http://journal.unoeste.br/index.php/ch
DOI: 10.5747/ch.2021.v18.h512
ISSN on-line 1809-8207

Colloquium Humanarum

Submetido: 17/02/2021 Correções: 07/06/2021 Aceite Final: 10/06/2021

ENSINO DE PROGRAMAÇÃO PARA CRIANÇAS E O DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL: ALGUMAS REFLEXÕES

Aletheia Machado de Oliveira

Doutoranda do Programa de Pós-Graduação em Educação, Faculdade de Ciências e Tecnologia, UNESP/Campus Presidente Prudente. Professora de informática na rede municipal de Juiz de Fora. **ORCID iD:** https://orcid.org/0000-0002-1058-0817. **E-mail:** aletheiaoliveirajf@gmail.com

RESUMO

Um dos grandes desafios enfrentados pela educação é o de refletir sobre os usos efetivos das tecnologias digitais da informação e comunicação, de modo a proporcionar aos estudantes uma aprendizagem contextualizada e significativa. Diante disso, o objetivo do artigo é refletir sobre os usos efetivos das tecnologias digitais da informação e comunicação na educação nestes tempos tecnológicos, para, assim, alcançar outras formas de fazer educação. O ensino da programação pode gerar estímulos ao desenvolvimento do pensamento computacional, permitindo que a criança possa vir a ser construtora de artefatos digitais. Ressaltamos, assim, que nosso estudo é de caráter teórico reflexivo, constituído por pesquisa bibliográfica, com contribuições de Castells (1999), Coll e Monereo (2010), Papert (1994), Selwyn (2017), dentre outros estudiosos. Apresentamos, em um primeiro momento, uma explanação sobre a abordagem construcionista e a educação, com base nos estudos do seu idealizador Seymour Papert. Em um segundo momento, refletimos sobre o ensino de programação para crianças para, em seguida, contextualizar o pensamento computacional e as estratégias de implementação adotadas. A partir desse estudo, vislumbramos um horizonte de possibilidades para a necessária mudança de paradigma na educação.

Palavras-chave: Crianças. Pensamento Computacional. Ensino de Programação.

TEACHING PROGRAMMING TO CHILDREN AND THE DEVELOPMENT OF COMPUTATIONAL THINKING: SOME REFLECTIONS

ABSTRACT

One of the great challenges faced by education is to reflect on the effective uses of digital information and communication technologies, in order to provide students with contextualized and meaningful learning. Therefore, the objective of the article is to reflect on the effective uses of digital information and communication technologies in education in these technological times, to achieve other ways of providing education. The teaching of programming can generate stimuli for the development of computational thinking, allowing the child to become a builder of digital artifacts. We emphasize, therefore, that our study is theoretical and reflective, consisting of bibliographical research, with contributions from Castells (1999), Coll and Monereo (2010), Papert (1994), Selwyn (2017), among other scholars. We present, at first, an explanation about the constructionist approach and education, based on the studies of its creator Seymour Papert. In a second moment, we reflect on teaching programming to children to then contextualize the computational thinking and implementation strategies adopted. From this study, we glimpse a horizon of possibilities for the necessary paradigm shift in education.

Keywords: Children. Computational Thinking. Programming Teaching.

ENSEÑAR PROGRAMACIÓN A NIÑOS Y DESARROLLO DEL PENSAMIENTO COMPUTACIONAL: ALGUNAS REFLEXIONES

RESUMEN
Uno de los grandes desafíos que enfrenta la educación es reflexionar sobre los usos efectivos de las tecnologías de la información y la comunicación digitales, con el fin de brindar a los estudiantes un aprendizaje contextualizado y significativo. Por tanto, el objetivo del artículo es reflexionar sobre los usos efectivos de las tecnologías de la información y la comunicación digitales en la educación en estos tiempos tecnológicos, para llegar a otras formas de educación. La enseñanza de la programación puede generar estímulos para el desarrollo del pensamiento computacional, permitiendo que el niño se convierta en un constructor de artefactos digitales. Destacamos, por tanto, que nuestro estudio tiene un carácter teórico reflexivo, consistente en una investigación bibliográfica, con aportes de Castells (1999), Coll y Monereo (2010), Papert (1994), Selwyn (2017), entre otros estudiosos. Presentamos, en un primer momento, una explicación sobre el enfoque y la educación construccionista, basada en los estudios de su creador Seymour Papert. En un segundo momento, reflexionamos sobre la enseñanza de la programación a los niños para luego contextualizar el pensamiento computacional y las estrategias de implementación adoptadas. A partir de este estudio, vislumbramos un horizonte de posibilidades para el necesario cambio de paradigma en educación.
Palabras-clave: Niños. Pensamiento computacional. Enseñanza de la programación.

1. INTRODUÇÃO

As tecnologias digitais da informação e comunicação (TDIC) estão presentes em várias áreas da sociedade, sendo utilizadas no trabalho, no lazer, na comunicação, nos serviços em geral e na educação.

De acordo com Castells (1999, p. 69), o que caracteriza sua presença é a aplicação de conhecimentos e informações em um "[...] ciclo de realimentação cumulativo entre a inovação e o uso", em que as tecnologias devem ser vistas não como "[...] ferramentas a serem aplicadas, mas processos a serem desenvolvidos". Assim, elas vão se tornando "[...] amplificadores e extensões da mente humana", alterando o modo como as pessoas vivem, trabalham, informam-se, relacionam-se, comunicam-se, aprendem, consomem e produzem.

Kenski (2012) comunga desse entendimento ao pontuar que as TDIC servem para, além de informar e comunicar, interagir, interligar pessoas e organizações e viabilizar o processo educacional, pois elas são os meios pelos quais as pessoas fazem a educação hoje. É interessante notar que a referida autora também aponta que as TDIC "[...] interferem em nosso modo de pensar, sentir, agir, de nos relacionarmos socialmente e adquirirmos conhecimentos. Criam uma nova cultura e num novo modelo de sociedade" e, nesse sentido, para os usuários que a usam de modo frequente, "[...] não são mais vistas como tecnologias, mas como complementos, como companhias, como continuação de seu espaço de vida" (KENSKI, 2012, p. 23-25).

Ao discorrer sobre as modificações e efeitos das tecnologias na vida cotidiana das pessoas e nas distintas áreas da sociedade, Coll e Monereo (2010) defendem a ideia de que as TDIC fazem parte de um novo paradigma tecnológico que modifica as práticas sociais e, em especial, as práticas educacionais neste começo do século XXI, nos modos de ensinar e aprender. O impacto das TDIC na educação deve ser entendido, de acordo com os referidos autores, "[…] como um aspecto particular de um fenômeno mais amplo relacionado com o papel das tecnologias [...]" na sociedade contemporânea (COLL; MONEREO, 2010, p. 15).

Nesse processo, um dos grandes desafios enfrentados pela educação é o de refletir sobre os usos efetivos, a fim de modificar a relação dos estudantes com as tecnologias digitais - de consumidores passivos para produtores de conteúdo. E, um dos caminhos que podemos trilhar para que tais mudanças aconteçam está no ensino de programação para crianças. Por isso, é fundamental propiciar, desde muito cedo, oportunidades para que elas reflitam, testem, descubram, questionem, construam e também compartilhem projetos elaborados por elas mesmas.

