



**UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS I
CENTRO DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

SAMARA LIMA GOUVEIA

PENSAMENTO COMPUTACIONAL: UM ESTUDO SOBRE A ABSTRAÇÃO

CAMPINA GRANDE

2021

SAMARA LIMA GOUVEIA

PENSAMENTO COMPUTACIONAL: UM ESTUDO SOBRE A ABSTRAÇÃO

Trabalho de Conclusão de Curso (Artigo) apresentado ao Departamento do Curso de Ciência da Computação na Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Me. Edson Holanda Cavalcante Júnior.

CAMPINA GRANDE

2021

É expressamente proibido a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano do trabalho.

G719p Gouveia, Samara Lima.
Pensamento computacional [manuscrito] : um estudo sobre a abstração / Samara Lima Gouveia. - 2021.
18 p.

Digitado.

Trabalho de Conclusão de Curso (Graduação em Computação) - Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia, 2021.

"Orientação : Prof. Me. Edson Holanda Cavalcante Júnior, Coordenação do Curso de Computação - CCT."

1. Pensamento computacional. 2. Abstração. 3. Ensino-aprendizagem. I. Título

21. ed. CDD 004.7

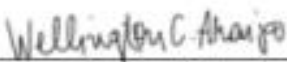
SAMARA LIMA GOUVEIA


Pensamento Computacional: Um Estudo Sobre a Abstração

Trabalho de Conclusão de Curso de Graduação em Ciência da Computação da Universidade Estadual da Paraíba, como requisito à obtenção do título de Bacharel em Ciência da Computação.

Aprovada em 09 de Junho de 2021.


Prof. Msc. Edson Holanda Cavalcante Júnior (DC - UEPB)
Orientador(a)


Prof. Dr. Wellington Candela De Araújo (DC - UEPB)
Examinador(a)


Prof. Msc. Antônio Carlos de Albuquerque (DC - UEPB)
Examinador(a)

“Ideias são à prova de balas.”
(V de Vingança)

SUMÁRIO

1 INTRODUÇÃO	7
2 PENSAMENTO COMPUTACIONAL	8
3 ABSTRAÇÃO.....	9
4 COMO ENSINAR A ABSTRAÇÃO	12
5 CONCLUSÃO	15
REFERÊNCIAS.....	16

PENSAMENTO COMPUTACIONAL: UM ESTUDO SOBRE A ABSTRAÇÃO

Samara Lima Gouveia¹

RESUMO

O pensamento computacional é uma habilidade não muito conhecida, mas que nos últimos anos veio ganhando visibilidade por apresentar benefícios que auxiliam nas atividades diárias e principalmente na resolução de problemas. É um processo que possui semelhanças com a ciência da computação, mas não se refere a saber utilizar um computador e sim a uma maneira de pensar. Dispõe de algumas competências que são necessárias desenvolver, entre elas uma das que merece maior destaque se chama abstração, sendo reconhecida como a base do pensamento computacional. Por possuir poucas pesquisas e não haver muitas orientações sobre o assunto, torna uma habilidade difícil de ser desenvolvida. Assim, foi feita uma pesquisa bibliográfica com o intuito de melhor compreender a abstração no contexto do pensamento computacional, como também verificar se é uma habilidade possível de ser ensinada. Como resultado, foi identificado que existem abordagens que podem ser utilizadas pelos professores em sala de aula, contribuindo para que os alunos consigam pensar de forma abstrata e consequentemente sejam capazes de desenvolver o pensamento computacional.

Palavras-chave: Pensamento Computacional. Abstração. Ensino-aprendizagem de computação.

¹ Universidade Estadual da Paraíba (UEPB) - Campina Grande – PB – Brasil
samara.gouveia@aluno.uepb.edu.br

ABSTRACT

Computational thinking is a little-known skill, but in recent years it has gained visibility for presenting benefits that help in daily activities and, above all, in solving problems. Although it is a process that has similarities with computer science, it does not refer to knowing how to use a computer, but to a way of thinking. For this reason, it has some competencies that are necessary to develop, among them, one of those which deserves more prominence is called abstraction, which is recognized as the basis of computational thinking. Since there is little research and not much guidance on the subject, computational thinking becomes a difficult skill to be developed. Thus, a bibliographic search was made in order to better understand abstraction in the context of computational thinking, as well as to verify whether it is a possible ability to be taught in teaching environments. As a result, it was identified that there are some approaches that can be used by teachers in the classroom, contributing for students to be able to think abstractly and, consequently, to be able to develop computational thinking.

Keywords: Computational Thinking. Abstraction. Computing teaching-learning.

