

Alan Turing tinha Pensamento Computacional? Reflexões sobre um campo em construção

José Armando Valente (IA&NIED/UNICAMP)¹

Fernanda Maria Pereira Freire (NIED/UNICAMP)²

Flavia Linhalis Arantes (NIED/UNICAMP)³

João Vilhete Viegas d'Abreu (NIED/UNICAMP)⁴

Tel Amiel (NIED/UNICAMP)⁵

Maria Cecília Calani Baranauskas (IC&NIED/UNICAMP)⁶

Resumo

A expressão pensamento computacional, amplamente usada nos últimos anos, reaviva um antigo debate em torno da importância de trazer práticas da ciência da computação para a educação. A partir da análise de um conjunto de estudos e de currículos procuramos: (i) mostrar algumas relações entre “pensamento computacional” e “pensamento algébrico” e (ii) refletir sobre possíveis impactos educacionais, sociais e econômicos desse crescente movimento. Nosso objetivo é contribuir para uma discussão fundamentada sobre o tema, considerando que, no contexto atual, existem questões econômicas e geopolíticas que resultam em diferentes metas para a educação.

Palavras-chave: Ciência da computação, pensamento algébrico, tecnologias educacionais, educação, TIC

Abstract

The term computational thinking, widely used in recent years, revives an old debate about the importance of bringing practices from computer science to education. From the analysis of a set of studies and curricula we try to: (i) show some relations between "computational thinking" and "algebraic thinking" and (ii) reflect on possible educational, social and economic impacts of this growing movement. Our objective is to contribute to a reasoned discussion about the theme, considering that there are current economic and geopolitical issues that result in different goals for education.

Keywords: Computer science, algebraic thinking, educational technologies, education, ICT

¹ Contato: jvalente@unicamp.br

² Contato: ffreire@unicamp.br

³ Contato: farantes@unicamp.br

⁴ Contato: jvilhete@unicamp.br

⁵ Contato: tamiel@unicamp.br

⁶ Contato: cecilia@ic.unicamp.br

1. Introdução

A expressão “pensamento computacional” tem sido utilizada nos últimos anos, particularmente após a publicação do *viewpoint* de autoria de Jeannette M. Wing na Revista *Communications* da ACM, no qual a pesquisadora o define como uma “(...) atitude e habilidade universalmente aplicáveis que todos, não apenas os cientistas da computação, estariam ansiosos por aprender e usar”⁷ (Wing, 2006, p. 33, tradução dos autores).

Wing (2006) ressalta a abstração como uma das características desse tipo específico de pensamento. Observação similar já estava presente no trabalho de Papert (1980) e que antecede em algumas décadas o estudo atual sobre o tema. Resguardada a evolução tecnológica ocorrida desde então, o que presenciamos é o renascimento da discussão em torno do pensamento computacional em novas bases (Tedre; Demming, 2016).

Não se pode negar que à medida que novas tecnologias se incorporam ao cotidiano das pessoas, mais e mais se ampliam as possibilidades de se identificar tarefas cognitivas que podem ser feitas de forma mais rápida e eficiente, utilizando procedimentos realizados por máquinas, envolvendo abstração, generalização e manipulação simbólica. No entanto, pensar e agir com base em um determinado raciocínio lógico, antecede o advento dos computadores.

A tentativa de entender e de explicar processos mentais não é nova. Até o início do século passado era uma atividade praticada por filósofos e psicólogos. No entanto, os resultados dos estudos sobre lógica matemática realizados no período entre 1910-1913 por Bertrand Russell e Alfred North Whitehead influenciaram alguns pesquisadores, como Norbert Wiener e John Von Neumann, conhecidos como ciberneticistas, que lançaram as bases do que se conhece hoje como ciência cognitiva (Gardner, 1985).

A fase cibernética da ciência da computação produziu resultados nada desprezíveis: a invenção da máquina de processamento da informação, a teoria de informação como uma teoria estatística e o uso da lógica matemática para entender operações do sistema nervoso. Em 1943, o trabalho de Warren McCulloch e Walter Pitts (*A Logical Calculus of Ideas Immanent in Nervous System*) propunha duas ideias importantes: a primeira, de que a lógica é a disciplina adequada para entender o cérebro e as atividades mentais e, a segunda, que o cérebro é um “dispositivo” que incorpora princípios de lógica nos seus

⁷ Do original em inglês “Computational Thinking. It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use” (Wing, 2006, p. 1).

componentes ou neurônios. Essas ideias foram centrais para o desenvolvimento dos computadores digitais (Varela et al., 1993).

Podemos, assim, entender que um modelo computacional é uma abstração lógico-matemática, sendo que o modelo mais importante é a máquina de Turing. Foi a lógica matemática que permitiu a Alan Turing, em 1936, desenvolver o conceito de uma máquina simples que pudesse realizar qualquer tipo de cálculo. Certamente, Turing usou pensamento lógico, dedutivo e matemático para criar o modelo que até hoje influencia diretamente a construção de artefatos computacionais.