De acordo com Selwyn (2017), vivemos em uma era em que instituições de ensino, tanto da educação básica quanto da superior, estão "[…] inundados de artefatos, plataformas e aplicativos digitais, o que torna impossível imaginar o futuro da educação sem as

tecnologias computadorizadas em posição de centralidade" (SELWYN, 2017, p. 87).

Nesse sentido, este artigo propõe refletir sobre os usos efetivos das TDIC na educação nestes tempos tecnológicos, vislumbrando outras formas de fazer educação, considerando a possibilidade de a criança, sob o estímulo ao desenvolvimento do pensamento computacional, ser construtora de artefatos digitais por meio do ensino de programação.

Para tanto, o estudo proposto possui mais três seções que se complementam, além desta, que é introdutória. A próxima seção discute a abordagem construcionista e a educação, com base nos estudos do seu idealizador Seymour Papert. Na seção 3 será abordado o ensino de programação para crianças por meio da linguagem de programação visual *Scratch* como possibilidade pedagógica no trabalho com as crianças. A seção 4 buscará contextualizar o pensamento computacional e as estratégias de implementação adotadas para estabelecermos relação com a educação. Por fim, na seção 5, serão apresentadas as considerações finais.

2. CONSTRUCIONISMO E EDUCAÇÃO: ALGUMAS REFLEXÕES

Inspirados nas discussões da abordagem construcionista de Seymour Papert, entendemos que a idealização da criança como construtora de artefatos digitais não é uma novidade. Papert já afirmava que as crianças devem ser criadoras em vez de consumidoras de conhecimento. Logo, a visão de escola apresentada pelo autor como um sistema educacional que impõe um único modo de saber, com um único estilo "[...] moldando a criança no papel de receptor passivo de conhecimento [...]", fazendo com que pareça às crianças e jovens "[...] como lenta, maçante e francamente fora de sintonia" é bastante criticada (PAPERT, 1994, p. 20).

Tendo isso em vista, novas exigências se desenham para a escola convidando-a a rever suas concepções de ensino. A partir de tal situação, há duas abordagens que o referido autor discute em seu livro sobre os usos de computadores e que estão ligadas ao uso das TDIC na aprendizagem: instrucionismo e construcionismo. Na abordagem instrucionista, temos uma criança que responde às perguntas do computador com o intuito de verificar se a informação foi retida por ela. Esta é uma abordagem bastante criticada por Papert, que questiona o princípio de que a ação de ensinar está pautado na instrução pela transmissão da informação, mecanização repetitiva, "dar o peixe sem questionar" ao aluno. Nessa abordagem, baseada num nível mais programático ou ideológico, determina-se o que ensinar ao aluno. Logo, a aprendizagem é mecânica e a única forma de melhorar o conhecimento é ensinar por instrução direta. O computador é usado apenas para transmitir informações na forma de instrução auxiliada (PAPERT, 1994).

A abordagem construcionista, termo cunhado por Papert a partir das ideias do construtivismo de Jean Piaget, o autor afirma que a metodologia representa "[...] uma alternativa às outras metodologias favorecidas pela escola de pensamento 'científico' dominante", a qual é constituída com o uso de aplicações computacionais na educação como um meio para permitir às crianças engajarem-se em atividades significativas e mais próximas da sua realidade (PAPERT, 1994, p. 31).

Essa abordagem fundamenta-se numa perspectiva de aprendizagem ativa e reflexiva, em que a criança constrói o próprio conhecimento por intermédio do computador. Essa construção é entendida por meio do fazer algo, ou seja, o aprendizado por meio do fazer e que seja do interesse significativo para as crianças. Cumpre lembrarmos, nesse ínterim, que o autor não coloca em dúvida o valor da instrução, mas a forma como ela é trabalhada no contexto escolar com ênfase ao conhecimento formal-abstrato, impedindo a aprendizagem efetiva e o maior progresso da educação. Nesses termos, por meio de uma metodologia construtiva, que o aluno se desenvolverá melhor descobrindo por si mesmo o conhecimento específico de que necessita (PAPERT, 1994).

A inspiração de Seymour Papert (1994, p. 148) ao criar uma linguagem de programação de fácil compreensão, denominada Logo, visando a colocar em prática os princípios construcionistas por meio dos computadores, é a de que "eles deveriam servir às crianças como instrumentos com os quais trabalhar e pensar, como meios para realizar projetos, como fonte de conceitos para pensar novas ideias". Logo, os computadores "[…] não apenas melhorariam a aprendizagem escolar, mas apoiariam formas diferentes de pensar e aprender" (PAPERT, 1994, p. 156).

Vieira, Campos e Raabe (2020), ao apresentarem um resumo histórico do legado de Seymour Papert e dos projetos com a linguagem

Logo realizados no Brasil, apontam que ele é considerado um dos autores fundamentais das TDIC na educação, principalmente quanto ao uso de computadores na aprendizagem. Ao criar a Logo, a ideia era o uso diferente do computador na aprendizagem, em que o aprendiz se torna sujeito ativo, protagonista e responsável sobre a sua própria aprendizagem.

Na perspectiva do matemático e educador, a tecnologia é vista como uma ferramenta modificadora da aprendizagem. Para isso, Papert se questionou sobre como criar condições para que a criança pudesse construir conhecimento. Assim, o construcionismo propõe que o uso significativo dos computadores está em "[…] adotar uma estratégia que visa estimular o pensamento de forma criativa, possibilitando aprender por meio da curiosidade e do envolvimento na atividade que está realizando" (GLIZT, 2020, p. 202).

Papert (1994) observou que os processos de aprendizagem tradicionais enfatizam a instrução pelo professor para, depois, a criança poder criar abstrações. Nessa perspectiva, ela retém, decora, ou guarda na memória as instruções para depois criar. A preocupação está em acelerar, consolidar o caminho da criança mais à frente de operações concretas. Todavia, para que o aprendizado efetivamente se consolide, alterando perspectivas mais significativas de educação, o mencionado autor diz que o melhor é fortalecer e sustentar o processo concreto em todas as idades. A criança terá a oportunidade, por exemplo, de fazer cálculos pela via do computador, para depois entender o que foi feito no quadro-negro ou papel na sala de aula. Por isso, o computador é visto por Papert como uma importante ferramenta para a que a criança desenvolva o gosto pela aprendizagem e o processo de construção do conhecimento.

Cabe destacar que essa contraposição entre instrucionismo e construcionismo é também discutida por Valente (1997) para o uso do computador no âmbito da educação e que igualmente pode estar interligada ao uso e à criação de artefatos digitais no contexto educacional. Para esse autor é possível empregar o computador como máquina de ensinar (informatização dos métodos de ensino tradicionais) ou como máquina para ser ensinada (construção de um objeto de interesse da criança, como uma obra de arte, um programa de computador ou jogos digitais).

