

**UNIVERSIDADE DE CAXIAS DO SUL
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
ÁREA DO CONHECIMENTO DE HUMANIDADES
PROGRAMA DE PÓS-GRADUAÇÃO EM EDUCAÇÃO**

PAULO ANTONIO PASQUAL JÚNIOR

**PENSAMENTO COMPUTACIONAL E FORMAÇÃO DE PROFESSORES: UMA
ANÁLISE A PARTIR DA PLATAFORMA CODE.ORG**

Caxias do Sul

2018

**UNIVERSIDADE DE CAXIAS DO SUL
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
ÁREA DO CONHECIMENTO DE HUMANIDADES
PROGRAMA DE PÓS-GRADUAÇÃO EM EDUCAÇÃO**

PAULO ANTONIO PASQUAL JÚNIOR

**PENSAMENTO COMPUTACIONAL E FORMAÇÃO DE PROFESSORES: UMA
ANÁLISE A PARTIR DA PLATAFORMA CODE.ORG**

Dissertação apresentada ao programa de Pós-Graduação Stricto Sensu em Educação da Universidade de Caxias do Sul, como requisito para obtenção do grau de Mestre em Educação.

Orientadora Professora Dra. Carla Beatris Valentini

Caxias do Sul

2018

Dados Internacionais de Catalogação na Publicação (CIP)
Universidade de Caxias do Sul
Sistema de Bibliotecas UCS - Processamento Técnico

P284p Pasqual Júnior, Paulo Antonio
Pensamento computacional e formação de professores : uma análise
a partir da plataforma Code.org / Paulo Antonio Pasqual Júnior. – 2018.
120 f. : il. ; 30 cm

Dissertação (Mestrado) - Universidade de Caxias do Sul, Programa
de Pós-Graduação em Educação, 2018.

Orientação: Carla Beatris Valentini.

1. Ensino auxiliado por computador. 2. Professores - Formação. 3.
Software educacional. I. Valentini, Carla Beatris, orient. II. Título.

CDU 2. ed.: 37.018.43:004

Catalogação na fonte elaborada pela(o) bibliotecária(o)
Carolina Machado Quadros - CRB 10/2236



UNIVERSIDADE DE CAXIAS DO SUL

“Pensamento computacional e formação de professores: uma análise a partir da Plataforma Code.org”

Paulo Antonio Pasqual Júnior

Dissertação de Mestrado submetida à Banca Examinadora designada pela Coordenação do Programa de Pós-Graduação em Educação da Universidade de Caxias do Sul, como parte dos requisitos necessários para a obtenção do título de Mestre em Educação. Linha de Pesquisa: Educação, Linguagem e Tecnologia.

Caxias do Sul, 03 de julho de 2018.

Banca Examinadora:

Dra. Carla Beatris Valentini (presidente – UCS)

Dra. Eliana Rela (UCS)

Dra. Carine Geltrudes Webber (UCS)

Dr. Eliseo Berni Reategui (UFRGS)

CAMPUS-SEDE

Rua Francisco Getúlio Vargas, 1130 – Bairro Petrópolis – CEP 95070-560 – Caxias do Sul – RS – Brasil

Ou: Caixa Postal 1352 – CEP 95020-972 – Caxias do Sul – RS – Brasil

Telefone / Telefax (54) 3218.2100 – www.ucs.br

Entidade Mantenedora: Fundação Universidade de Caxias do Sul – CNPJ 88 648 761/0001-03 – CGCTE 029/0089530

AGRADECIMENTOS

Agradeço aos meus pais, Onildes Pasqual e Paulo A. Pasqual, e ao meu irmão, Jean Bruno Pasqual, pelas diversas vezes que compreenderam as minhas ausências.

Ao Gustavo, pelo incentivo que me deu para iniciar o mestrado e por sempre me incentivar a acreditar mais no meu potencial.

Aos meus colegas, amigos e professores do mestrado, pelas inúmeras interações e trocas que possibilitam a construção dos conhecimentos durante esse tempo.