1 INTRODUÇÃO

Saber ler e escrever são competências fundamentais que os seres humanos precisam desenvolver. Vivemos em um mundo que cada vez nos exige mais conhecimentos e habilidades. No século XXI, uma das mais importantes e menos conhecida é o **Pensamento Computacional (PC)** (BLIKSTEIN, 2008). O PC é uma ideia que está bastante relacionada com a ciência da computação por apresentar conceitos concretos e abstratos, podendo ser aplicados universalmente. Foi discutido em 1980 por Seymour Papert através do termo “pensamento processual”, ficando conhecido como um método para resolver problemas utilizando um computador, mas só em 2006 que obteve maior visibilidade após a definição de Wing (BEECHER, 2017).

Wing (2006) afirma que o PC é uma habilidade essencial que o indivíduo precisa desenvolver tanto quanto ler, escrever e saber aritmética. Para que seja possível adquirir essa habilidade é necessário conhecer alguns conceitos que são considerados fundamentais (BEECHER, 2017). Nesse artigo iremos utilizar os conceitos definidos por Selby e Woollard (2013) no qual afirmam que são: abstração, decomposição, design algorítmico, avaliação e generalizações. Dentre eles, a abstração é a mais importante (WING, 2010), pois além de ser uma das competências do PC, ela também está presente nas outras habilidades, sendo necessário desenvolvê-la para poder aplicá-las. Assim, pode-se dizer que utilizar o PC na resolução de problemas é fazer uso de abstrações (CASTILHO, 2018).

Desenvolver a abstração não é uma tarefa simples, além de não ser um conceito fácil de ser ensinado, porque “não se presta a uma definição completa, abrangente e concreta, nem há quaisquer regras específicas relacionadas com a aplicação da abstração”² (HAZZAN e KRAMER, 2007, p. 1, tradução nossa). Qualquer indivíduo mesmo sendo treinado, pode abstrair a informação errada, por isso que é necessário ter uma habilidade mais avançada para entender como aplicar a abstração (NICHOLSON, 2009). Portanto, é importante que os professores sejam orientados para que seja possível o ensino da abstração nas salas de aulas, eles “precisam de mais esclarecimentos para compreender o PC, a relação entre o PC e a abstração, e orientações relacionadas à criação de objetivos, currículo e avaliações, como também objetivos detalhados e resultados por nível de ensino para efetivamente ensinar abstração”³ (LIEBE e CAMP, 2019, p. 7, tradução nossa), no qual o currículo é voltado para conteúdos que devem estar presentes no projeto pedagógico.

Tendo em vista que existem poucos estudos acerca da abstração, percebeu-se a necessidade de fazer uma pesquisa bibliográfica para investigar como é o processo de abstrair e se é uma técnica capaz de ser ensinada. Os meios utilizados para consulta foram livros, artigos e teses, no qual a pesquisa foi feita inicialmente através dos termos “computational thinking” ou “pensamento computacional” para compreender melhor como funciona o PC, depois os termos

² “Does not lend itself to a full, comprehensive and concrete definition, nor are there any specific rules concerned with the application of abstraction.”

³ “[...] need more clarification understanding computational thinking, the relationship between computational thinking and abstraction, and related guidance creating objectives, curriculum, and assessments. Similarly, teachers need detailed objectives and outcomes by grade level to effectively teach abstraction.”

foram alterados para “abstraction and computational thinking” ou “abstração e pensamento computacional” com o intuito de aumentar o escopo para obter resultados que além do PC abordassem a abstração. As bases de dados usadas para as pesquisas foram a plataforma ResearchGate e os anais dos eventos Congresso da Sociedade Brasileira de Computação (CSBC) e Simpósio Brasileiro de Informática na Educação (SBIE).

O presente artigo apresenta na seção 2 a definição do PC e quais são suas características. Na seção 3 aborda a abstração, sendo definida no contexto da computação e do PC, como também os tipos de abstração de acordo com Piaget. Na seção 4 tem-se os métodos que podem ser usados para ensinar a abstração. E finalizando o trabalho, pode-se encontrar na seção 5 a conclusão.

2 PENSAMENTO COMPUTACIONAL

Segundo Wing (2006), o PC é uma forma de “resolver problemas, projetar sistemas e entender o comportamento humano, baseando-se nos conceitos fundamentais da ciência da computação”⁴ (p. 33, tradução nossa). É uma maneira de pensar recursivamente, utilizando a abstração e a decomposição na solução de problemas grandes e complexos, pois não é necessário conhecer todos os detalhes do processo. Apesar do nome lembrar um computador, está relacionado a uma maneira dos humanos pensarem, não sendo um jeito de programar e sim conceituar (WING, 2006). No entanto, Blikstein (2008) aponta que saber programar um computador é uma das etapas necessárias para “pensar computacionalmente”, tornando possível a realização das tarefas cognitivas de modo automático.

