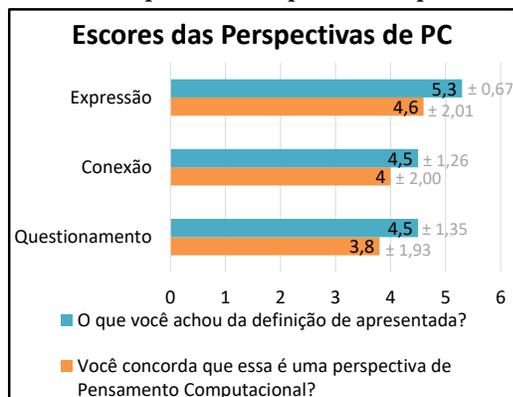
De maneira geral, a concordância dos especialistas em relação a essa prática já era esperada. Testar e depurar é uma prática fundamental para lidar com problemas que aparecem durante o desenvolvimento de uma solução e, ao final, para avaliar sua eficácia em resolver o problema [4]. Essa prática permite avaliar a acurácia de uma solução computacional e se ela satisfaz regras estabelecidas, se funciona para as condições de contorno e todas as entradas e situações relevantes, e se atua conforme esperado para entradas [11]. Procuramos apresentar uma definição que simplifica essas visões sobre o processo de teste e depuração.

4.4 Perspectivas

As perspectivas de Expressão, Conexão e Questionamento obtiveram escores mais baixos se comparados aos itens das dimensões de

Conceitos e Práticas (ver Figura 3). Considerando o desvio padrão desses escores, e comparando com o feedback oferecido, percebemos que não há consenso entre os especialistas quanto à inclusão da dimensão Perspectivas em um framework de PC. No entanto, três especialistas indicaram alta discordância em todas as Perspectivas apresentadas, enquanto os outros sete concordaram expressamente com a inclusão.

Figura 3: Escores quantitativos para as Perspectivas de PC.



O feedback contrário considera que estas perspectivas não são exclusivas de PC ou são consequência do processo de aprender a resolver problemas:

Entendo que é uma perspectiva que não se relaciona exclusivamente com o PC e por isso não faz sentido estar como perspectiva do PC (J08)

A dimensão Perspectiva foi incluída por Brennan and Resnick [4] em seu framework, ao observarem a evolução dos estudantes que trabalham com Scratch em relação à compreensão de si mesmos, de suas relações com os outros e do mundo tecnológico ao seu redor. Para os autores, esta nova percepção não é capturada pelo enquadramento de conceitos e práticas. Por isso, configuram uma dimensão específica de PC. Neste sentido, as perspectivas de PC são uma espécie de passo final, uma consequência do PC e, embora não sejam exclusivas dele, são por ele fomentadas. Apesar de esta dimensão ser pouco explorada na literatura, muitas outras avaliações precisam abordar as perspectivas de PC e examinar sua relação com os resultados de aprendizagem [7]. Por isso, optamos por manter as três perspectivas e promover adaptações das definições conforme o feedback dos especialistas.

4.4.1 Expressão. Definição apresentada: *Com as perspectivas de expressão, o estudante entende que não é apenas um consumidor de tecnologia e sente-se capaz de expressar-se de maneira ativa com a Computação. O estudante enxerga a criação de artefatos computacionais como uma forma de expressão criativa. São variáveis da perspectiva Expressão: (E1) Eu posso criar artefatos computacionais. (E2) Eu posso expressar minhas ideias por meio da computação.*

A ideia central da perspectiva de expressão é que um “pensador computacional” vê a computação como mais do que algo para consumir. Computação é algo que eles podem usar para design e autoexpressão [4]. Neste contexto, as variáveis E1 e E2 expressam

a visão de sentir-se capaz de criar e explorar a criatividade através da computação. Recebemos pouco feedback sugerindo alterações nestas variáveis. Um dos especialistas sugeriu a troca do termos “artefatos computacionais” e “computação” por tecnologias digitais. Empregamos o termo artefato computacional, no sentido de que ele pode ser qualquer produto feito utilizando a computação, como algoritmos, programas, sites, circuitos e dispositivos, dentre outros. Por isso, não alteramos a definição geral da perspectiva de expressão, nem as variáveis E1 e E2.