Nessa vertente, é preciso destacar a ação dos professores. Para se tornarem instigadores, ajudantes e orientadores (JONASSEN, 2000), isto é, agentes de aprendizagem (VALENTE, 2005) ou consultores de ideias, bem como orientadores ou mediadores (COLL; MONEREO, 2010) é essencial que eles compreendam o uso das TDIC como ferramentas para a criação de ambientes que encorajem as crianças a produzir conhecimento. Tal compreensão, contudo, requer que a formação docente (inicial e continuada) deve estar baseada no uso crítico e reflexivo das TDIC e integrada ao seu processo formativo de forma prática e teórica, e não simplesmente baseada na preparação técnica, "[…] sem conectividade com a realidade local de cada escola […]", terminando por desqualificar "[…] a própria ideia de uso das tecnologias digitais como elementos estruturantes da cultura digital" (BONILLA; PRETTO, 2015, p. 508).

Assim, entendemos que a proposta de Seymour Papert e o olhar diferenciado para a aprendizagem nos mostra o quanto é importante a ação do aprendiz quando se utiliza as TDIC, assim como é imprescindível a postura encorajadora do professor. Para muito além do acesso e consumo de tecnologia, é fundamental propiciar, desde muito cedo, oportunidades para que as crianças reflitam, testem, descubram, questionem, construam e compartilhem projetos elaborados por elas mesmas, pois é preciso "[…] aprender não apenas a interagir com o computador, mas também a criar com ele" (SOBREIRA; TAKINAMI; SANTOS, 2013, p. 136).

Dessa forma, entendemos que, por intermédio do computador, a criança tem a possibilidade de construir artefatos digitais de seu interesse. Tal perspectiva serve de estímulo ao desenvolvimento do pensamento computacional por meio do ensino de programação, assunto que discutiremos nas próximas seções.

3. O ENSINO DE PROGRAMAÇÃO PARA CRIANÇAS

O ensino de programação não é uma prática restrita apenas à área da Computação ou a pessoas que se interessam em seguir na área de tecnologia, mas envolve a todos. Tais práticas ajudam "[…] a desenvolver habilidades para auxiliar na resolução de problemas, ser um elemento ativo na construção do conhecimento e entender o contexto tecnológico em que vivemos", sendo possível encontrar experiências

na educação básica (ZANETTI; BORGES; RICARTE, 2016, p. 21).

Exemplificando tais discussões sobre o trabalho com o ensino de programação desde a infância, trazemos como referência o livro de Raabe, Zorzo e Blikstein (2020) composto por uma série de artigos que discutem acerca da importância do ensino de programação para o desenvolvimento do pensamento computacional. O livro está organizado em três partes, a saber: fundamentos, pesquisas e relatos de experiência. Na segunda parte, os artigos relatam como o pensamento computacional pode ser empregado na educação básica, incluindo aspectos do ensino de programação, uso de jogos digitais, clubes de computação e berçários *hackers*. Na terceira parte, os relatos de experiência envolvendo o ensino de programação estão organizados conforme as etapas da educação básica (educação infantil, anos iniciais do ensino fundamental, anos finais do ensino fundamental e ensino médio) e retratam iniciativas adotadas em várias regiões brasileiras para estimular professores e instituições escolares que pretendem adotar o trabalho com programação na educação básica.

Outros trabalhos que vêm mostrando a importância de apoiar o ensino de programação na educação básica podem ser encontrados em eventos da Sociedade Brasileira de Computação (SBC), como o Congresso Brasileiro de Informática na Educação (CBIE), e nos eventos paralelos integrantes do CBIE, em especial Simpósio Brasileiro de Informática na Educação (SBIE), Workshop de Informática na Escola (WIE) e Workshop de Ensino em Pensamento Computacional, Algoritmos e Programação (WalgProg); e o Simpósio Brasileiro de Educação em Computação (EduComp).

O crescimento dessa abordagem bem como a sua popularização se dão porque o desenvolvimento do pensamento computacional por meio do ensino de programação vem demonstrando resultados positivos, pois estimula o aprendizado de programação e a inventividade e criatividade, assim como possibilita o compartilhamento do que foi criado para além do espaço escolar, oportunizando o entrelaçamento com conteúdos trabalhados nos componentes curriculares de cada área do conhecimento e encorajando o trabalho colaborativo/cooperativo ao favorecer "[…] práticas em que o uso do computador está voltado para a construção do conhecimento" (SOBREIRA; VIVEIRO; D'ABREU, 2020, p. 29).

Dentre as práticas pedagógicas relacionadas ao uso do pensamento computacional em ensino de programação, cinco categorias foram definidas por Zanetti, Borges e Ricarte (2016, p. 26), conforme quadro abaixo, com sua respectiva descrição.

Quadro 1. Categorias de práticas pedagógicas

Categoria	Descrição
Computação "Desplugada" (CD)	Métodos que procuram promover o ensino de computação sem o uso de computador, utilizando atividades lúdicas para atingir pessoas de todas as idades.
Jogos Digitais (JD)	Utilização de jogos digitais para o ensino de conceitos de computação ou programação de computadores.
Linguagem de Programação (LP)	Utilização de linguagens de programação de alto nível (linguagem C) ou orientada a objetos (*Java* ou *Python*).
Linguagem de Programação Visual (LPV)	Utilização de linguagens de programação visual, como *Scratch* ou *App Inventor*.
Robótica Pedagógica (RP)	Utilização de artefatos robóticos, reais ou virtuais, como ferramenta para de ensino de programação.

Fonte: Zanetti, Borges e Ricarte (2016).

Observamos a partir desse quadro que são várias as práticas pedagógicas que podem ser aplicadas junto às crianças no ensino de programação. Devemos, porém, antes analisar a faixa etária de cada criança para que tais práticas possam ser aplicadas, assim como é necessário conhecer previamente cada uma para que o trabalho seja positivo. Alinhar aos conteúdos trabalhados aos temas que sejam de interesse das crianças e construir objetivos bem definidos são aspectos que também devem ser considerados para o sucesso do trabalho com programação.

Nesse artigo vamos apoiar nossas discussões na segunda e quarta categoria, pois os jogos digitais (segunda categoria) são usados no cotidiano das crianças e podem auxiliá-las na construção do conhecimento. Outro aspecto é que as linguagens de programação visual (quarta categoria) são próprias para o trabalho com esse público por possibilitarem às crianças atividades de construção de jogos digitais, para que aprendam conceitos, valores e atitudes referentes aos diferentes conteúdos escolares.

Em relação ao uso de jogos digitais, temos uma grande variedade de jogos disponibilizados via internet e que podem ser usados no ensino de programação, dentre eles o *Code Baymax®*[1], *Code Monkey®*[2], *Code Studio®*[3], *Lightbo®t*[4], *Ninja Shape®*[5], *The Foos®*[6] *e Kodable®*[7]. No que tange às linguagens de programação visual, desde as primeiras experiências com a Logo[8], temos o uso de ferramentas de linguagem de programação visual em que a construção de artefatos ocorre por meio de elementos gráficos como o *Scratch Jr.®*[9], *Scratch®*[10], *Kodu®*[11], *MIT APP Inventor®*[12], dentre outros.