Considerando esses fatos podemos questionar se, realmente, para desenvolver modelos computacionais é necessário ter “pensamento computacional” ou, se o “pensamento computacional” é uma derivação do pensamento matemático. Se essa última hipótese for plausível, qual a razão de hoje presenciarmos um movimento global de valorização do pensamento computacional, a ponto de se supor que todos “estariam ansiosos por aprender e usar”, como afirma Wing (2006)?

Neste artigo procuramos problematizar a questão do pensamento computacional, desdobrando-a em algumas direções que, nos parecem, precisam ser analisadas a fim de refletirmos sobre possíveis impactos educacionais, sociais e econômicos.

2. Alguns pontos de vista sobre Pensamento Computacional

Segundo Tedre e Denning (2016), Papert parece ter sido o primeiro a usar o termo “pensamento computacional” em seu livro *Mindstorms* (Papert, 1980, p. 182). Papert desenvolveu uma abordagem empírica para a construção de conhecimento utilizando computadores e a linguagem Logo. O autor considera a resolução de problemas com o auxílio do computador como uma forma alternativa de pensar; uma abordagem distinta, mas não superior de pensamento (Turkle; Papert, 1990), condicionando o seu aprendizado à possibilidade de o aprendiz alternar entre, pelo menos, dois modelos de pensamento (um mecânico e outro não mecânico).

Para Wing (2006) o pensamento computacional envolve resolução de problemas e concepção de sistemas com base em conceitos fundamentais da ciência da computação, bem como a reformulação de um problema aparentemente difícil em outro problema cuja solução já é conhecida. O pensamento computacional, para a autora, envolve abstração e decomposição para lidar com um problema complexo; escolha de uma representação adequada e modelagem de aspectos relevantes do problema para torná-lo tratável

computacionalmente. Implica ainda processamento paralelo e pensamento recursivo, interpretação de código, corretude e eficiência.

A autora afirma que o pensamento computacional pode servir como um instrumento de mudança na maneira como biólogos, estatísticos, economistas, entre outros cientistas, pensam, representam e resolvem seus problemas. O pensamento computacional, prevê Wing, será parte do conjunto de habilidades não só de outros cientistas, mas de todas as pessoas, a ponto de se tornar tão arraigado no cotidiano quanto as palavras “algoritmo” e “pré-condição” já o são. Esse novo movimento a favor do pensamento computacional não se limita, portanto, àqueles que produzem ferramentas computacionais, mas visa incluir também aqueles que usam essas ferramentas e aqueles que se envolvem em procedimentos algorítmicos. Wing propõe, ainda, acrescentar o pensamento computacional à capacidade analítica que cada criança desenvolve com a leitura, a escrita e a aritmética⁸.

Aho (2011) analisa os processos de pensamento em computação descritos desde a década de 1950, considerando autores que desenharam e implementaram modelos computacionais. O autor chama a atenção para a necessidade de se clarificar o próprio conceito de “computação”, uma vez que a natureza dos sistemas que exibem comportamento computacional é variada e o termo significa coisas diferentes para pessoas diferentes, dependendo do tipo de sistema computacional que se estuda e do tipo de problema que se investiga⁹. Alinhado ao entendimento de Wing, Aho relaciona o pensamento computacional a processos envolvidos na modelagem de problemas, de modo que suas soluções possam ser representadas como passos de algoritmos. Para ele, uma parte importante desse processo é encontrar modelos apropriados de computação com os quais seja possível formular o problema e derivar suas soluções. Aho acredita que bons modelos computacionais para processos biológicos, por exemplo, ainda são incipientes e afirma não saber se será possível elaborar um modelo computacional para o cérebro humano que explique fenômenos emergentes, como a consciência ou a inteligência.

Dijkstra (1974, apud Tedre; Denning, 2016) acreditava que a singularidade da computação vem do “pensamento algorítmico”, caracterizado por ele como: (1) domínio da linguagem natural para preencher a lacuna entre os problemas informalmente expressos e as soluções formais; (2) capacidade de inventar os próprios formalismos e conceitos na

8 Do original em inglês “To reading, writing, and arithmetic, we should add computational thinking to every child’s analytical ability” (Wing, 2006, p. 33).

9 O pensamento-rede (Proulx, 2010) é um exemplo da diversidade de pensamentos que pode surgir na interação entre o homem e as tecnologias de informação e comunicação, neste caso, o uso intensivo de redes digitais de comunicação.

solução de problemas; (3) agilidade para alternar entre diferentes níveis semânticos, em uma espécie de lente ou de “zoom mental” (Tedre; Denning, 2016, p. 121).