Foram os escritos de Seymour Papert que inspiraram o desenvolvimento do PC, mesmo sua definição inicial divergindo das ideias seguintes sobre o assunto (BEECHER, 2017). Ele definiu como um tipo de pensamento desenvolvido por uma criança ao utilizar um computador, no qual se referia “como instrumento para aprender, desenvolver a criatividade e ‘concretizar’ o pensamento computacional” (apud CASTILHO, 2018, p. 4). Acreditava que o uso de tecnologias e ideias computacionais “podem fornecer às crianças novas possibilidades de aprendizagem, pensamento e crescimento emocional e cognitivo”⁵ (PAPERT, 1980, p. 17-18, tradução nossa).

Palts e Pedaste (2019) fizeram um estudo para definir um modelo que possua todas as características do PC. Eles analisaram as abordagens sugeridas por vários autores com o intuito de chegar em um ponto em comum, porém as habilidades acabam divergindo, não havendo uma definição única. Barr e Stephenson (2011) indicam essas habilidades como sendo: coleta de dados, abstração, análise de dados, representação de dados, decomposição do problema, algoritmos e procedimentos, automação, paralelização e simulação. Enquanto a CSTA (2011), aponta como sendo importantes: a formulação de problemas, organização e análise de dados, abstrações, automação por meio do pensamento algorítmico, avaliação para eficiência e correção e generalização.

⁴ “Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science.”

⁵ “I believe that certain uses of very powerful computational technology and computational ideas can provide children with new possibilities for learning, thinking, and growing emotionally as well as cognitively.”

Da mesma forma, Selby e Woollard (2013) fizeram uma investigação para definir de maneira mais restrita as habilidades do PC. Eles chegaram à conclusão de que os termos **abstração** e **decomposição** foram os que mais estavam de acordo com a literatura, enquanto **design algorítmico**, **avaliação** e **generalizações** foram os mais bem definidos. Assim, para que o indivíduo consiga ter uma boa capacidade de pensamento é importante que possua essas competências.

Apesar de ter um conjunto de técnicas que fazem parte do PC, a abstração é considerada como a chave, sendo “o processo de pensamento mais importante e de alto nível no PC”⁶ (WING, 2010, p. 1, tradução nossa). Ela é importante para o desenvolvimento do pensamento e comunicação dos indivíduos. Através da representação, permitiu que os seres humanos evoluíssem cognitivamente para ser o que são hoje, sendo considerada universal, mas o processo de examiná-la não é (NICHOLSON, 2009).

3 ABSTRAÇÃO

A abstração “é essencial na solução de problemas complexos, pois permite que o solucionador de problemas pense em termos de ideias conceituais, e não em termos de seus detalhes”⁷ (HAZZAN e KRAMER, 2007, p. 4, tradução nossa). Ela pode ser definida de várias maneiras, na computação:

É frequentemente caracterizada em termos de dois tipos principais; abstração por parametrização e abstração por especificação (Liskov e Guttag, 2000; Abbott e Sun, 2008). O primeiro envolve abstração de software, onde os elementos computacionais são representados de forma abstrata e os parâmetros são inseridos apenas na instanciação. Esse tipo de abstração é importante na criação de soluções de programação elegantes e generalizáveis. A abstração por especificação ocorre nos estágios iniciais de coleta e design de requisitos e envolve a simplificação da complexidade do mundo real para que possa ser representada em um sistema de computador⁸ (NICHOLSON, 2009, p. 2, tradução nossa).

Enquanto no contexto do PC, a abstração pode ser definida como:

O processo que busca tornar um artefato mais compreensível, reduzindo os detalhes desnecessários. A habilidade em abstração consiste em escolher os detalhes certos a serem ocultados para que o problema se torne mais fácil, sem perder nada que seja importante.

⁶ “The most important and high-level thought process in computational thinking is the abstraction process.”

⁷ “Abstraction is especially essential in solving complex problems as it enables the problem solver to think in terms of conceptual ideas rather than in terms of their details.”

⁸ “Within computing in particular, abstraction is often characterised in terms of two key types; abstraction by parameterization and abstraction by specification (Liskov and Guttag, 2000; Abbott and Sun, 2008). The former entails software abstraction, where computational elements are represented in an abstract form and parameters are only entered on instantiation. This type of abstraction is important in creating elegant and generalisable programming solutions. Abstraction by specification takes place at the earlier stages of requirements gathering and design and involves simplifying real world complexity so it can be represented in a computer system.”

Uma parte essencial disso é escolher uma boa representação de um sistema⁹ (CSIZMADIA et al., 2015, p. 7, tradução nossa).

Dessa forma, é necessário que ocorra a decomposição ou redução dos problemas em suas partes mais essenciais, podendo ser obtida através da generalização de padrões ou da descoberta dos princípios subjacentes que governam os padrões em um problema ou modelo (BENNEDSEN e CASPERSEN, 2006; KRAMER, 2007).