Aos meus amigos, que compreenderam a minha ausência e a recusa de diversos convites.

Aos professores da banca de qualificação, que muito ajudaram com novas ideias para a construção desta dissertação.

À ClassCode que, ao mesmo tempo em que desviou a minha atenção, contribuiu muito para uma visão crítico-prática de atuação com o pensamento computacional.

E, por fim, um especial agradecimento a minha querida orientadora, professora Dr.^a Carla Beatris Valentini, que me pegou pela mão e me ajudou a dar os meus primeiros passos como pesquisador. Sou grato pela paciência e motivação com as quais orientou meu trabalho.

“Tudo que se ensina a uma criança, a criança não pode mais, ela mesma, descobrir ou inventar.”

Jean Piaget

RESUMO

O pensamento computacional tem sido discutido na literatura como uma nova competência para o cidadão do século XXI. Nesse caminho, a inclusão da Ciência da Computação e de linguagens de programação nas escolas têm acontecido em diversos países. O tema perpassa a educação e, consequentemente, a formação de professores. Embora diversas discussões na academia tenham emergido acerca do tema, ainda existem poucos estudos que se dedicam a discutir questões ligadas à formação de professores no cenário nacional. Esta dissertação objetiva analisar as concepções de ensino e aprendizagem presentes em uma plataforma *on-line*, com o intuito de criar indicadores para a docência que possam nortear processos de formação pedagógica. Para amparar teoricamente este estudo, foram utilizadas as bases teóricas do construtivismo e construcionismo, tendo como expoentes Jean Piaget, Seymour Papert e seus seguidores. Esta pesquisa caracteriza-se como estudo de caso, de cunho exploratório, e tem como fonte de análise a plataforma *Code.org*. Por se tratar de um estudo de caso, foram elencadas como fontes de evidências postagens de dois tópicos de um fórum, planos de aula e atividades *on-line*. A metodologia utilizada para a análise das fontes de evidências foi a Análise Textual Discursiva, em que emergiram três grandes categorias: Concepções de Aprendizagem, Aspectos Dificultadores da Aprendizagem e Aspectos Promotores da Aprendizagem. Os resultados obtidos a partir das categorias emergentes evidenciam que as concepções de aprendizagem presentes na plataforma são prioritariamente empiristas, trazendo uma proposta em que os materiais disponíveis para o professor sugerem um modelo de ensino e aprendizagem baseado em uma pedagogia diretiva. Os achados ainda apontam para estratégias de ensino e aprendizagem baseadas em atividades *on-line*, atividades *off-line*, interação entre pares e, por fim, apresentam aspectos dificultadores para o ensino e a aprendizagem do pensamento computacional, sendo eles tempo, estrutura e dificuldades com TI.

Palavras-chave: Pensamento Computacional; Concepções de Aprendizagem; Formação de Professores.

ABSTRACT

The computational thinking has been discussed in the literature as a new competence for the citizen of the 21st century. In this way, the inclusion of Computer Science and programming languages in schools has been happening in several countries. The theme pervades education and consequently, teacher training. Although several discussions have emerged about the subject in the academy, there are still few studies dedicated to discuss issues related to the teachers training in the national scenario. This dissertation aims to analyze the concepts of teaching and learning present on an online platform, with intent to create indicators for teaching, that may guide pedagogical training processes. In order to support this study theoretically, the theoretical bases of constructivism and constructionism were used, having Jean Piaget, Seymour Papert and their followers as exponents. This research is characterized as an exploratory case study and has the Code.org platform as source of analysis. As a case study, postings of two forum topics, lesson plans and online activities were listed as sources of evidence. The methodology used for the analysis of the sources of evidence was Discursive Textual Analysis, from which three broad categories emerged: Conceptions of Learning, Difficulting Aspects of Learning and Promoting Aspects of Learning. The obtained results from the emerging categories show that the Conceptions of Learning present in the platform, are primarily empiricists, bringing a purpose in which the available materials for the teacher suggest a teaching and learning model based on a directive pedagogy. The findings still point out to strategies for teaching learning based on online activities, offline activities, interaction between peers, and eventually, show difficulting aspects for teaching and learning of computational thinking, which are: time, structure and difficulties with IT.