4.4.2 Conexão. Definição apresentada: *Com as perspectivas de conexão, o estudante entende a comunicação e colaboração com os outros como elementos atrelados à Computação. As perspectivas de conexão envolvem o valor da criação com os outros e o valor da criação para os outros. Os estudantes sentem que são capazes de fazer mais do que poderiam fazer por conta própria ao colaborar. Os estudantes sentem que há valor em criar para os outros e ter suas criações entretendo, envolvendo, ajudando ou educando outras pessoas. São variáveis da perspectiva Conexão: (C1) Eu posso criar com outras pessoas. (C2) Eu posso criar para outras pessoas.*

Os especialistas questionam a inclusão dessa perspectiva, apresentando diferentes justificativas:

Vejo isso mais como a nona competência geral da BNCC: “Empatia e Cooperação” e não como uma relação com o PC. (J02)

Muitos aprendem computação sem colaborar. (J08)

Eu não veria necessariamente a colaboração como um ponto forte do PC, se pensássemos no PC como uma abordagem focada somente na programação de computadores. Se pensarmos no PC como uma abordagem bem mais abrangente, que visa à formação de resolvidores de problemas (*problem solvers*), aí, sim, a colaboração entra com toda a força. (J04)

Em face a esse feedback, é necessário especificarmos que as definições empregadas estão atreladas à programação com o objetivo de promover um framework operacional que possa ser empregado na avaliação da aquisição do PC em abordagens de ensino de computação, especialmente as de longo termo, como é o caso de currículos de computação nas escolas. Por isso, defendemos que a concepção geral sobre PC é abrangente. Isso justifica a inclusão da perspectiva de conexão, uma vez que o trabalho com os outros e para os outros é frequentemente exercitado na aprendizagem da computação.

Feedback adicional oferecido trata da forma das definições de Conexão e das variáveis C1 e C2. Estes foram acatados nas alterações que realizamos.

4.4.3 Questionamento. Definição apresentada: *Com a perspectiva de questionamento, o estudante estabelece conexão entre suas habilidades com a Computação e a capacidade de compreender e aperfeiçoar o mundo tecnológico. Os estudantes não sentem desconexão entre as tecnologias que os cercam e suas habilidades para negociar realidades do mundo tecnológico. São variáveis da perspectiva Questionamento: (Q1) Eu posso questionar para entender artefatos e soluções computacionais. (Q2) Eu posso alterar artefatos e soluções computacionais já existentes.*

Questionamento obteve o menor escore dentre as perspectivas apresentadas. O feedback que questiona a inclusão dessa perspectiva é similar ao feedback de mesma natureza para Expressão e Conexão, apontando que Questionamento não é algo exclusivo de PC. Já consideramos estas observações no início dessa seção, mas reforçamos nossa ideia sobre a perspectiva, trazendo a visão de Brennan and Resnick [4] sobre como essa perspectiva pode ser observada:

O questionamento envolve interrogar o que é dado como certo e, em alguns casos, responder a esse interrogatório por meio do design. Por exemplo, o ambiente de programação Scratch é um artefato computacional que tem certas possibilidades e limitações de design. Alguns jovens membros da comunidade questionaram essas limitações e se uniram para fazer uma versão derivada do Scratch que incluía blocos que eles achavam que deveriam ser incluídos e desenvolveram um site onde outras pessoas poderiam baixar sua versão modificada do Scratch. Isso envolveu não apenas o reconhecimento de que Scratch é um artefato projetado no mundo que pode ser modificado, mas também que eles, como designers de mídia computacional, tinham poderes para modificá-lo. (p11.)

Um dos especialistas sugere a mudança de nome da perspectiva:

Questionamento dá a ideia de dúvidas, indagações e não de certezas. Talvez repensar o nome Questionamento para outro mais sugestivo. (J05)

Mesmo considerando válida essa sugestão, não alteramos a nomenclatura, por esta ser considerada em diversos trabalhos da área [1, 7]. As alterações sugeridas para a forma tratam, em sua maioria, dos termos “não sentem desconexão” empregados na definição geral de questionamento. Alteramos a definição empregando as sugestões apresentadas.