Outros autores também têm buscado definir o pensamento computacional de maneira prática usando, por exemplo, matrizes e listas de conteúdos e competências. Zapata-Ros (2015) afirma que o pensamento computacional é um pensamento específico, isto é:

[...] una forma de pensar propicia para el análisis y la relación de ideas, para la organización y la representación lógica. Esas habilidades se ven favorecidas con ciertas actividades y con ciertos entornos de aprendizaje desde las primeras etapas (Zapata-Ros, 2015, p. 1).

Para o autor o pensamento computacional é um complexo conjunto de habilidades, mais precisamente, catorze: metacognição, *sinéctica*, análise descendente e ascendente, recursividade, método por aproximação de sucessivas tentativas e erros, heurística, iteração, pensamento divergente, criatividade, resolução de problemas, pensamento abstrato, métodos colaborativos e padrões.

As competências e habilidades identificadas por pesquisadores interessados no pensamento computacional tem se materializado na forma de currículos e recomendações de instituições de ensino, organizações e empresas, como se verá na próxima seção.

3. Pensamento Computacional e o ensino

O documento elaborado pelas organizações estadunidenses *National Science Foundation*, *International Society for Technology in Education* e *Computer Science Teachers Association* (CSTA, 2011) apresenta, talvez, a versão mais prática e influente de definição do pensamento computacional. Considerado como um processo de resolução de problemas, o pensamento computacional inclui (mas não se restringe) as seguintes características¹⁰:

- Formular problemas de maneira que permita o uso do computador e outras ferramentas para que sejam resolvidos;
- Organizar e analisar dados de maneira lógica;
- Representar dados através de abstrações como modelos e simulações;
- Automatizar soluções através do pensamento algorítmico (passos sequenciais);
- Identificar, analisar e implementar soluções possíveis com o objetivo de atingir a combinação mais eficiente e eficaz de passos e recursos;

¹⁰ Tradução dos autores.

- Generalizar e transferir esse processo de resolução de problemas para uma variedade de problemas.

Já, o documento que orienta o curso *Advance Placement Computer Science Principles Course*¹¹ nos EUA, sugere uma lista diversificada de temas, incluindo: aspectos humanos e sociais da computação; design de artefatos computacionais em contextos criativos/algorítmicos; abstração; análise de problemas e artefatos; comunicação e clareza quanto aos processos e produtos; e colaboração. Essas áreas são pensadas como um processo cíclico de depuração, utilizado por profissionais da computação e de outras áreas do conhecimento. O documento enfatiza que o currículo está intimamente ligado à ciência da computação e que esse curso serve como alternativa ao curso de introdução à computação no ensino superior, embora o primeiro abranja conhecimentos que são úteis em várias disciplinas, pelo fato de usar “processos criativos no desenvolvimento de artefatos computacionais” (College Board, 2016, p.4).

Temas similares são usados no currículo de computação no ensino básico nos EUA, por organizações como a *Association for Computer Machinery (ACM)* que tem vários parceiros, incluindo empresas como Microsoft, Amazon e Google¹².

No Brasil, a terceira versão do documento da Base Nacional Comum Curricular (BNCC), disponibilizada pelo Ministério da Educação em 2017, inclui o pensamento computacional na disciplina de Matemática:

Outro aspecto a ser considerado é que a aprendizagem de Álgebra pode contribuir para o desenvolvimento do pensamento computacional dos alunos, tendo em vista que eles precisam ser capazes de traduzir uma situação dada em outras linguagens, como transformar situações-problema, apresentadas em língua materna, em fórmulas, tabelas e gráficos e vice-versa.

Associado ao pensamento computacional, cumpre salientar a importância dos algoritmos e de seus fluxogramas, que podem ser objetos de estudo nas aulas de Matemática. Um algoritmo é uma sequência finita de procedimentos que permite resolver um determinado problema. Assim, o algoritmo é a decomposição de um procedimento complexo em suas partes mais simples, relacionando-as e ordenando-as, e pode ser representado graficamente por um fluxograma. A linguagem algorítmica tem pontos em comum com a linguagem algébrica, sobretudo em relação ao conceito de variável. Outra habilidade relativa à álgebra que mantém estreita relação com o pensamento computacional é a identificação de padrões para se estabelecer generalizações, propriedades e algoritmos (Brasil, 2016. p. 227).

¹¹ Cursos avançados para alunos do ensino médio para que obtenham créditos para o ensino superior (Curso de Princípios da Ciência da Computação).

¹² Não por menos, um dos parceiros é a *College Board*, responsável pelos cursos de *Advanced Placement (AP)* mencionados anteriormente (K12CS, 2017).

O documento da BNCC vincula fortemente o pensamento computacional ao pensamento algébrico, provavelmente em função da interdependência entre as áreas da computação e da matemática, tornando ainda mais difícil distingui-los.