Keywords: Computational Thinking; Conceptions of Learning; Teacher Training.

LISTA DE FIGURAS

Figura 1- Cooperação.....	26
Figura 2 - Léa Fagundes e Seymour Papert	31
Figura 3- Pensamento computacional.....	42
Figura 4 - Publicações sobre PC.....	45
Figura 5 - Teses e dissertações.....	48
Figura 6 - Interface Scratch	53
Figura 7 - Scratch para educadores.....	54
Figura 8 - Interface do Tynker	55
Figura 9 - Programação Kodu	56
Figura 10 - Programação AppInventor.....	57
Figura 11 - Fórum	67
Figura 12- Mapa	68
Figura 13 - Curso 2	68
Figura 14 - Fase 4	69
Figura 15 - Curso 2	70
Figura 16 - Vocabulário da aula.....	74
Figura 17 - Avaliação	76
Figura 18 - Mensagem de erro 1	80
Figura 19 - Mensagem de erro 2.....	80
Figura 20 - Mensagem de erro 3.....	80
Figura 21 - Mensagem Desafio 1	81
Figura 22 - Personagem - erro	81
Figura 23 - Mensagem de Acerto.....	81
Figura 24 - Atividade aviões de papel.....	90
Figura 25 - Labirinto e sequência	92
Figura 26 - Labirinto e Sequência 2.....	93
Figura 27- Laços de Repetição	94

LISTA DE QUADROS

Quadro 1 - Artigos Analisados.....	45
Quadro 2- Categorias e subcategorias.....	64
Quadro 3 - Estrutura dos planos de aula	71

SUMÁRIO

1	APRESENTAÇÃO	11
2	ALGUMAS QUESTÕES SOBRE APRENDIZAGEM.....	17
2.1	MODELOS PEDAGÓGICOS.....	17
2.2	APRENDIZAGEM EM PIAGET.....	20
2.3	PAPERT VIDA E OBRA	27
2.4	O CONSTRUCIONISMO DE PAPERT	32
3	PENSAMENTO COMPUTACIONAL.....	35
3.1	UM BREVE HISTÓRICO DO PC NA EDUCAÇÃO	35
3.2	PERCURSO DO CONCEITO DE PENSAMENTO COMPUTACIONAL	38
3.3	PENSAMENTO COMPUTACIONAL: TRABALHOS INVESTIGADOS.....	44
3.4	PC: PLATAFORMAS VIRTUAIS E FORMAÇÃO DE PROFESSORES	50
3.4.1	Scratch	53
3.4.2	Tynker	55
3.4.3	Kodu.....	56
3.4.4	Appinventor	56
3.4.5	Code.Org	57
3.4.6	Programaê!	58
4	MÉTODO	59
4.1	DELINEAMENTO METODOLÓGICO	59
4.2	CONTEXTO E ORGANIZAÇÃO DOS DADOS	60
4.2.1	Fórum.....	61
4.2.2	Atividades.....	62
4.2.3	Área Do Professor E Planos De Aula.....	63
4.3	PROCEDIMENTOS DE ANÁLISE.....	63
5	ANÁLISE E DISCUSSÃO	66
5.1	A PLATAFORMA <i>CODE.ORG</i> : UMA VISÃO GERAL	66
5.2	CONCEPÇÕES DE APRENDIZAGEM EMERGENTES NA PLATAFORMA	71

5.2.1	Planos De Aula	73
5.2.2	Atividades/Tutoriais.....	79
5.2.3	Fórum.....	84
5.3	ASPECTOS PROMOTORES DO PROCESSO DE APRENDIZAGEM.....	85
5.3.1	Interação Entre Pares	85
5.3.2	Atividades <i>Off-Line</i>	89
5.3.3	Atividades <i>On-Line</i>	92
5.4	DIFICULTADORES DA PRÁTICA COM O PC.....	95
5.5	INDICADORES PARA A DOCÊNCIA COM O PC.....	98
6	CONSIDERAÇÕES FINAIS	106
7	REFERÊNCIAS	113