Segundo Furini, Teixeira e Trentin (2020), as linguagens de programação visual Logo e *Scratch* trouxeram grandes avanços em relação a pensar em uma educação que encontra na tecnologia um meio potencial de motivar, de inventar, de buscar e de resolver problemas por meio dessas ferramentas tecnológicas.

Reforçando tal assertiva, temos os estudos de Sobreira, Takinami e Santos (2013, p. 136) que apontam seis possibilidades pedagógicas como o uso do *Scratch*, "[…] o que contribui efetivamente para a promoção da contextualização curricular e da atribuição de significados aos conteúdos programáticos". Dentre essas possibilidades, os autores destacam as animações, jogos de tabuleiro, jogos digitais, *games* literários, *quiz* e comunicação com outras interfaces (*Android, Arduino* e *Makey Makey*). Na prática, o contato com o *Scratch* possibilita uma aprendizagem colaborativa, interativa, construtiva e significativa, com foco no protagonismo do aluno.

Por ser de fácil manuseio, não é necessário digitar nenhum comando complicado. Apenas conectar os blocos ou pegar e largar (*drag and drop*) numa perspectiva interativa e amigável. De acordo com Marji (2014) e Jesus, Vasconcelos e Lima (2016), a utilização dessa ferramenta, por um lado, contém um grupo diverso de funcionalidades que desenvolvem a criatividade dos usuários, e, por outro, representa um meio de aprendizagem.

O *Scratch* é uma linguagem de programação visual criada a partir da linguagem Logo (de Seymour Papert). Foi desenvolvido no *Massachusetts Institute of Technology* (MIT), em 2007, pela equipe do *Lifelong Kindergarten*, coordenada por Mitchel Resnick. Tem código aberto, podendo ser instalado no computador (versão *offline)* ou utilizado a partir de um navegador (versão *online*)[1]. É disponibilizado gratuitamente e funciona nos principais sistemas operacionais (Mac, Windows e Linux) (SOBREIRA; TAKANAMI; SANTOS, 2013).

Projetado para ser usado para as idades entre 08 e 16 anos e por outras pessoas de diferentes idades, atualmente encontra-se nas versões 1.4 e 2.0, sendo que os projetos do *Scratch* 2.0 têm extensão de arquivo *.sb2* para diferenciá-los dos projetos elaborados com a versão 1.4, que é *.sb.* Pode ser utilizado em várias áreas do conhecimento em articulação com os componentes curriculares. Essa linguagem de programação visual tem por base a metáfora dos legos (construção com peças; neste caso blocos). Após a criação de jogos digitais, histórias

[1] Página oficial: https://www.superjogosclick.com.br/jogos/operacao-big-hero-code-baymax/
[2] Página oficial: https://www.codemonkey.com/
[3] Página oficial: https://code.org
[4] Página oficial: https://lightbot.com/hocflash.html
[5] Página oficial: http://www.bigdino.com/game/1511/Ninja-Shape
[6] Página oficial: https://codespark.com/
[7] Página oficial: https://www.kodable.com/hour-of-code
[8] Página oficial: https://www.nied.unicamp.br/biblioteca/super-logo-3-0-para-windows-7-a-10/
[9] Página oficial: https://jfo8000.github.io/ScratchJr-Desktop/
[10] Página oficial: http://scratch.mit.edu
[11] Página oficial: http://www.kodugamelab.com/
[12] Página oficial: https://appinventor.mit.edu/

animadas, tutorais, entre outros, estes poderão ser disponibilizados na *Web* (*online*) de modo a partilhar com todo o público em geral o resultado do que foi criado (MARJI, 2014; JESUS; VASCONCELOS; LIMA, 2016).

Dito isso, entendemos que o trabalho com programação com uso do *Scratch* e demais linguagens de programação visual, de acordo com as discussões dos autores ora mencionados, possibilita que os alunos criem projetos mais próximos da sua realidade e permite o protagonismo quando se utiliza as TDIC.

4. EXPLORANDO O PENSAMENTO COMPUTACIONAL POR MEIO DA CRIAÇÃO DE ARTEFATOS DIGITAIS

Comecemos por compreender a expressão pensamento computacional, pois a sua delimitação ainda não é feita em termos precisos e objetivos, sendo investigado por diversos pesquisadores nacionais e internacionais (BRACKMANN, 2017; VIEIRA; SANTANA; RAABE, 2017; VALENTE, 2016, 2019; WING, 2006, 2008, 2010, 2014). Em nível conceitual, Jeannette Wing popularizou essa expressão por meio de um artigo publicado em uma revista influente no âmbito acadêmico da Computação *Communications of the ACM*, em 2006. No ano de 2016, foi traduzido para o português e publicado na Revista Brasileira de Ensino de Ciência e Tecnologia.

De acordo com Brackmann (2017), Wing conceituou o termo de mais de uma forma. Em 2006, descreve como uma "[...] habilidade fundamental para todos, não somente para cientistas da computação" (WING, 2006, p.02). Em uma publicação no ano de 2008, explica que "[...] é uma abordagem para resolver problemas, projetando sistemas e compreensão do comportamento humano que se baseia em conceitos fundamentais para computação" (WING, 2008, p. 3717). Já no ano de 2010, conceitua como "[...] processos de pensamento envolvidos na formulação de problemas e suas soluções para que as soluções sejam representadas de uma forma que possam ser realizadas por um agente de processamento de informações" (WING, 2010, p. 01). Por sua vez, em 2014, publicou outro artigo com o objetivo de apresentar uma definição mais clara em que o pensamento computacional é descrito como "[...] o processo de pensamento envolvido na formulação de um problema e na expressão de sua(s) solução(ões) de forma que um computador – homem ou máquina- possa executar com eficácia" (WING, 2014, s.p.).

Segundo Valente (2019), as ideias de Wing são criticadas pelo fato de ela explicar o termo em estreita relação com a Ciência da Computação e não reconhecer trabalhos anteriores que discutem e estudam os usos das TDIC sobre novas maneiras de aprender a aprender, como, por exemplo, os estudos do professor Seymour Papert, que desenvolveu, a Logo, uma linguagem de programação para crianças.

Os autores Vieira, Santana e Raabe (2017) reforçam essa afirmativa ao dizerem que o termo, apesar de ter sido popularizado, de ser apontado como uma nova forma para o uso da tecnologia nas escolas e de melhorar a capacidade dos estudantes em resolver problemas usando conhecimentos da Computação, já vinha sendo empregado desde a década de 1980 por meio de amplo empreendimento de uso da Logo em várias escolas brasileiras por José Armando Valente e Léa da Cruz Fagundes, pesquisadores e precursores da Logo no Brasil.