Hazzan, Frorer e Manes (1997) afirmam que a abstração possui três faces. A primeira face consiste em ignorar os detalhes. Ela pode ser usada de duas maneiras: a primeira maneira é “quando você não sabe o suficiente para distinguir entre os elementos individuais de uma classe”¹⁰ (p. 221, tradução nossa). Por exemplo, quando se tem uma multidão de pessoas, inicialmente são apenas alguns rostos indiferenciados, após serem observados por algum tempo será possível diferenciá-los.

A segunda maneira que a primeira face pode ser usada é:

Quando você sabe muito e deseja suprimir os detalhes para ter uma ideia melhor do que está acontecendo. Isso pode acontecer quando você conhece muito bem os indivíduos de um grupo e começa a classificar as pessoas por atributos importantes (ou sem importância!)¹¹ (p. 221, tradução nossa).

Portanto, é preciso pensar bem ao abstrair algum artefato, analisar as informações que deverão ser deixadas de fora, pois dependendo da escolha das informações pode facilitar ou dificultar a resolução do problema. A segunda face da abstração é que ela é relativa. O que é abstrato para uma pessoa não necessariamente vai ser para outra, do mesmo modo, o que é concreto para uma pessoa pode não ser para outra. E a terceira face da abstração consiste em analisar a estrutura, reconhecer as partes importantes e explorar essas partes e suas propriedades.

De acordo com Piaget, existem dois tipos de abstração: empírica e reflexionante. Ele define a abstração empírica como “a abstração que se apoia sobre os objetos físicos ou sobre os aspectos materiais da própria ação, tais como movimento, empurrões etc.” (CASTILHO, 2018, p. 33). Esse tipo de abstração retira características de objetos ou ações que podem ser observados, como comer uma fruta, ouvir uma música, dirigir um carro. (BECKER, 2017).

Enquanto a abstração reflexionante divide-se em duas partes, podendo ser pseudo-empírica e refletida.

A abstração pseudo-empírica consiste em retirar dos observáveis não suas características, mas aquilo que o sujeito colocou neles. Por ela, o sujeito projeta no mundo dos observáveis suas coordenações de ações. Ao retirar características dos observáveis, não retira o que

⁹ “Abstraction is the process of making an artefact more understandable through reducing the unnecessary detail. The skill in abstraction is in choosing the right detail to hide so that the problem becomes easier, without losing anything that is important. A key part of it is in choosing a good representation of a system.”

¹⁰ “The first one is: when you don’t know enough to distinguish among the individual elements of a class.”

¹¹ “The second one is what we were just talking about: when you do know a lot and you want to suppress the details to get a better idea of what’s going on. This might happen when you know the individuals in a group very well, and you start classifying people by important (or unimportant!) attributes.”

pertence aos observáveis, mas o que ele, sujeito, colocou neles (BECKER, 2017, p. 379).

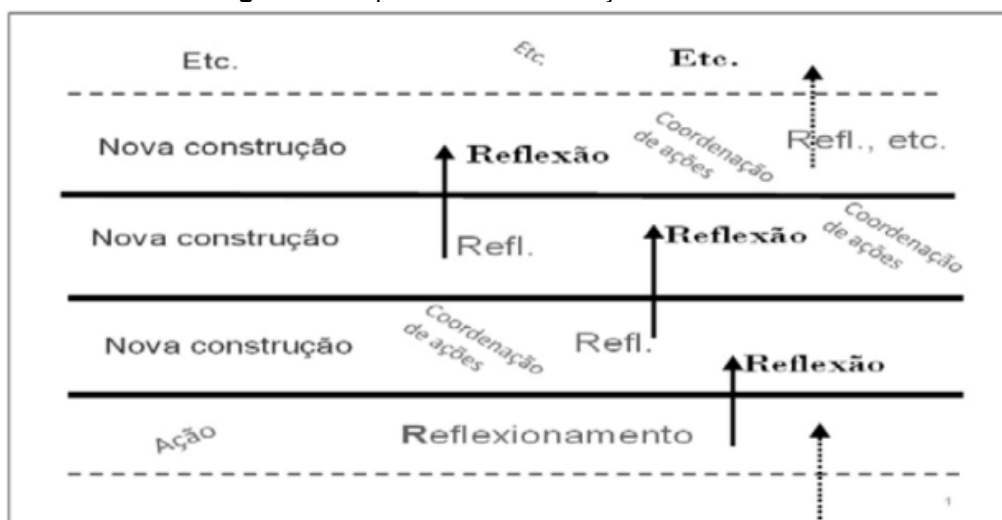
Em outras palavras, a abstração é pseudo-empírica quando a ação do indivíduo modifica o objeto, “sendo enriquecido por propriedades retiradas de coordenações já construídas por este (por exemplo, ordenar os elementos de um conjunto)” (NOGUEIRA e PAVANELLO, 2008, p. 120).