4. Pensamento algébrico e computacional

Kieran (1992, 2004, 2007, apud Fernandes; Savioli, 2016) afirma que o pensamento algébrico consiste na atividade de generalizar, analisar e representar relações matemáticas e relações entre quantidades e grandezas, resolver problemas, modelar, justificar, realizar previsões, estabelecer padrões e regras. Ocorrem manifestações de indícios de pensamento algébrico ao revelar e argumentar a respeito de ideias algébricas, mesmo que em linguagem natural, ou por meio de diagramas, tabelas, expressões numéricas e gráficos ou, ainda, pela transição entre esses elementos e notações.

Fiorentini, Miorim e Miguel (1993 apud Fernandes; Savioli, 2016) definem o pensamento algébrico como um tipo de pensamento que pode ser manifestado por meio da percepção de regularidades, aspectos invariantes e variantes, tentativas de expressar e explicar a estrutura de uma situação-problema e o processo de generalização. Fiorentini, Fernandes e Cristovão (2005 apud Fernandes; Savioli, 2016) afirmam que o pensamento algébrico pode ser desenvolvido gradativamente antes da linguagem simbólica (algébrica, propriamente dita) e se manifesta quando um sujeito estabelece relações/comparações entre expressões numéricas ou padrões geométricos; utiliza expressões de estruturas aritméticas diferentes para uma mesma situação-problema; produz vários significados para uma expressão numérica; interpreta igualdade como equivalência entre grandezas ou entre expressões numéricas; desenvolve algum tipo de processo de generalização; percebe e tenta expressar regularidades ou invariâncias; desenvolve/cria uma linguagem mais concisa ou sincopada ao expressar-se matematicamente.

Kaput (1999 apud Fernandes; Savioli, 2016) associa o pensamento algébrico à capacidade de utilizar diferentes sistemas de representação, raciocinar dedutiva e indutivamente, relacionar, generalizar, modelar, realizar conjecturas e argumentar a respeito dessas conjecturas. Este pensamento tem como principais aspectos a generalização e formalização de padrões e restrições, manipulação de formalismos, estudo de estruturas abstratas a partir de cálculos e relações, estudo de funções, relações e variação de variáveis, utilização de múltiplas linguagens na modelagem matemática e no controle de fenômenos. Blanton e Kaput (2005 apud Fernandes; Savioli, 2016) apontam que o pensamento algébrico é um tipo de pensamento por meio do qual se pode generalizar ideias matemáticas a partir de um conjunto de casos particulares e tem como principais

características o uso da aritmética como campo para expressar e formalizar generalizações (aritmética generalizada); generalização de padrões numéricos para descrever relações funcionais, em que ocorre a exploração entre correspondência de quantidades, relações recursivas e desenvolvimento de regras para descrever relações (pensamento funcional); modelagem de situações e a generalização a respeito de sistemas matemáticos abstratos de cálculos e relações.

Para Ponte, Branco e Matos (2009 apud Fernandes; Savioli, 2016) este pensamento inclui a capacidade de lidar com expressões algébricas, equações, inequações, sistemas de equações e funções, assim como a percepção de relações e estruturas matemáticas, análise de propriedades matemáticas e interpretação de símbolos matemáticos. Além disso, o pensamento algébrico se relaciona à representação de forma simbólica, raciocínio dedutivo e indutivo e utilização de representações diversas de objetos algébricos para resolver problemas de diferentes domínios.

Para Lins e Gimenez (1997 apud Fernandes; Savioli, 2016) pensamento algébrico relaciona-se à aritmética e é um dos modos de produzir significado para a álgebra. Tem como principais características o “aritmecismo” (produção de significados apenas em relação a números e operações aritméticas), internalismo (considerar números e operações apenas segundo suas propriedades) e a analiticidade (ao operar com números desconhecidos como se fossem conhecidos). Além disso, manifesta-se por meio da investigação de regularidades, sistematização de propriedades observadas, resolução e discussão de problemas algébricos, modelagem de situações e estabelecimento de padrões.

Na tentativa de definir as especificidades do pensamento algorítmico, Knuth (1985, apud Tedre; Denning, 2016) comparou textos da área de matemática que discutiam padrões de pensamento e os contrastou com os padrões usados por cientistas da computação. Ele cita a representação da realidade, a redução a problemas mais simples e o raciocínio abstrato como comuns a ambos; a complexidade e a causalidade, no entanto, só se fizeram presentes nos padrões descritos pelos cientistas da computação, considerando a complexidade (ou economia de processamento) e o design de procedimentos que geram ação no mundo.

Não é difícil identificar nessas definições pontos de intersecção entre o que é da ordem do “algébrico” e do “computacional”. Expressões como representar, modelar, formular e resolver problemas, generalizar, padronizar, identificar, analisar, são usadas para explicar a natureza desses pensamentos. A dificuldade em se definir claramente um ou outro tipo de pensamento, talvez resida no fato de ambos serem atividades cognitivas complexas que

envolvem uma simbologia e muitos processos, alguns dos quais são comuns a vários tipos de pensamento, além dos dois tratados aqui.