No ano de 2011, organizações americanas propuseram uma definição operacional para o termo. Em parceria com pesquisadores da Ciência da Computação e das áreas de Humanas, as organizações *National Science Foundation* (NSF), *International Society for Technology in Education* (ISTE) e a *American Computer Science Teachers Associatino* (CSTA) compreendem-no como um processo de resolução de problemas que inclui as seguintes características:

- Formular problemas de uma forma que nos permita usar o computador e outras ferramentas para ajudar a resolvê-los;
- Organizar e analisar dados de forma lógica;
- Representar dados por meio de abstrações, como modelos e simulações;
- Automatizar soluções por meio do pensamento algorítmico (etapas ordenadas);
- Identificar, analisar e implementar soluções possíveis como o objetivo de alcançar de forma eficiente a combinação de etapas e recursos;

> • Generalizar e transferir esse processo de resolução de problemas para uma ampla variedade de problemas (ISTE/CSTA, 2011, p. 07).

Além dessas características, um agrupamento de atitudes pode ser elencado nas dimensões do pensamento computacional, o qual inclui: confiança, persistência, tolerância, capacidade de lidar com problemas em aberto e capacidade de se comunicar e trabalhar com outras pessoas, podendo ser empregada e desenvolvida em todas as disciplinas (ISTE/CSTA, 2011; VALENTE, 2019).

No contexto brasileiro, podemos citar a tese de Brackmann (2017), que propõe uma definição sobre essa expressão, assim como o documento *Referenciais de Formação em Computação: Educação Básica da Sociedade Brasileira de Computação (SBC),* de 2017, que também traz um entendimento sobre o termo e os artigos de Ribeiro, Foss e Cavalheiro (2020) e Pinto e Nascimento (2018).

A tese de Brackmann (2017) discute a possibilidade de desenvolver o pensamento computacional na educação básica utilizando exclusivamente atividades desplugadas. Nela se discute o que é o pensamento computacional, seus pilares, importância da adoção na educação básica, panorama mundial do estado da arte do pensamento computacional nas escolas, incluindo iniciativas públicas e privadas, pesquisas recentes com o estado da arte da abordagem desplugada e a avaliação dos estudantes. Assim, o autor propõe uma definição para o termo.

> O Pensamento Computacional é uma distinta capacidade criativa, crítica e estratégica humana de saber utilizar os fundamentos da Computação, nas mais diversas áreas do conhecimento, com a finalidade de identificar e resolver problemas, de maneira individual ou colaborativa, através de passos claros, de tal forma que uma pessoa ou máquina possa executá-los eficazmente (BRACKMANN, 2017, p.29).

Já o documento *Referenciais de Formação em Computação: Educação Básica da Sociedade Brasileira de Computação (SBC)* apresenta uma proposta de implementação do ensino de Computação, enfatizando competências e habilidades dos principais eixos que compõem a área e como podem ser trabalhados ao longo da educação básica, da educação infantil e do ensino médio (SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 2017).

Nesse sentido, o documento discute o que é Computação, apresenta os três eixos de ensino de Computação (Pensamento Computacional, Mundo Digital e Cultura Digital) e traz as competências e habilidades a serem desenvolvidas em cada eixo por etapa escolar. A Computação, segundo o documento, é uma área nova e envolve "[...] tanto técnicas de resolução e análise de problemas quanto a construção e uso de máquinas para auxiliar na execução das soluções", podendo assumir distintos pontos de vista: uma ciência que investiga a resolução de problemas; uma ciência que proporciona a criação de um mundo novo; e uma ciência que muda radicalmente o comportamento da sociedade (SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 2017, p. 02-03).

Atendo-se ao eixo pensamento computacional, o documento o define como "[...] a capacidade de sistematizar, representar, analisar e resolver problemas" (SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 2017, p. 03). Os pilares do pensamento computacional desenvolvem as seguintes competências: abstração, que é capacidade de extrair apenas as características importantes de um problema para descrever informações e processos, e técnicas para construir soluções algorítmicas; automação, que é a capacidade de descrever as soluções por meio de algoritmos utilizando-se de um meio eletrônico que possibilitem executar partes ou todo o algoritmo proposto; e análise, que é capacidade de analisar criticamente os problemas e soluções para identificar não somente os resultados gerados na automação, mas também ser capaz de avaliar a eficiência e a correção destas soluções. Em relação às habilidades, o documento apresenta-as para cada etapa da educação básica, dividida pelos eixos elencados anteriormente. Nos anos iniciais do ensino fundamental, as habilidades relacionadas ao pensamento computacional são:

> - Representar em experiências concretas as principais abstrações para descrever dados: registros, listas, grafos, imagens.
> - Identificar as principais abstrações para construir processos: escolha, composição e repetição, simulando e definindo algoritmos simples que representem situações do cotidiano infantil; e
> - Utilizar linguagem lúdica visual para representar algoritmos (SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 2017, p.07).

Ribeiro, Foss e Cavalheiro (2020) também esclarecem o significado do termo em seu artigo alinhado aos documentos supracitados e detalham de forma minuciosa os três pilares do pensamento computacional apresentando os aspectos de cada um. Iremos focar aqui nos aspectos de cada pilar, pois a compreensão de cada um já foi mencionada anteriormente e está em consonância com o artigo dos autores.

O primeiro pilar, abstração, compreende os seguintes aspectos: abstrações para representar dados por meio de registros (coleção de informações de um objeto), listas (sequência de informações), grafos (representação) e imagens; abstrações para descrever processos como o uso de operadores de composição (juntar vários passos na descrição de um algoritmo), escolha (definir pontos de escolha de um algoritmo) e repetição (possibilidade que ações possam ser repetidas em um algoritmo de forma controlada); e técnicas para construir algoritmos, sendo que a decomposição (decompor o problema em partes menores), generalização (construir uma solução genérica a partir de outra, possibilitando que o novo algoritmo seja utilizado em outro contexto) e transformação (utilizar a solução do problema para solucionar outros) são os aspectos de destaque (RIBEIRO; FOSS; CAVALHEIRO, 2020).

O segundo pilar, automação, envolve os seguintes aspectos: máquinas (escolha do tipo de máquina para realizar a tarefa); linguagem (escolha do tipo de linguagem a ser utilizada para descrever a solução) e modelagem computacional (utilização de modelos para "[...] auxiliar no entendimento de um problema, permitindo a simulação do comportamento dos sistemas envolvidos, bem como de soluções propostas" (RIBEIRO; FOSS; CAVALHEIRO, 2020, p. 27)). O terceiro e último pilar, análise, envolve viabilidade (avaliar se a solução possui ou não as características desejadas); correção (verificar se o algoritmo construído é de fato correto com refinamentos e transformações); e eficiência (avaliar se o algoritmo é eficiente) (RIBEIRO; FOSS; CAVALHEIRO, 2020).

Pinto e Nascimento (2018) mostram como o pensamento computacional é importante para dar poder, criatividade e inventividade aos alunos. Analisando os conceitos referentes ao termo, os autores dizem que é possível encontrar alguns pontos em comum entre a teoria piagetiana e o pensamento computacional, dentre eles o aprendizado através do fazer e a criatividade. Logo, algumas estratégias vêm sendo empregadas na comunidade escolar, dentre elas as *FabLabs*, o movimento *maker* e o uso de jogos, sendo que as habilidades que são mais estimuladas no ensino de programação são as propostas pelo ISTE/CSTA (2011). Os autores concluem que o pensamento computacional potencializa a criatividade, a inventividade e as formas de resolução de problemas, que se configuram os grandes desafios da nova sociedade e do aprendizado.