A abstração reflexionante capta qualidades das ações que são realizadas internamente pelo sujeito, não sendo possíveis de serem observadas (BECKER, 2017).

Ela transpõe a um plano superior (reflexionamento) o que retirou de um patamar precedente, sendo denominado esse processo de reflexionamento. Na abstração reflexionante, o sujeito reconstrói sobre o novo plano o que foi colhido do plano de partida, podendo também colocar em relação os elementos extraídos do plano anterior com os já situados no novo plano. Esta reorganização é chamada de reflexão. Quando uma abstração reflexionante torna-se consciente, estamos diante de uma abstração refletida. A abstração refletida consiste na tomada de consciência de uma abstração reflexionante. Ela possibilita a formação de meta-reflexões, reflexão da reflexão, e torna possível a constituição de sistemas lógico-matemáticos de cunho científico (PICETTI, 2008, p. 2-3).

Na figura 1 pode-se observar uma representação de como funciona o processo da abstração reflexionante. Piaget afirma que é um procedimento “sem fim e, sobretudo, sem começo absoluto” (apud BECKER, 2017, p. 377).

Figura 1 – O processo de abstração reflexionante



Fonte: BECKER (2017, p. 377)

Piaget fez um estudo sobre o desenvolvimento cognitivo das crianças desde bebê até a fase adulta. Ele chegou à conclusão de que há quatro estágios diferentes de desenvolvimento, sendo eles: sensório-motor, pré-operacional, operacional concreto e operacional formal. Os dois primeiros estágios têm início quando a criança tem cerca de 7 anos, o terceiro estágio tem duração dos 7 anos até os 12 anos e o último estágio vai dos 12 anos até a fase adulta (apud

KRAMER, 2007). Os estágios acontecem em sequência, não sendo possível passar por um estágio sem antes ter vivenciado o anterior.

No entanto, os estágios podem coexistir e também pode ocorrer que indivíduos adultos, apesar de idade superior, em alguns casos, não tenham chegado ao operatório formal, que se constitui basicamente de elaborações de experiências mentais, também chamadas de experiências lógicas ou reflexivas (CASTILHO, 2018, p. 27).

O estágio sensório-motor é o período da infância em que não são utilizados símbolos, a inteligência é percebida pela coordenação motora e o conhecimento sobre o mundo é adquirido através das experiências físicas. No estágio pré-operacional a inteligência é percebida pelo uso de símbolos, a memória e imaginação são desenvolvidas e é o momento em que o pensamento egocêntrico prevalece. No estágio operacional concreto a inteligência é vista através da manipulação lógica de símbolos que estão relacionados aos objetos concretos. E por último há o operacional formal, o estágio em que a inteligência é demonstrada pelos conceitos abstratos relacionados ao uso lógico de símbolos (apud HUITT e HUMMEL, 2003). É no estágio operacional formal que o indivíduo é capaz de pensar de forma abstrata (KRAMER, 2007).

As abstrações tanto empíricas quanto reflexionantes estão presentes em todos os níveis de desenvolvimento. No estágio sensório-motor, a criança trabalha com abstrações empíricas, enquanto no final desse estágio aparecem as abstrações reflexionantes do tipo pseudo-empíricas que são predominantes no estágio pré-operatório, e por fim tem-se a abstração refletida no estágio operatório formal (apud PICETTI, 2008, BECKER, 2017).

4 COMO ENSINAR A ABSTRAÇÃO

A abstração é uma habilidade fundamental para programadores, engenheiros e até para desenvolver soluções computacionais eficientes. Infelizmente existem poucas pesquisas que oferecem orientações para o ensino da abstração, como também a idade em que os alunos podem começar a aprender, tornando o ensino um desafio (CAMP e LIEBE, 2019). Entretanto, “evidências mais recentes sugerem que os alunos usam elementos de abstração em seu processo geral de aprendizagem, e que a habilidade de abstração pode ser explicitamente ensinada”¹² (WAITE, J. et al., 2016, p. 112, tradução nossa).

Syslo e Kwiatkowska (2014) através de conceitos abstratos de computação conseguiram mostrar que crianças de 5 a 12 anos são capazes de lidar com a abstração. Da mesma forma, Gibson (2012) afirma que crianças de 5 anos estão aptas a trabalharem com a abstração e utilizar o raciocínio computacional. Entretanto, Piaget observou que crianças com idades abaixo de 7 anos estão no estágio pré-operacional e não entendem a lógica concreta, sendo possível manipular apenas informações físicas, enquanto que crianças mais velhas com idades de 7 a 11 anos que estão no estágio operacional concreto só podem resolver problemas reais e não ideias abstratas (ARMONI, 2012).