5 DEFINIÇÃO OPERACIONAL DE PENSAMENTO COMPUTACIONAL

Nesta seção, apresentamos a definição operacional de PC após o processo de análise do feedback oferecido pelos especialistas participantes do painel. Vale lembrar que as definições apresentadas tiveram como base principal os trabalhos de Brennan and Resnick [4], Grover and Pea [11], Selby [26] e Seehorn et al. [24]. A Tabela 2 sintetiza os Conceitos, Práticas e Perspectivas de PC.

5.1 Pensamento Computacional

Pensamento Computacional (PC) é uma abordagem para a resolução de problemas que envolve diversas habilidades, como abstração, decomposição, reconhecimento de padrões, dentre outras. PC está intimamente relacionado à ciência da computação de modo que, ao utilizar o termo resolução de problemas, incluímos a formulação de problemas e expressão de soluções que possam ser computadas.

Diversas abordagens podem ser empregadas para fomentar a aquisição de PC, envolvendo ou não a programação de computadores. Em nossa concepção, a programação de computadores e a criação de artefatos computacionais são elementos fundamentais para o desenvolvimento das habilidades de PC. Nesse contexto, PC

pode ser usado para projetar sistemas, criar novos conhecimentos e melhorar a compreensão do poder e das limitações da computação na era contemporânea. Por isso, em nossa definição operacional, classificamos as habilidades de PC em três dimensões-chave: conceitos, práticas e perspectivas. Os conceitos de PC estão relacionados à programação de computadores. As práticas de PC são habilidades relacionadas às estratégias que os estudantes empregam enquanto criam seus programas, ou seja, enquanto manipulam os conceitos de PC. As perspectivas de PC envolvem habilidades relacionadas à visão do estudante sobre a computação e suas possibilidades.

Tabela 2: Conceitos, Práticas e Perspectivas de PC.

Conceitos	Práticas	Perspectivas
Sequências Repetições	Iterar Abstrair	Expressão: E1: Eu posso criar artefatos computacionais. E2: Eu posso expressar minhas ideias através da computação.
Condicionais	Decompor	Conexão: C1: Eu posso criar artefatos computacionais com outras pessoas. C2: Eu posso criar artefatos computacionais para outras pessoas.
Dados Operadores	Generalizar Testar e depurar	Questionamento: Q1: Eu posso questionar artefatos e soluções computacionais existentes. Q2: Eu posso alterar artefatos e soluções computacionais existentes.
Paralelismos		
Funções		

5.2 Conceitos

Sequências: Especificação de uma série de etapas ou instruções individuais que podem ser executadas de modo a reproduzir uma ou mais ações.

Repetições: Estruturas empregadas para executar a mesma sequência repetidas vezes. As repetições podem ter condições que determinem quantas vezes uma determinada sequência é repetida: nenhuma vez, uma vez, várias vezes ou ininterruptamente.

Condicionais: Estruturas que permitem a execução de diferentes instruções a depender de determinadas circunstâncias. Condicionais oferecem aos algoritmos/programas a capacidade de mudar o fluxo de execução com base em determinadas condições.

Dados: O conceito de dados relaciona-se com a habilidade de manipular dados. Consideramos as seguintes manipulações: armazenar, acessar, exibir, atualizar e excluir.

Operadores: Fornecem suporte para expressões matemáticas e lógicas, permitindo que o programador execute manipulações aritméticas, lógicas ou relacionais.

Paralelismos: Sequências de instruções acontecendo simultaneamente. Essas sequências de instruções não precisam ser diferentes, mas devem ser independentes entre si.

Funções: Seções de uma sequência que executam tarefas específicas. Elas podem ser simples ou complexas, consumindo eventualmente dados de entrada (parâmetros) e gerando dados de saída (resultado).

5.3 Práticas

Iterar: Iteração é um processo adaptativo, no qual uma solução é desenvolvida em etapas. A prática de Iterar equivale ao desenvolvimento incremental, onde uma solução ou programa é desenvolvido iterativamente, com testes e depuração frequentes para desenvolver melhorias.

Abstrair: Prática de identificar e selecionar, na representação do problema, os elementos que são relevantes para desenvolver uma solução.

Decompor: prática que consiste em dividir o problema em subproblemas menores, tornando-o de mais simples resolução.