Em se tratando da adoção do pensamento computacional, Brackmann *et al*. (2020) apresentam, por meio de uma revisão sistemática de literatura, o panorama global de diversos países que já adotaram ou estão em processo de adoção do pensamento computacional na educação formal ou informal. Ao longo do artigo, os autores destacam iniciativas governamentais adotadas em alguns países como na Alemanha, Argentina, Austrália, Coreia do Sul, Escócia, Estados Unidos, Estônia, França, Finlândia, Grécia e Reino Unido. Apontam para movimentos de organização não governamentais (ONGs) e comerciais que aplicam o pensamento computacional como a *European Schoolnet*, Code.org, Programaê! e SuperGeeks. Por fim, apresentam o estado da arte do pensamento computacional no Brasil.

Rodriguez, Reis e Isotani (2017) também destacam um aumento crescente de projetos e iniciativas com a proposta de desenvolvimento do pensamento computacional nas escolas brasileiras, reforçando a importância dessas iniciativas para o ensino da computação nas

escolas. Como proposta de disseminação e desenvolvimento do pensamento computacional, os autores citam dois projetos realizados em escolas públicas envolvendo alunos da educação básica nas cidades de Belo Horizonte/ MG e Pelotas/RS. Já as iniciativas nacionais referem-se ao desenvolvimento de projetos de jogos digitais usando *Scratch.*

Outros trabalhos relacionados que discutem iniciativas com propostas de explorar o pensamento computacional usando *Scratch* com crianças podem ser verificados em Kaminski e Boscariolli (2018), assim como na obra de Raabe, Zorzo e Blikstein (2020), composta de vários textos que relatam o uso do *Scratch* no desenvolvimento do pensamento computacional, no artigo de Oro *et al*. (2017), que analisa os códigos de programação criados por dois grupos de alunos com 10 a 11 anos em oficinas promovidas pela Escola de *Hackers* - um projeto interinstitucional de Passo Fundo/RS que oportuniza espaço para a aprendizagem de programação de computadores - e no artigo de Zanetti *et al.* (2017), com uma proposta de atividades de programação para crianças de 10 a 13 anos, objetivando explorar o pensamento computacional.

Quanto a estratégias de implementação do pensamento computacional no currículo na educação básica, formação de professores e avaliação do aluno, é importante trazermos para essa discussão o artigo de Valente (2016) e o *Currículo Referência em Tecnologia e Computação: da educação infantil ao ensino fundamental* (RAABE; BRACKMANN; CAMPOS, 2020).

O documento *Currículo Referência em Tecnologia e Computação: da educação infantil ao ensino fundamental*[13] também trata da implementação de práticas que ajudem a desenvolver nos alunos competências e habilidades relacionadas à tecnologia e à computação. Foi elaborado a partir de uma análise criteriosa dos principais aprendizados advindos das referências curriculares nacionais e internacionais que já incorporam temas de inovação e tecnologia em seus currículos. Está organizado em três eixos estruturantes (Cultura Digital, Tecnologia Digital e Pensamento Computacional), que subdividem em dez conceitos associados a 147 habilidades. A definição de pensamento computacional apresentada também está afinada ao documento anterior, sendo que os conceitos principais desse eixo são: abstração, algoritmos, decomposição e reconhecimento de padrões. O acesso ao documento se dá de duas formas: pela página virtual ou por impressão, via *download,* na página (RAABE; BRACKMANN; CAMPOS, 2020).

O artigo de Valente (2016) discute as estratégias para implementação do pensamento computacional na educação básica, adotadas em diferentes países, com três grandes categorias apontadas pelo autor. A primeira categoria refere-se a atividades de Ciência da Computação, especialmente à programação, sendo subdividida em duas outras subcategorias, a saber: (i) programação fora da sala de aula, e (ii) disciplinas no currículo sobre letramento digital. Nessa categoria, o foco está no aprendizado de programação, conceitos da Ciência da Computação e preparo para o mercado de trabalho. No Brasil, as escolas técnicas do Ensino Médio adotam essa linha. Como o destaque está no aprendizado de programação, as atividades podem ocorrer tanto fora da sala de aula, com iniciativas de empresas e organizações que têm produzido materiais com foco estritamente na programação de computadores, quanto como parte das disciplinas no currículo em que a programação tem sido interpretada como letramento digital ou da inclusão digital e usada como meio para criar coisas, desenvolver a criatividade, colocar em práticas as ideias, participação social (VALENTE, 2016).

Na segunda categoria, intitulada “O pensamento computacional como disciplina curricular”, Valente (2016) destaca sua implantação como disciplina e cita a Inglaterra como o país mais adiantado na introdução de atividades que possibilitam aos alunos compreender e refletir sobre os impactos, as diferentes formas de utilização e as relações com a sociedade. O ponto positivo da proposta é o fato das atividades não enfocarem somente a programação e o ponto negativo, mas o fato do pensamento computacional ser direcionado a uma única disciplina e não ser incorporado às demais.

Na última categoria, o pensamento computacional como uma atividade transversal ao currículo, o autor enfatiza que a sua exploração ainda está em processo de elaboração, apesar de duas situações terem sido verificadas: na Itália, integrado como tema a ser

[13] Maiores informações sobre o documento em: http://curriculo.cieb.net.br

estudado nas disciplinas da educação básica, com a exploração de atividades como atitude crítica do uso das tecnologias digitais para o trabalho, a vida, a comunicação e o uso do computador para compartilhar, produzir e cooperar através da Internet; e no Brasil, com o Projeto Um Computador por Aluno (Projeto UCA). Embora tenham sido utilizados na realização de atividades curriculares e tenham ficado restritos a uma disciplina, "[...] os projetos que os alunos desenvolveram tinham um caráter interdisciplinar, uma vez que podiam explorar conceitos de Ciências, de Matemática, de Português e Arte" (VALENTE, 2016, p. 885).

Além disso, exemplos de como o pensamento computacional pode ser explorado na educação, a questão da formação de professores e a avaliação do aluno com relação ao desenvolvimento do pensamento computacional são pontos também discutidos pelo autor referenciado. Para que tenham a possibilidade de desenvolver atividades referentes ao pensamento computacional, a formação de professores deverá perpassar pela formação inicial e continuada, a fim de que eles possam se sentir confiantes e motivados a incorporar em suas práticas pedagógicas os conceitos e recursos aprendidos. Já a avaliação "[...] não deve ser se o aluno aprendeu ou não a programar, mas no nível de consciência que ele tem sobre os conceitos computacionais e como isso se manifesta nas diversas atividades que realiza" (VALENTE, 2016, p. 892).