¹² “However, more recent evidence suggests that pupils use elements of abstraction in their general process of learning, and that the skill of abstraction can be explicitly taught.”

Bruner (1960) defende o modelo de ensino espiral, no qual os conceitos devem ser revistos e o conteúdo ensinado deve corresponder ao estágio de desenvolvimento do indivíduo. Ele diz que “o mais importante para ensinar conceitos básicos é que a criança seja ajudada a passar progressivamente do pensamento concreto para a utilização de modos de pensamento mais conceitualmente adequados”¹³ (tradução nossa, p. 38). Seguindo essa lógica, há a possibilidade de as ideias abstratas serem ensinadas as crianças com idades de 5 a 7 anos através de referências a objetos concretos em que as informações devem ser manipuladas fisicamente (ARMONI, 2012).

Wing (2008) afirma que “definir a abstração ‘certa’ é fundamental”¹⁴ (p. 3718, tradução nossa), pois se as informações não forem muito bem escolhidas podem gerar consequências negativas. Blackwell et al. aponta exemplos de casos em que a abstração não foi bem definida, porém “o problema não é com a abstração em si, mas com abstrações inúteis que escondem informações relevantes ou classificam as coisas incorretamente”¹⁵ (apud NICHOLSON, 2009, p. 3, tradução nossa). De acordo com Armoni, a abstração não é uma habilidade fácil de ser ensinada e não existem métodos comprovados que possam avaliar essa competência (apud WAITE, J. et al., 2018). No entanto, alguns autores sugeriram algumas abordagens que podem ser utilizadas por um instrutor dentro da sala de aula com o intuito de incentivar e auxiliar os alunos a desenvolverem a abstração.

Kramer (2007) afirma que a abstração deve ser ensinada de forma indireta, ou seja, através de outros tópicos, sendo a matemática um excelente veículo para isso. Quando havia menos conteúdo de matemática nos cursos de graduação, os alunos pareciam ser menos capazes de lidar com problemas mais complexos. Keith Devlin observa que o principal benefício de aprender matemática é o desenvolvimento da “capacidade de raciocinar de forma precisa e analítica sobre estruturas abstratas formalmente definidas”¹⁶ (apud KRAMER, 2007, p. 7, tradução nossa).

Para Hazzan (2008), existem outras maneiras que podem ser utilizadas para ensinar a abstração. A primeira consiste em que os educadores apliquem ideias aos seus ensinamentos através de ilustrações. Por exemplo, o professor pode desenvolver algum problema para a turma no qual a solução será feita em etapas dando orientações aos alunos. O segundo método aplica processos reflexivos para fazer com que os alunos pensem entre diferentes níveis de abstração. E a última abordagem é baseada na anterior, pois não é suficiente apresentar apenas definições, para que haja um melhor entendimento da abstração é necessário que o aluno tenha um papel ativo, deve fazer e refletir sobre o que está fazendo.

Diferente das abordagens anteriores, Armoni desenvolveu uma estrutura que apresenta uma hierarquia de níveis para ensinar a abstração (apud RICH e YADAV, 2020). Esta hierarquia possui quatro níveis no qual podem ser visualizados na tabela 1:

¹³ “What is most important for teaching basic concepts is that the child be helped to pass progressively from concrete thinking to the utilization of more conceptually adequate modes of thought.”

¹⁴ “In working with rich abstractions, defining the ‘right’ abstraction is critical.”

¹⁵ “In each of the examples given by Blackwell et al. the problem is not with abstraction per se, but with unhelpful abstractions that hide information which is relevant, or classify things incorrectly.”

¹⁶ “The main benefit of learning and doing mathematics is not the specific content; rather it’s the fact that it develops the ability to reason precisely and analytically about formally defined abstract structures.”

Tabela 1 – Estrutura de níveis desenvolvida por Armoni

Nível	Descrição
Problema	O problema é considerado como uma entidade própria com atributos como solvabilidade e complexidade.
Objeto	Uma solução para o problema na forma de um algoritmo. É considerado como sua própria entidade, não conectado a nenhuma linguagem de programação específica.
Programa	A solução do problema é considerada como um algoritmo escrito em uma determinada linguagem de programação.
Execução	A solução do problema é considerada como uma execução particular de um programa em uma determinada máquina.

Fonte: RICH e YADAV (2020, p. 8)

Na metodologia de Armoni, os instrutores precisam dominar os níveis de abstração e saber distinguir esses níveis para que os alunos saibam exatamente o nível que estão lidando. Além disso, ela indica utilizar componentes linguísticos para a distinção dos níveis. Por exemplo, utilizar uma palavra diferente para diferenciar cada nível. E por último, iniciar a discussão de acordo com a hierarquia dos níveis começando de cima para baixo (apud STATTER e ARMONI, 2016).