Argumentos como os mencionados acima têm sido utilizados por matemáticos que trabalham na área de uso de tecnologias na educação, como Celia Hoyles, para justificar a ideia de que é suficiente o desenvolvimento do pensamento matemático para ser capaz de resolver problemas. A autora enfatiza que o uso de ferramentas flexíveis é importante, principalmente se elas permitem ajustar e combinar parâmetros, programar, e auxiliar o aluno a investigar por si só diferentes alternativas (Hoyles, 2016).

Não se trata, portanto, de criar alternativas de “pensamentos”, mas de se entender, como menciona Gadaniadis et al. (2016), que existe uma conexão natural entre o pensamento computacional e a matemática tanto na estrutura lógica de ambos quanto na possibilidade de modelar relações. O que parece haver de novo é que o pensamento computacional possibilita novas abordagens criativas para a resolução de problemas de matemática e uma ampla gama de atividades com a qual os estudantes de todos os níveis podem se envolver. A pesquisa de Hoyles tem como objetivo usar o pensamento computacional na educação matemática, desde a educação pré-escolar até a matemática de graduação e na formação de professores de matemática (Hoyles, 2017)¹³.

5. Alguns desdobramentos educacionais, sociais e econômicos

Vimos, até agora, falando sobre a aplicação de uma série de estratégias, métodos e modelos a uma classe de atividades que envolvem artefatos computacionais. Cabe perguntar, assim, quais as razões que justificam um movimento (global) crescente que advoga a favor da necessidade do pensamento computacional para qualquer profissional do presente e, principalmente, do futuro.

No artigo de Wing (2006) há um subtexto a partir do qual se pode inferir que o pensamento computacional pode ser aplicado a qualquer área do conhecimento. Assim, a elaboração de currículos, materiais, plataformas, cursos e certificações em torno do pensamento computacional pode gerar um mercado lucrativo, caso governos, escolas e instituições sejam convencidos dessa necessidade. Em vários contextos, a formação em

¹³ Nessa mesma linha do que é proposto por Hoyles, porém não vinculado à área de matemática, é possível pensar em projetos relacionados a outras disciplinas do currículo que exploram as ideias de pensamento computacional em atividades como robótica, produção de narrativas digitais e *maker* (Valente, 2016).

computação é considerada tão essencial para a formação de alunos a ponto de ser incluída em currículos, junto com outras áreas de conhecimento (como artes, ou música).

A manifestação mais concreta do interesse pelo pensamento computacional é a sua rápida e crescente inserção como componente curricular obrigatório no ensino básico em diversos países, entre eles, Inglaterra, EUA, Israel, Austrália. O pensamento computacional nesses currículos envolve desde habilidades de colaboração até conhecimentos sobre o impacto das tecnologias na sociedade, tópicos já presentes em vários currículos anteriores de formação docente e discente (ITEA, 2007; ISTE, 2007; UNESCO, 2009).

Para além dos currículos formais, parece interessante apontar as iniciativas voltadas, especificamente, para o ensino de linguagens de programação, que são de alta visibilidade e influenciam o senso comum a respeito da demanda e da necessidade de programadores, bem como da importância das habilidades computacionais para qualquer área do conhecimento. O exemplo mais visível é a iniciativa *Code.org*, dedicada ao ensino de programação, que tem atividades para alunos do segundo ano do ensino fundamental, disponível em mais de 45 idiomas. Plataformas como *Code.org* são promovidas por agentes de empresas, governos e ativistas ao redor do mundo, sob o argumento de que alunos se beneficiam do programa e que a indústria precisa de programadores (Singer, 2017). Apoiam-se também em uma noção já explorada de que se trata de uma nova alfabetização, imprescindível na sociedade contemporânea (Frag, 2016).

Iniciativas como a do *Code.org* e de outros projetos (por exemplo, o *Women Who Code*) se apoiam em um discurso de promoção de equidade voltado, particularmente, para minorias e mulheres. Denunciam um campo de atuação (computação e tecnologia) que tem sido pouco representado por grupos que se sentem (e são) excluídos desse universo, perpetuando uma mentalidade única em torno do desenvolvimento de empresas nessa área. No entanto, por trás desse discurso,

[...] há um interesse em garantir uma massa de trabalhadores que possa suprir a demanda atual e futura da indústria de software. Como ressalta Audrey Watters a noção de que “todos devemos saber programar” não deve ser vista de maneira tão abrangente. É necessário melhor entender o que “programar” significa e para quem, ou seja, é importante entender as diferentes metodologias e objetivos que compõem as motivações do “aprender a programar” (Watters, 2014). Nesse sentido, podemos tentar acelerar cada vez mais a exposição de crianças às novas mídias para que possam melhor competir por vagas em universidades. Podemos também fazer do ensino médio cada vez mais uma coleção de competências, e inserir a programação como mais uma “linguagem” a ser desenvolvida (Amiel et al., 2015, p. 1-2).