Considerando toda a discussão relacionada ao entendimento do termo, sua inclusão no currículo da educação básica, competências e habilidades desenvolvidas e panorama de adoção em diferentes países e no contexto brasileiro, ressaltamos e concordamos com Valente (2016) que as ideias do pensamento computacional devem ser trabalhadas de forma interligada às disciplinas do currículo, sendo exploradas por meio de atividades desplugadas, robótica educacional, ferramenta de programação visual *Scratch*, produção de narrativas digitais, uso de simulações, criação de jogos digitais, entre outros. Cada atividade apresenta características diferentes, porém, o propósito está no desenvolvimento do pensamento computacional.

5. CONSIDERAÇÕES FINAIS

Nosso objetivo, neste artigo, foi refletir sobre os usos efetivos das tecnologias digitais da informação e comunicação na educação nestes tempos tecnológicos, vislumbrando outras formas de fazer educação, considerando a possibilidade de a criança vir a ser construtora de artefatos digitais por meio do ensino de programação, isto é, por meio de estímulos ao desenvolvimento do pensamento computacional.

Foi possível observar que as reflexões dos autores para com o potencial das tecnologias da informação e da comunicação deve caminhar para os seus contextos de uso, sempre com a participação ativa dos estudantes. É importante destacar, nesse ínterim, que o ensino de programação não é uma prática apenas restrita a área da Computação ou a pessoas que se interessam em seguir na área de tecnologia, mas, sim, uma ação que envolve a todos.

Logo, verificamos uma crescente valorização do trabalho com a programação nas séries iniciais do ensino fundamental com crianças. Nesse processo, criar condições para que a criança construa conhecimento é o caminho. O construcionismo de Papert propõe o uso significativo dos computadores aliado às práticas pedagógicas elencadas anteriormente. Consequentemente, com a efetivação dessa proposta, temos o desenvolvimento do pensamento computacional ao potencializar a criatividade, a inventividade, as formas de resolução de problemas, sendo esses os grandes desafios da nova sociedade e do aprendizado.

Para a educação, vislumbramos um horizonte de possibilidades às práticas pedagógicas como ensino e aprendizagem mais colaborativos, valorização das experiências e conhecimentos dos alunos, erro como fonte de aprendizado, diálogos, interações, possibilidades de criação de conteúdos e liberdade em divulgar suas produções.

REFERÊNCIAS

BONILLA, Maria Helena; PRETTO, Nelson De Luca. Política educativa e cultura digital: entre práticas escolares e práticas sociais. **Revista Perspectiva**, Florianópolis, v. 33, n. 2, p. 499-521, maio/ago. 2015. Disponível em: https://periodicos.ufsc.br/index.php/perspectiva/article/view/2175-795X.2015v33n2p499. Acesso em: 09 out. 2020. https://doi.org/10.5007/2175-795X.2015v33n2p499

BRACKMANN, Christian Puhlmann. **Desenvolvimento do pensamento computacional através de atividades**

desplugadas na educação básica. 2017. p. 226. Tese (Doutorado em Informática na Educação) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017.

BRACKMANN, Christian Puhlmann *et al*. Panorama global da adoção do pensamento computacional. *In*: RAABE, André; ZORZO, Avelino F.; BLIKSTEIN, Paulo. **Computação na educação básica:** fundamentos e experiências. Porto Alegre: Penso, 2020, p. 31-48.

CASTELLS, Manuel. A revolução da Tecnologia da Informação. *In*: CASTELLS, Manuel. **A sociedade em rede**. 6. ed. São Paulo: Paz e Terra, 1999, p. 67-118.

COLL, César; MONEREO, Carles. Educação e aprendizagem no século XXI: novas ferramentas, novos cenários, novas finalidades. *In*: COLL, César; MONEREO, Carles. **Psicologia da educação virtual**: aprender e ensinar com as tecnologias da informação e da comunicação. Porto Alegre: Artmed, 2010, p. 15-46.

FURINI, Caroline da Silva; TEIXEIRA, Adriano Canabarro; TRENTIN, Marco. O ensino de programação de computadores na educação infantil. *In*: RAABE, André; ZORZO, Avelino F.; BLIKSTEIN, Paulo. **Computação na educação básica:** fundamentos e experiências. Porto Alegre: Penso, 2020, p. 67-78.

GLIZT, Fabiana Rodrigues de Oliveira. O pensamento computacional nos anos iniciais do ensino fundamental. *In*: RAABE, André; ZORZO, Avelino F.; BLIKSTEIN, Paulo. **Computação na educação básica:** fundamentos e experiências. Porto Alegre: Penso, 2020, p. 201-216.

ISTE/CSTA. *Computational Thinking Teacher Resource*. 2 ed., 2011. Disponível em: https://cdn.iste.org/www-root/ct-documents/ct-teacher-resources_2ed-pdf.pdf?sfvrsn=2. Acesso em: 12 jan. 2021.

JESUS, Carla; VASCONCELOS, José Braga de; LIMA, Rui. **Scratch e Kodu:** iniciação à programação no ensino básico. Lisboa: FCA - Editora de Informática, Ltda. 2016, p. 195.

JONASSEN, David H. **Computadores, Ferramentas Cognitivas**: desenvolver o pensamento crítico nas escolas. Editora: Porto Editora, 2000, p. 316.

KAMINSKI, Marcia Regina; BOSCARIOLI, Clodis. Criação de jogos digitais na perspectiva de introdução à Modelagem Matemática nos anos iniciais. **Revista Thema**, Pelotas, v. 18, n. 4, p. 1538-1548, ago. 2018. Disponível em: https://www.researchgate.net/publication/328050322_Criacao_de_jogos_digitais_na_perspectiva_de_introducao_a_Modelagem_Matematica_nos_anos_iniciais. Acesso em: 12 out. 2020. https://doi.org/10.15536/thema.15.2018.1538-1548.1060

KENSKI, Vani Moreira. O que são tecnologias? Como convivemos com as tecnologias? *In*: KENSKI, Vani Moreira. **Tecnologias e ensino presencial e a distância**. 9. ed. Campinas, SP: Papirus, 2012, p. 17-27.

MARJI, Majed. **Aprenda a programar com Scratch:** uma introdução visual à programação com jogos, arte, ciência e matemática. São Paulo: Editora Novatec, 2014, p. 284.

ORO, Neuza Teresinha *et al*. Programação de computadores com o Scratch: contando histórias construídas por alunos da escola de Hackers. **Revista Tecnologias, Sociedade e Conhecimento**, Campinas, v. 4, n. 1, p. 144-163, dez. 2017. Disponível em: https://www.nied.unicamp.br/revista/index.php/tsc/article/view/155. Acesso em: 15 nov. 2020.

PAPERT, Seymour. **A máquina das crianças:** repensando a escola na era da informática. Porto Alegre: Artes Médicas, 1994, p. 195.

PINTO, Sérgio C. C. da Silva; NASCIMENTO, Gisele S. R. do. O pensamento computacional e a nova sociedade. *In*: VALENTE, José Armando; FREIRE, Fernanda Maria Pereira; ARANTES, Flávia Linhalis. **Tecnologia e Educação:** passado, presente e o que está por vir. Campinas, São Paulo: NIED/UNICAMP, 2018, p. 302-322.