Além das abordagens citadas, existem outras atividades que também podem servir de auxílio para o ensino da abstração na sala de aula, como o uso de diagramas rotulados que ajudam o aluno a conseguir identificar o que é mais importante em um objeto. Dessa maneira, os professores fornecem exemplos de modelos mostrando os detalhes que podem ser ignorados. A utilização de mapas conceituais é outra forma que tem como objetivo ensinar classificação e agrupamento, dando uma base de como funcionam os níveis de abstração na programação. Além disso, pode ser também utilizado o storyboard, pois enquanto os alunos planejam escrever, há a possibilidade de decidirem o que deve ser ignorado e incluído. Assim, o storyboard é como se fosse um algoritmo e a história contada é o programa (WAITE, J. et al., 2016).

Cooper, Perez e Rainey recomendam “que o papel do professor no processo de abstração de aprendizagem seja estudado. É essencial entender se o resultado, ou a aprendizagem do aluno que a maioria dos estudos examina, está acontecendo. Alternativamente, ao abordar a entrada de aprendizagem, ou instrução, os pesquisadores podem orientar as melhores práticas de ensino”¹⁷ (apud LIEBE e CAMP, 2019, p. 3, tradução nossa).

¹⁷ “ [...] that the role of the teacher in the process of learning abstraction should be studied. It is essential to understand if the output, or student learning which most studies examine, is happening. Alternately, by addressing learning input, or instruction, researchers can guide teaching best practices.”

5 CONCLUSÃO

Conforme a pesquisa realizada, pode-se concluir que a tarefa de abstrair não é simples, o ponto chave é saber escolher os detalhes mais importantes de acordo com o contexto para que não haja o mal uso da abstração. Além disso, percebe-se que estamos lidando desde cedo com a abstração, seja ela empírica ou reflexionante, em cada estágio do desenvolvimento humano. Porém, observamos na literatura que ocorre divergências com relação a idade em que o indivíduo se torna capaz de lidar com a abstração. Syslo, Kwiatkowska e Gibson afirmam que a partir de 5 anos a criança consegue manipular ideias abstratas, porém Piaget declara que abaixo de 7 anos prevalece o estágio em que só é possível compreender as informações físicas.

Com relação ao ensino da abstração, é possível utilizar abordagens que auxiliem o aluno a desenvolver a abstração. Dos métodos citados, o que mais aparece é o ensino dos níveis de abstração, sendo importante que o aprendizado comece do nível mais simples e aos poucos ocorra o aumento da dificuldade para que o aluno saiba lidar com cada grau de abstração de um problema.

Observa-se também a importância dos processos reflexivos através de atividades que incentivem o aluno a ser mais ativo na construção do conhecimento, utilizando métodos que o leve a refletir e possa identificar quais são as partes mais importantes de um determinado ponto, fazendo assim com que ele passe a ter um olhar mais centrado, deixando de lado as partes desnecessárias.

Por se tratar de um assunto que não possui muitas pesquisas a respeito, é importante que os professores sejam orientados para que o ensino-aprendizagem ocorra da forma mais adequada. Entretanto, ainda faltam evidências que mostrem formas de avaliar essa habilidade. Dessa forma, como trabalho futuro temos como possibilidade pesquisar maneiras de avaliar a abstração para comprovar se ela está realmente sendo desenvolvida e assim ter a certeza de que as abordagens utilizadas nesse artigo são eficazes.