Generalizar: prática de reconhecer padrões para identificar categorias de problemas similares e aplicar soluções que já foram utilizadas previamente.

Testar e Depurar: Prática de identificação e correção de erros nas soluções desenvolvidas. Testar e depurar faz parte do processo de avaliação de uma solução.

5.4 Perspectivas

Expressão: Com a perspectiva Expressão, o estudante entende que não é apenas um consumidor de tecnologia e sente-se capaz de expressar-se de maneira ativa com a Computação. O estudante enxerga a criação de artefatos computacionais como uma forma de expressão criativa.

São variáveis da perspectiva Expressão: (E1) Eu posso criar artefatos computacionais, e (E2) Eu posso expressar minhas ideias por meio da computação.

Conexão: Com a perspectiva Conexão, o estudante entende a comunicação e a colaboração com os outros como elementos relacionados à Computação. A perspectiva de Conexão envolve o valor da criação com os outros e o valor da criação para os outros. Os estudantes sentem-se capazes de fazer mais do que poderiam fazer por conta própria ao colaborar com os outros. O estudantes sentem que há valor em criar para os outros e ter suas criações entretendo, envolvendo, ajudando ou educando outras pessoas.

São variáveis da perspectiva Conexão: (C1) Eu posso criar artefatos computacionais com outras pessoas, e (C2) Eu posso criar artefatos computacionais para outras pessoas.

Questionamento: Com a perspectiva de Questionamento, o estudante estabelece conexão entre suas habilidades com a Computação e a capacidade de compreender e aperfeiçoar o mundo tecnológico. Os estudantes sentem que seus conhecimentos e habilidades com a Computação lhes permitem questionar soluções e artefatos computacionais já existentes e, em alguns casos, promover alterações ou novos artefatos a partir deste questionamento.

São variáveis da perspectiva Questionamento: (Q1) Eu posso questionar artefatos e soluções computacionais já existentes, e (Q2) Eu posso alterar artefatos e soluções computacionais já existentes.

5.5 Ameaças à validade

Nosso estudo está sujeito a algumas ameaças à validade. Um risco associado a este estudo é o fator humano, uma vez que possíveis vieses dos pesquisadores durante a análise qualitativa e triangulação dos dados podem interferir. A participação de apenas dez especialistas também pode ser considerado uma ameaça à validade, embora tenhamos adotado a quantidade indicada de especialistas para compor um painel. É importante ressaltar que um painel de especialistas não precisa representar estatisticamente a área, mas garantir que os participantes tenham experiência com o elemento da pesquisa. Essa metodologia, é normalmente empregada na validação de conteúdo em instrumentos de avaliação, portanto, adaptamos a metodologia para a realidade de nossa proposta. Fizemos uso de amostragem por conveniência, mas é importante salientar que foram convidados pesquisadores da área de pensamento computacional de diferentes instituições do país.

6 CONCLUSÕES

Neste trabalho, apresentamos uma definição operacional para Pensamento Computacional, onde os fatores associados à aquisição de PC estão divididos em três dimensões: Conceitos, Práticas e Perspectivas. Consideramos a programação de computadores como um importante meio para o desenvolvimento do PC e, frequentemente explorado pelas iniciativas de fomento ao PC e ensino de Computação. Assim, os conceitos de PC foram descritos em função da utilização de conceitos de programação.

Nosso trabalho baseia-se em definições apresentadas na literatura e foi submetido a um painel de especialistas composto por pesquisadores da área de Educação em Computação. O feedback fornecido foi utilizado para melhorar a definição operacional proposta e o conteúdo. Consequentemente, nosso processo de apresentação dos resultados também discute quais são os conceitos de PC, as práticas de PC, as perspectivas de PC e suas respectivas definições. Outras discussões importantes trazidas são: como qual o papel da programação de computadores ou até que ponto as habilidades que não são exclusivas de PC podem ser empregadas para representá-lo.

Por se tratar de uma definição operacional, nosso trabalho pode ser utilizado como framework no desenvolvimento de instrumentos para avaliação da aquisição de PC. A associação da aquisição de PC ao ensino-aprendizagem de programação visa fornecer uma maneira objetiva pela qual o PC possa ser medido. A importância deste trabalho reside no fato de que não há consenso sobre quais habilidades constituem PC e, isso afeta o modo como PC pode ser medido e como a eficácia de experiências distintas podem ser comparadas. Além disso, as definições apresentadas oferecem um aporte para que pesquisadores da área e professores possam elaborar experiências de ensino-aprendizagem que permitam uma avaliação posterior efetiva do desenvolvimento dos estudantes.