RAABE, André; BRACKMANN, Christian Puhlmann; CAMPOS, Flávio. **Currículo de Referência em Tecnologia e Computação:** da educação infantil ao ensino fundamental. 2 ed. São Paulo: CIEB, 2020, p. 104.

RAABE, André; ZORZO, Avelino F.; BLIKSTEIN, Paulo. **Computação na educação básica:**

fundamentos e experiências. Porto Alegre: Penso, 2020, p. 316.

RIBEIRO, Leila; FOSS, Luciana; CAVALHEIRO, Simone André da Costa. Entendendo o pensamento computacional. *In*: RAABE, André; ZORZO, Avelino F.; BLIKSTEIN, Paulo. **Computação na educação básica:** fundamentos e experiências. Porto Alegre: Penso, 2020, p. 16-30.

RODRIGUEZ, Carla Lopes; REIS, Rachel Carlos Duque; ISOTANI, Seiji. Recursos e estratégias para desenvolvimento e avaliação do pensamento computacional na escola. **Revista Tecnologias, Sociedade e Conhecimento**, Campinas, v. 4, n. 1, p. 59-81, dez. 2017. Disponível em: https://www.nied.unicamp.br/revista/index.php/tsc/article/view/165. Acesso em: 15 nov. 2020.

SELWYN, Neil. Educação e Tecnologia: questões críticas. *In*: FERREIRA, Giselle Martins dos Santos; ROSADO, Luiz Alexandre da Silva; CARVALHO, Jaciara de Sá (orgs.) **Educação e Tecnologias**: abordagens críticas. Rio de Janeiro: SESES, 2017. p. 85-103. Disponível em: https://ticpe.files.wordpress.com/2017/04/ebook-ticpe-2017.pdf. Acesso em: 10 out. 2020.

SOBREIRA, Elaine Silva Rocha; TAKINAMI, Olga Kikue; SANTOS, Verônica Gomes dos. Programando, criando e inovando com o Scratch: em busca da formação do cidadão do século XXI. *In*: Jornada de Atualização em Informática na Educação, 2, 2013, Campinas, São Paulo. **Anais** [...]. Disponível em: https://www.br-ie.org/pub/index.php/pie/article/view/2592. Acesso em: 17 jan. 2021. https://doi.org/10.5753/CBIE.JAIE.2013.126

SOBREIRA, Elaine Silva Rocha; VIVEIRO, Alessandra Aparecida; D'ABREU, João Vilhete Viegas. Cultura *maker* e jogos digitais. *In*: MEIRA, Luciano; BLIKSTEIN, Paulo. **Ludicidade, jogos digitais e gamificação na aprendizagem**. Porto Alegre: Penso, 2020, p. 27-38.

SOCIEDADE BRASILEIRA DE COMPUTAÇÃO. **Referenciais de Formação em Computação:** educação básica. 2017. Disponível em: https://www.sbc.org.br/files/ComputacaoEducacaoBasica-versaofinal-julho2017.pdf. Acesso em: 03 jan. 2021.

VALENTE, José Armando. **Informática na educação:** instrucionismo x construcionismo. Manuscrito não publicado: NIED, 1997.

_____. **A Espiral da Espiral de Aprendizagem:** o processo de compreensão do papel das tecnologias de informação e comunicação na educação. 2005. 232f. Tese (Livre Docência). Instituto de Artes. Universidade Estadual de Campinas, Campinas, 2005. Disponível em: http://repositorio.unicamp.br/handle/REPOSIP/284458 . Acesso em: 03 jun. 2021.

_____. 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: 02 jun. 2021.

_____. Pensamento Computacional, Letramento Computacional ou Competência Digital? Novos desafios da educação**. Revista Educação e Cultura Contemporânea**, Rio de Janeiro, v. 16, n. 43, p. 147-168, fev. 2019. Disponível em: http://periodicos.estacio.br/index.php/reeduc/article/view/5852/47965988. Acesso em: 02 jun. 2021. https://doi.org/10.5935/2238-1279.20190008

VIEIRA, Marli Fátima Vick; CAMPOS, Flavio Rodrigues; RAABE, André. O legado de Papert e da linguagem Logo no Brasil. *In*: RAABE, André; ZORZO, Avelino F.; BLIKSTEIN, Paulo. **Computação na educação básica:** fundamentos e experiências. Porto Alegre: Penso, 2020, p. 49-63.

VIEIRA, Marli Fátima Vick; SANTANA, André Luiz Maciel; RAABE, André Luiz Alice. Do Logo ao Pensamento Computacional: o que se pode aprender com os resultados do uso da linguagem Logo nas escolas brasileiras. **Revista Tecnologias, Sociedade e Conhecimento**, Campinas, v. 4, n. 1, p. 82-106, dez. 2017. Disponível em: https://econtents.bc.unicamp.br/inpec/index.php/tsc/article/view/14486. Acesso em: 02 jun. 2021.

WING, Jeannette M. Computational thinking. **Communications of the ACM**, v. 49, n. 3, p. 33-35, mar. 2006. Disponível em:

https://www.researchgate.net/publication/274309848_Computational_Thinking. Acesso em: 07 jun. 2021. https://doi.org/10.1145/1118178.1118215

_____. Computational thinking and thinking about computing. **Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences**, v. 366, n. 1881, p. 3717–3725, jul. 2008. Disponível em: https://www.researchgate.net/publication/23142610_Computational_thinking_and_thinking_about_computing. Acesso em: 07 jun. 2021. https://doi.org/10.1098/rsta.2008.0118

_____. Computational Thinking: What and Why?. 2010. Disponível em: http://www.cs.cmu.edu/~CompThink/papers/TheLinkWing.pdf. Acesso em: 03 jun. 2021.

_____. Computational Thinking Benefits Society. **Social Issues in Computing**, 2014. Disponível em: http://socialissues.cs.toronto.edu/2014/01/computational-thinking/. Acesso em: 03 jun. 2021.

ZANETTI, Humberto Augusto Piovesana; BORGES, Marcos Augusto Francisco; RICARTE, Ivan Luiz Marques. Pensamento Computacional no Ensino de Programação: uma revisão sistemática da literatura. *In*: Congresso Brasileiro de Informática na Educação, 5, 2016, Uberlândia, Minas Gerais. **Anais** [...]. Disponível em: https://www.br-ie.org/pub/index.php/sbie/article/view/6677/4566. Acesso em: 03 jun. 2021. https://doi.org/10.5753/cbie.sbie.2016.21

ZANETTI, Humberto Augusto Piovesana *et al.* Proposta de ensino de programação para crianças com Scratch e Pensamento Computacional. **Revista Tecnologias, Sociedade e Conhecimento**, Campinas, v. 4, n. 01, p. 43-58, dez. 2017. Disponível em: https://econtents.bc.unicamp.br/inpec/index.php/tsc/article/view/14484. Acesso em: 03 jun. 2021.