Na esteira desse movimento há, na verdade, uma forte motivação econômica. A tendência é a de que a presença e a mediação de dispositivos computacionais cresçam em todas as esferas econômicas. Entretanto, para manter esse crescimento é preciso também haver aumento de oferta de mão de obra. Nesse caso, no entanto, a busca por trabalhadores com “pensamento computacional” muitas vezes se traduz na instrumentalização através da programação. Multiplicam-se academias ou (*bootcamps*) que se propõem a ensinar programação de maneira rápida e intensiva¹⁴, deixando evidente a diferença profunda entre a “programação instrumental” e o que, de fato, constitui um engenheiro, um analista ou um programador efetivamente envolvido na indústria (Norvig, 2014).

A urgência do mercado não permite uma formação de qualidade e o déficit de mão de obra na área de programação é tão grande que ela se torna, de acordo com uma nova abordagem epistemológica, inviável. As grandes empresas que precisam desse tipo de mão de obra são as mesmas que financiam projetos como *Code.org* (Google, Facebook, Microsoft)¹⁵. O objetivo dessas iniciativas pode ser o de promover o desenvolvimento do pensamento computacional de maneira abrangente, mas na prática, para a grande maioria dos envolvidos, o aprendizado se resume a um curso básico na web de alguma linguagem de programação, junto a um enunciado que remete a uma experiência divertida e simples.

Esta tática tem levado a uma mudança radical na política educacional fomentada pela entrada das empresas do Vale do Silício na educação. Trata-se de um embate filosófico sobre os propósitos da escola: o de produzir “cidadãos com conhecimento ou trabalhadores com habilidades” (Singer, 2017).

Aprender a programar não é algo simples – exige estudo e dedicação – muito além do que a aprendizagem sintática de uma linguagem de programação pode oferecer (Norvig, 2014). A sensibilização ou a formação oferecida por essas plataformas e a oferta de cursos sobre lógica de programação não são um problema em si; a promessa de sucesso, alinhada a um cenário de urgência na transformação de currículos e políticas de financiamento educacionais, sim. Em um currículo onde matérias como música, sociologia e educação física sempre correm o risco de serem abandonadas, há de se pensar com cuidado sobre a inserção de novas prioridades curriculares. O mesmo vale para priorização de gastos de governo. Uma corrida em torno do pensamento computacional pode levar a gastos

¹⁴ Veja-se uma listagem, por exemplo, em Coursereport (2017).

¹⁵ Isso não é uma novidade. Empresas de tecnologia regularmente patrocinam estudos, currículos, formação e avaliações que envolvem a tecnologia na educação e correlatos como “habilidades para o século XXI”.

significativos de verbas públicas, justificados por uma corrida competitiva ou por interesses comerciais¹⁶.

Este movimento se reflete em todas as áreas do conhecimento, incluindo as humanidades tornadas “digitais”. Não se trata de uma “nova área do conhecimento, mas de uma gama de atividades” (Marques, 2017) envolvendo uma aproximação com cientistas da computação e aplicando tecnologias computacionais aos estudos realizados nas humanidades. Allington, Brouillette e Golumbia (2016) sugerem que a digitalização e a promoção desse modo de pensar, não serve somente para criar novas cooperações e atividades interdisciplinares, mas promove uma subversão das estruturas da educação pública a serviço do capital especulativo:

Essa é uma das principais razões pelas quais a digitalização de arquivos e o desenvolvimento de ferramentas de software – atividades suspeitosamente caras que agências de fomento influentes tem entusiasticamente apoiado – podem exercer atração tão poderosa, efetivamente permitindo que atividades acadêmicas sejam reconfiguradas no modelo de uma startup da tecnologia, com financiamento público, privado e doações no lugar do capital oriundo do Vale do Silício¹⁷ (Allington; Brouillette; Golumbia, 2016).

Essas e outras questões nos levam a enxergar o pensamento computacional em um contexto contemporâneo onde grandes empresas do Vale do Silício, grandes fundações e o capital especulativo influenciam de maneira decisiva o direcionamento das políticas educacionais. Os debates em torno das definições de pensamento computacional, suas limitações e aplicabilidades estão tomando corpo e compõem grande parte da literatura. Cremos que parte de seu apoio advém de cientistas da computação e outros entusiastas que encontraram na noção de pensamento computacional um conceito “guarda chuva” que poderia unir atores em torno da bandeira de um movimento único. Vemos, no entanto um limitado debate em torno do momento e do contexto político e econômico que levaram ao recente ressurgimento do pensamento computacional.