1 APRESENTAÇÃO

“O ideal da educação não é aprender ao máximo, maximizar os resultados, mas é antes de tudo aprender a aprender, é aprender a se desenvolver e aprender a continuar a se desenvolver depois da escola.”
Jean Piaget

Comecei muito cedo minha vida de professor como instrutor de informática em uma escola de cursos livres. Mesmo diante da ingenuidade de jovem, pensava que a educação não poderia estar desvinculada das tecnologias e que elas viriam agregar aos processos de aprendizagem, embora em minha visão, ainda primitiva, essa mudança estivesse muito mais ligada à inclusão das tecnologias como um recurso a ser utilizado para a substituição de processos, do que efetivamente para uma mudança de paradigma.

A experiência de ensinar (e aprender) me instigou ao ingresso no curso de Licenciatura em Computação que, dentre muitas competências, possibilitou uma visão diferenciada sobre as perspectivas da aprendizagem e da tecnologia. Nessa trajetória de minha constituição docente, percebi que ser professor ia além das experiências diretas que tive na educação básica e que tinha, por isso, obrigação de buscar um rompimento com o paradigma da transmissão em minha prática.

Lembro-me que um dos meus primeiros contatos com os referenciais teóricos da educação foi com o livro *Pedagogia da Autonomia: Saberes Necessários à Prática Educativa*, de Paulo Freire. Esse momento foi realmente um divisor de águas nas minhas concepções de educação e, a partir daí, procurei levar como subsídio algumas das lições de Freire, sempre buscando interlocuções com o campo das tecnologias.

Nesse caminho, entendi que as Tecnologias da Informação e Comunicação (TICs) tinham papel fundamental na educação contemporânea e se faziam necessárias para qualquer professor. Percebi a grande escolha que havia feito ao me aventurar no campo da Informática na Educação.

Em algum período do curso, não me lembro ao certo qual, fui apresentado ao LOGO e, na sequência, ao *Scratch*, duas plataformas que fomentavam a programação, em especial pelas crianças. Pensar que uma criança poderia criar o seu próprio *game*, história ou animação era realmente interessante para mim, que sempre achei a escola um lugar de quase inércia, afinal, minhas experiências com a educação básica não foram muito divertidas.

Quando criança, pensava em criar, primeiramente, um desenho animado (talvez outras crianças da minha época também tivessem essa vontade) e depois, quando entrei no mundo dos

games, minha vontade sempre foi ir além de ser um simples jogador. Unir o *Scratch* com um sonho antigo e ajudar os meus alunos a criarem seus próprios games seria como voltar à infância e realizar meu próprio sonho. Percebo que a escola poderia aproveitar mais essa vontade de criar. Isso me faz lembrar aquela famosa metáfora de Rubem Alves, que diz o seguinte: “Há escolas que são gaiolas e há escolas que são asas”. Talvez as escolas de hoje ainda sejam muito mais gaiolas do que asas, afinal de contas, as crianças são naturalmente criativas e aprendem com a escola a arte de fazer sempre a mesma coisa, de copiar em vez de criar, de fazer o exercício tal e qual está no exemplo. É tanto tempo para (des) aprender, que falta tempo para criar. Basta perceber quantos dos grandes gênios tiveram dificuldades em enfrentar a escola regular.

Certa vez, ouvi Carlos Skliar dando um depoimento sobre uma enquete feita às crianças que iriam à escola pela primeira vez. Na entrada, segundo ele, elas estavam afoitas, empolgadas com a nova experiência do que seria a escola. Na saída, elas estavam desanimadas, porque a escola não era nada do que elas pensavam. Essa fala do Skliar me faz pensar o quanto é preciso refletir sobre as práticas escolares e como as tecnologias têm nos oportunizado repensar o fazer escolar.