REFERÊNCIAS

- Armoni, M. (2012). Teaching CS in Kindergarten: How early can the Pipeline Begin? *ACM Inroads* 3, 4:18-19.
- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is Involved and what is the role of the computer science education community?. *Acm Inroads*, 2(1), 48-54. <https://doi.org/10.1145/1929887.1929905>
- Becker, F. (2017). Abstração Pseudoempírica : significado epistemológico e impacto metodológico. *Educação & Realidade*, v. 42, n. 1, p. 371–393.
- Beecher, K. (2017). *Computational Thinking: A beginner's guide to problem-solving and programming*, BCS.
- Bennedsen, J., & Caspersen, M. E. (2006). Abstraction ability as an indicator of success for learning object-oriented programming? *ACM SIGCSE Bulletin*, 38(2), 39-43. <https://doi.org/10.1145/1138403.1138430>
- Blikstein, P. (2008). "O Pensamento Computacional e a Reinvenção do Computador na Educação". Disponível em: <<http://bit.ly/1XlbNn>>. Acesso em: 10 de abr. 2016.
- Bruner, J. S. (1960). *The Process of Education*. Cambridge, MA: Harvard University Press.
- Castilho, M. I., Borges, K. S. & Fagundes, L. da C. (2018). A Abstração Reflexionante no Pensamento Computacional e no Desenvolvimento de Projetos de Robótica em um Makerspace Educacional. In *RENOTE – Revista Novas Tecnologias na Educação*, v. 16, n. 1.
- Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). *Computational thinking-A guide for teachers*.
- CSTA. Operational Definition of Computational Thinking. (2011). <http://www.csta.acm.org/Curricu-lum/sub/CurrFiles/CompThinkingFlyer.pdf>
- Frorer, P., Hazzan, O. & Manes, M. (1997). *Revealing the faces of abstraction*. *International Journal of Computers for Mathematical Learning* 2: 217-228, Kluwer Academic Publishers.
- Gibson, J.P. (2012). Teaching graph algorithms to children of all ages. *Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education*(2012), 34–39.
- Hazzan, O. (2008). Reflections on teaching abstraction and other softideas. *ACM SIGCSE Bulletin* 40.2, 40-43.
- Huitt, W., & Hummel, J. (2003). Piaget's theory of cognitive development. *Educational Psychology Interactive*. Valdosta, GA: Valdosta State University.
- Hazzan, O., & Kramer, J. (2007). "Abstraction in computer science & software engineering: a pedagogical perspective", *Frontier Journal*, vol. 4, pp. 6–14.

Kramer, J. (2007). "Is abstraction the key to computing?" In: Communications of the ACM 50.4, pp. 36–42. issn: 00010782. doi: 10.1145/1232743.1232745.

Liebe, C., & Camp, T. (2019). An Examination of Abstraction in K-12 Computer Science Education. In 19th Koli Calling International Conference on Computing Education Research (Koli Calling '19), November 21–24, 2019, Koli, Finland. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3364510.3364526>.

Nicholson K., Good J., & Howland K. (2009). Concrete Thoughts on Abstraction. Psychology of Programming Workshop (PPIG 2009), University of Limerick, Ireland.

Nogueira, C. M. I., & Pavanello, R. M. (2008). A abstração reflexionante e a produção do conhecimento matemático. *Revista Bolema*, (30), 111-130.

Palts, T., & Pedaste, M. (2019). A Model for Developing Computational Thinking Skills. *Informatics in Education*, 2020, Vol. 19, No. 1, 113–128. 10.15388/infedu.2020.06.

Papert, S. (1980). *Mindstorms: children, computers and powerful ideas*. New York: Basic Books, Inc.

Picetti, J. S. Formação continuada de professores: da abstração reflexionante à tomada de consciência. Porto Alegre: UFRGS, 2008. 144f. Tese (Doutorado em Educação) – Programa de Pós-Graduação em Educação. Faculdade de Educação. Universidade Federal do Rio Grande do Sul, Porto Alegre, 2008.

Rich, K.M., & Yadav, A. (2020). Applying Levels of Abstraction to Mathematics Word Problems. *TechTrends* 64, 395–403. <https://doi.org/10.1007/s11528-020-00479-3>.

Selby, C. C., & Woollard, J. (2013). Computational Thinking: The Developing Definition. University of Southampton (E-prints) 6 pp. <https://eprints.soton.ac.uk/356481>

Statter, D., & Armoni, M. (2016). Teaching abstract thinking in introduction to computer science for 7th graders. In Proceedings of the 11th Workshop in Primary and Secondary Computing Education - WiPSCE '16 (pp. 80–83). New York: ACM.

Syslo, M. M., & Kwiatkowska, A. B. (2014). Playing with computing at a children's university. In Proceedings of the 9th Workshop in Primary and Secondary Computing Education. ACM, pp. 104–107.

Waite, J. L., Curzon, P., Marsh, W., & Sentance, S. (2016). Abstraction and common classroom activities. In Proceedings of the 11th Workshop in Primary and Secondary Computing Education. ACM, pp. 112-113. <https://doi.org/10.1145/2978249.2978272>.

Waite, J. L., Curzon, P., Marsh, W., Sentance, S., & Hadwen-Bennett, A. (2018). Abstraction in action: K-5 teachers' uses of levels of abstraction, particularly the design level, in teaching programming. *International Journal of Computer*

Science Education in Schools, 2(1), 14–40.
<https://doi.org/10.21585/ijcses.v2i1.23>.

Wing, J. M. (2006). Computational thinking. *Commun. ACM* 49, 33–35.
<https://doi.org/10.1145/1118178.1118215>

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717-3725.
<https://doi.org/10.1098/rsta.2008.0118>

Wing, J. M. (2010). Computational thinking: What and why?. Unpublished manuscript Computer Science Department, Carnegie Mellon University, Pittsburgh, PA.