¹⁶ Veja-se, por exemplo, a recente (junho de 2017) audiência pública para compra de Robótica Educacional para o ensino infantil, fundamental e médio. Há ênfase clara no ensino de programação nos anos finais do ensino fundamental e médio, menções à competitividade e a preparação para o mundo do trabalho. A íntegra da gravação da audiência pública está disponível em FNDE MEC (2017)

¹⁷ Do original em inglês: “This is one of the main reasons why the digitization of archives and the development of software tools — conspicuously expensive activities that influential funding bodies have enthusiastically supported — can exert such powerful attraction, effectively enabling scholarship to be reconfigured on the model of the tech startup, with public, private, and charitable funding in place of Silicon Valley venture capital” (Allington; Brouillette; Golumbia, 2016).

6. Considerações finais

Neste texto buscamos (re)conhecer as especificidades do pensamento computacional, a partir de estudos e de currículos. Consideramos a importância de compreendermos quem são os atores (individuais e institucionais) desse movimento a favor do pensamento computacional, de onde falam (geográfica e institucionalmente) e suas motivações para não repetirmos erros anteriores, talvez, por ingenuidade.

Hemmendinger (2010) diz que o pensamento computacional pode ajudar economistas, artistas e outros profissionais a “entender como usar a computação para resolver seus problemas; para criar, e descobrir novas perguntas que podem ser exploradas de maneira frutífera”. Entretanto, é bom lembrar, que o autor pede modéstia aos seus colegas da ciência da computação afirmando que “no final, porém, talvez devêssemos falar menos sobre o pensamento computacional e focar mais no fazer computacional - realizando o trabalho (ou a brincadeira!) de alguém de novas maneiras usando ferramentas computacionais” (Hemmendinger, 2010, p. 4)¹⁸.

Entendemos que esse movimento em torno do pensamento computacional é ainda um campo em construção, avaliado ainda de maneira pouco crítica. Nosso objetivo é que este artigo possa servir como (mais) um ponto de ancoragem para uma análise mais bem fundamentada. Ao contrário do que ocorreu nos primórdios da área de informática na educação, no contexto atual em que se insere o pensamento computacional, existe um campo de batalha que envolve questões econômicas e geopolíticas que resultam em diferentes metas para a educação.

7. Referências

AHO, A. V., Ubiquity symposium: Computation and computational thinking. **Ubiquity**, vol. 2011, no. January, 2011.

ALLINGTON, D.; BROUILLETTE, S.; GOLUMBIA, D. **Neoliberal Tools (and Archives)**. A Political History of Digital Humanities. Los Angeles, CA: LA Review of Books, 2016
Disponível em: <<https://lareviewofbooks.org/article/neoliberal-tools-archives-political-history-digital-humanities/>>. Acesso em: 19 set. 2017.

AMIEL, T.; FEDEL, G. S.; ARANTES, F. L.; AGUADO, A. G.; Dominando para não ser dominado: Autonomia tecnológica com o Projeto Jovem Hacker, 07/2015, **16º Workshop Internacional de Software Livre (WSL)**, evento realizado em conjunto com o Fórum Internacional de Software Livre (FISL 16), pp.1-13, Porto Alegre, RS, BRASIL, 2015.

¹⁸ Do original em inglês: “In the end, though, perhaps we should talk less about computational thinking, and focus more on computational doing — carrying out one’s work (and one’s play!) in new ways by using computational tools” (Hemmendinger, 2010, p. 4).

BRASIL. Ministério da Educação. **Base Nacional Comum Curricular**. Proposta preliminar. Terceira versão revista. Brasília: MEC, 2016. Disponível em: <http://basenacionalcomum.mec.gov.br/images/pdf/4.2_BNCC-Final_MA.pdf>. Acesso em: 29 mai. 2017.

COLLEGE BOARD. **AP Computer**: Science Principles Including the Curriculum Framework. New York: College Board, 2016.

COURSEREPORT Find the Best Bootcamp for You. **CourseReport**. Disponível em: <<https://www.coursereport.com>>. Acesso em 12 ago. 2017.

CSTA - Computer Science Teachers Association. **Operational Definition of Computational Thinking - for K–12 Education**. 2011, pp. 1-1. Disponível em <<https://csta.acm.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf>>. Acesso em: 12 ago. 2017.

ISTE. **National Educational Technology Standards (NETS•S)** and Performance Indicators for Students. Washington: ISTE, 2007. Disponível em: <<http://www.iste.org>>. Acesso em: 12 ago. 2017.

ITEA. **Standards for technology literacy**: Content for the study of technology. Reston, Virginia: International Technology Education Association, 2007. Disponível em: <<http://www.iteaconnect.org/>>. Acesso em: 12 ago. 2017.

FARAG, B. Please don't learn to code. 2016. **Crunch Network**. Disponível em: <<https://techcrunch.com/2016/05/10/please-dont-learn-to-code/>>. Acesso em: 15 set. 2017.