Como trabalhos futuros, pretendemos elaborar uma avaliação de PC a partir da definição operacional apresentada. Dois instrumentos devem ser elaborados e submetidos a um processo de validação com análise de conteúdo por parte de painel de especialistas e análises psicométricas como a Teoria da Resposta ao Item.

AGRADECIMENTOS

Agradecemos aos especialistas pela participação no painel e pelas ricas contribuições dadas.

REFERÊNCIAS

- [1] Catherine Adams, Maria Cutumisu, and Chang Lu. 2019. Measuring K-12 computational thinking concepts, practices and perspectives: An examination of current CT assessments. In *Society for Information Technology & Teacher Education International Conference*. Association for the Advancement of Computing in Education (AACE), 275–285.
- [2] Ana Liz Souto O. Araujo, Wilkerson L. Andrade, Dalton D. Serey Guerrero, and Monilly Ramos Araujo Melo. 2019. How Many Abilities Can We Measure in Computational Thinking? A Study on Bebras Challenge. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (Minneapolis, MN, USA) (*SIGCSE '19*). Association for Computing Machinery, New York, NY, USA, 545–551. <https://doi.org/10.1145/3287324.3287405>
- [3] David Barr, John Harrison, and Leslie Conery. 2011. Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology* 38, 6 (2011), 20–23.
- [4] Karen Brennan and Mitchel Resnick. 2012. New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada*. 1–25.
- [5] CIEB. 2018. Currículo de Tecnologia e Computação da Educação Infantil ao Ensino Fundamental. <http://curriculo.cieb.net.br/>
- [6] Stephen Cooper and Wanda Dann. 2015. Programming: A Key Component of Computational Thinking in CS Courses for Non-majors. *ACM Inroads* 6, 1 (Feb. 2015), 50–54. <https://doi.org/10.1145/2723169>
- [7] Maria Cutumisu, Cathy Adams, and Chang Lu. 2019. A Scoping Review of Empirical Research on Recent Computational Thinking Assessments. *Journal of Science Education and Technology* 28, 6 (2019), 651–676.
- [8] Valentina Dagiene and Gabrielė Stupurienė. 2016. Bebras - a Sustainable Community Building Model for the Concept Based Learning of Informatics and Computational Thinking. *INFORMATICS IN EDUCATION* 15 (05 2016), 25–44. <https://doi.org/10.15388/infedu.2016.02>
- [9] Lindsey Gouws, Karen Bradshaw, and Peter Wentworth. 2013. First Year Student Performance in a Test for Computational Thinking. In *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference* (East London, South Africa) (*SAICSIT '13*). Association for Computing Machinery, New York, NY, USA, 271–277. <https://doi.org/10.1145/2513456.2513484>
- [10] Joan S Grant and Linda L Davis. 1997. Selection and use of content experts for instrument development. *Research in nursing & health* 20, 3 (1997), 269–274.
- [11] Shuchi Grover and Roy Pea. 2017. *Computational Thinking: A Competency Whose Time Has Come*.
- [12] Shuchi Grover, Roy Pea, and Stephen Cooper. 2015. Designing for deeper learning in a blended computer science course for middle school students. *Computer science education* 25, 2 (2015), 199–237.
- [13] Ting-Chia Hsu, Shao-Chen Chang, and Yu-Ting Hung. 2018. How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education* 126 (2018), 296–310.
- [14] Filiz Kalelioglu, Yasemin Gulbahar, and Volkan Kukul. 2016. A Framework for Computational Thinking Based on a Systematic Research Review. *Baltic Journal of Modern Computing* 4 (05 2016), 583–596.
- [15] Sze Yee Lye and Joyce Hwee Ling Koh. 2014. Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior* 41 (2014), 51–61.
- [16] Jesús Moreno-León, Gregorio Robles, and Marcos Román-González. 2015. Dr. Scratch: Automatic Analysis of Scratch Projects to Assess and Foster Computational Thinking. *RED-Revista de Educación a Distancia* (09 2015).