FERNANDES R. K.; SAVIOLI, A. M. P. D. Características de pensamento algébrico manifestadas por estudantes do 5º ano do ensino fundamental. **Revista Paranaense de Educação Matemática - RPEM**, Campo Mourão, PR, v.5, n.8, p.131-151, jan.-jun. 2016. Disponível em: <http://www.fecilcam.br/revista/index.php/rpem/article/viewFile/1228/pdf_168>. Acesso em: 29 ago. 2017.

FNDE MEC Audiência Pública 04/2017 - Robótica Educacional. 2017. **YouTube**. Disponível em: <<https://www.youtube.com/watch?v=8crrMWWYOQs>>. Acesso em: 12 ago. 2017.

GADANIDIS, G.; HUGHES, J.M.; MINNITI, L.; WHITE, B. Computational Thinking, Grade 1 Students and the Binomial Theorem. **Digital Experiences in Mathematics Education**. 2016. Disponível em: <doi 10.1007/s40751-016-0019-3>. Acesso em: 03 ago. 2017.

GARDNER, H. **The Mind's New Science**: a history of the cognitive revolution. New York: Basic Books, 1985.

HEMMENDINGER, D. A plea for modesty. **ACM Inroads**, v. 1, n. 2, p. 4–7, 2010. Pre-print available at: <http://athena.union.edu/~hemmendd/History/ct.pdf>. Acesso em: 19 ago. 2017.

HOYLES, C. Apresentação feita no **Constructionism 2016, Session 19, Plenary 5**, 2016. Disponível em: <<https://www.youtube.com/watch?v=QpcqFsYpWt4&feature=youtu.be>>. Acesso em: 03 ago. 2017.

HOYLES, C. **Computational Thinking in Mathematics Education**. Disponível em: <<http://ctmath.ca/partners/celia-hoyles>>. Acesso em: 04 ago. 2017.

K12CS Site **K–12 Computer Science Framework**. Disponível em: <<https://k12cs.org/>>. Acesso em: 12 ago. 2017.

MARQUES, F. A realidade que emerge da avalanche de dados. **Pesquisa FAPESP** n. 355, 2017. Disponível em: <<http://revistapesquisa.fapesp.br/2017/05/23/a-realidade-que-emerge-da-avalanche-de-dados/>>. Acesso em: 19 set. 2017.

NORVIG, P. **Teach yourself programming in ten years**. 2014. Disponível em: <<http://norvig.com/21-days.html>>. Acesso em: 29 ago. 2017

PAPERT, S. **Mindstorms: Children, Computers, and Powerful Ideas**. New York: Basic Books, 1980.

PROULX, S. Trajetórias de uso das tecnologias de comunicação: as formas de apropriação da cultura digital como desafios de uma 'sociedade do conhecimento'. **Trabalhos em Linguística Aplicada**, vol. 49 n. 2 Campinas July/Dec. 2010. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-18132010000200008>. Acesso em: 29 ago. 2017.

SINGER, N. How Google Took Over the Classroom. **New York Times**, 13 maio 2017. Disponível em: <<https://www.nytimes.com/2017/05/13/technology/google-education-chromebooks-schools.html>>. Acesso em: 19 set. 2017.

TEDRE, M.; DENNING, P. J. The long quest for computational thinking. In: Proceedings of the 16th Koli Calling Conference on Computing Education Research, **Anais...**2016. Disponível em: <<http://denninginstitute.com/pjd/PUBS/long-quest-ct.pdf>>. Acesso em: 19 set. 2017.

TURKLE, S.; PAPERT, S. Epistemological Pluralism: Styles and Voices within the Computer Culture. **Signs: Journal of Women in Culture and Society**, v. 16, n. 1, p. 128–157, 1990.

UNESCO **Padrões de competência em TIC para professores**. 2009. Disponível em: <<http://cst.unesco-ci.org/sites/projects/cst/default.aspx>>. Acesso em: 12 ago. 2017.

VALENTE, J. A. Integração do pensamento computacional no currículo da educação básica: diferentes estratégias usadas e questões de formação de professores e avaliação do aluno. **Revista e-Curriculum**, São Paulo, v.14, n.03, p. 864 – 897 jul./set.2016. Disponível em: <<https://revistas.pucsp.br//index.php/curriculum/article/view/29051>>. Acesso em: 23 jun. 2017.

VARELA, F. J.; THOMPSON, E.; ROSCH, E. **The Embodied Mind: cognitive science and human experience**. Cambridge, MA: MIT Press, 1993.

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

ZAPATA-ROS, M. Pensamiento computacional: Una nueva alfabetización digital. **Revista de Educación a Distancia**, v. 46, n. 4, p. 1-47, 2015. Disponível em: <<http://www.um.es/ead/red/46/zapata.pdf>>. Acesso em: 05 mai. 2017.