- [17] Luiz Pasquali. 2011. *Psicometria: Teoria dos Testes na Psicologia e na Educacao* (5 ed.). Vozes.
- [18] Denise Polit and Cheryl Beck. 2006. The Content Validity Index: Are you sure you know what's being reported? Critique and recommendations. *Research in nursing & health* 29 (10 2006), 489–97. <https://doi.org/10.1002/nur.20147>
- [19] Wouter Rijke, Lars Bollen, Tessa Eysink, and Jos Tolboom. 2018. Computational Thinking in Primary School: An Examination of Abstraction and Decomposition in Different Age Groups. *Informatics in Education* 17 (04 2018). <https://doi.org/10.15388/infedu.2018.05>
- [20] Marcos Román-González, Juan-Carlos Pérez-González, Jesús Moreno-León, and Gregorio Robles. 2018. Can computational talent be detected? Predictive validity of the Computational Thinking Test. *International Journal of Child-Computer Interaction* (07 2018). <https://doi.org/10.1016/j.ijcci.2018.06.004>
- [21] Bianca L. Santana, Christina Chavez, and Roberto Bittencourt. 2020. Uma Proposta de Avaliação de Conceitos, Práticas e Perspectivas de Pensamento Computacional. In *CTRL+E 2020 - V Congresso sobre Tecnologias na Educação*. <https://doi.org/10.5753/ctrl.2020.11393>
- [22] Priscila SC Santos, Luis Gustavo J Araujo, and Roberto A Bittencourt. 2018. A Mapping Study of Computational Thinking and Programming in Brazilian K-12 Education. In *2018 IEEE Frontiers in Education Conference (FIE)*. IEEE, 1–8.
- [23] SBC. 2017. Referenciais de Formação em Computação: Educação Básica. <http://www.sbc.org.br/files/ComputacaoEducacaoBasica-versaofinal-julho2017.pdf>.
- [24] Deborah Seehorn, Stephen Carey, Brian Fuschetto, Irene Lee, Daniel Moix, Dianne O'Grady-Cunniff, Barbara Boucher Owens, Chris Stephenson, and Anita Verno. 2011. *CSTA K–12 Computer Science Standards: Revised 2011*. Technical Report. CSTA/ACM, New York, NY, USA. 104111.
- [25] Cynthia Selby and John Woollard. 2013. Computational thinking: the developing definition. (2013). <https://eprints.soton.ac.uk/356481/>
- [26] Cynthia Collins Selby. 2014. *How can the teaching of programming be used to enhance computational thinking skills?* Ph.D. Dissertation. University of Southampton. <https://eprints.soton.ac.uk/366256/>
- [27] Cynthia C. Selby. 2015. Relationships: Computational Thinking, Pedagogy of Programming, and Bloom's Taxonomy. In *Proceedings of the Workshop in Primary and Secondary Computing Education* (London, United Kingdom) (*WiPSCE '15*). ACM, New York, NY, USA, 80–87. <https://doi.org/10.1145/2818314.2818315>
- [28] Matti Tedre and Peter J. Denning. 2016. The Long Quest for Computational Thinking. In *Proceedings of the 16th Koli Calling International Conference on Computing Education Research* (Koli, Finland) (*Koli Calling '16*). Association for Computing Machinery, New York, NY, USA, 120–129. <https://doi.org/10.1145/2999541.2999542>
- [29] The College Board. 2017. *AP Computer Science Principles Course and Exam Description*. College Board. <https://secure-media.collegeboard.org/digitalServices/pdf/ap/ap-computer-science-principles-course-and-exam-description.pdf>
- [30] Henry M. Walker. 2015. Computational Thinking in a Non-majors CS Course Requires a Programming Component. *ACM Inroads* 6, 1 (Feb. 2015), 58–61. <https://doi.org/10.1145/2727126>
- [31] Jeannette M Wing. 2006. Computational thinking. *Commun. ACM* 49, 3 (2006), 33–35.
- [32] Baichang Zhong, Qiyun Wang, Jie Chen, and Yi Li. 2016. An exploration of three-dimensional integrated assessment for computational thinking. *Journal of Educational Computing Research* 53, 4 (2016), 562–590.