A ideia de se incluir o Pensamento Computacional (PC) na escola vem ao encontro de uma nova escola, uma escola que possibilite a autoria, a criatividade e o desenvolvimento de competências que se aproximem do cotidiano do aluno contemporâneo. E dessa demanda emerge a necessidade de se pensar a docência.

Junto do início desta dissertação, passei a atuar como sócio fundador e docente da ClassCode, uma escola de tecnologia para jovens, que tem como objetivo desenvolver o pensamento computacional de crianças e adolescentes por meio do desenvolvimento de *games*. Esta nova demanda me inquietou ainda mais a respeito do ensino e da aprendizagem, não apenas do PC, mas de maneira geral acerca da epistemologia por trás das práticas pedagógicas. Com isso, surgiu a inquietação de pensar a formação de professor para a atuação neste novo contexto. Não há como pensar a formação do professor sem antes pensar as concepções de conhecimento (epistemologia) que estão no pano de fundo da ação docente.

Há muitas correntes que tentam explicar a maneira como o conhecimento surge. Sejam elas oriundas da filosofia ou da psicologia, todas buscam explicar como o conhecimento é desenvolvido na mente humana. A filosofia, por exemplo, buscou explicar, desde a antiguidade clássica, a origem do conhecimento humano. Aristóteles acreditava que o conhecimento se dava a partir da experiência. Nessa visão epistemológica, tudo o que o sujeito aprende é fruto dos sentidos. Ou seja, nessa visão todo indivíduo é vazio e se preenche com as experiências do

mundo. Outra vertente, ainda oriunda da antiguidade clássica, é o Inatismo, uma corrente que se origina em Platão e que tenta explicar que todos os conhecimentos do ser humano são inatos. Dentro dessa perspectiva epistemológica, não é preciso ensinar nada a ninguém, pois o indivíduo irá aprender aquilo que ele estiver pré-disposto a aprender desde o seu nascimento (BECKER, 1993). Não é possível ensinar matemática para aqueles que não nasceram para a matemática, por exemplo.

Essas duas epistemologias se consolidaram como a principal maneira de conceber a aprendizagem humana. Da antiguidade clássica aos dias de hoje, essas visões se concretizaram a partir de outros expoentes da filosofia durante a modernidade, como Locke e outros.

Nossas escolas, até bem pouco tempo, para não dizer atualmente, tiveram seus modelos pedagógicos baseados nesse tipo de corrente filosófica. Nessa perspectiva, o professor tem papel fundamental na educação e o foco pedagógico está no ensino. O professor tem responsabilidade de garantir que o aluno aprenda, pois o conhecimento é “passado” do professor para o aluno por meio dos sentidos. O famoso “transmitir conhecimento”.

Nesse modelo escolar, os fracassos são muitos e se justificam por causa da “incapacidade do aluno de aprender”. É comum ouvirmos falas como “Esse aluno nunca vai conseguir aprender isso” ou, ainda, “Ele não tem capacidade para isso”. Esses problemas da educação permeiam a epistemologia inatista, pois acreditam que algumas coisas não podem ser aprendidas por aqueles cuja disposição para isso não existe. O modelo pedagógico mais difundido até bem pouco tempo se baseava nessas duas correntes: O conhecimento é fruto da experiência dos sentidos e, além disso, só pode ser aprendido por aqueles que, de alguma forma, possuem pré-disposição para aprender.

Durante o século XX, esse cenário começou a mudar e muitos teóricos passaram a se questionar sobre a verdadeira gênese da Aprendizagem. Jean Piaget foi um desses grandes nomes. Ele se preocupou em investigar como o conhecimento era desenvolvido pela criança e observou milhares de crianças durante os mais de cinquenta anos de trabalho. Piaget desenvolveu um vasto trabalho no campo da Epistemologia Genética, uma área interdisciplinar entre a filosofia e a psicologia, que se preocupou com a origem do conhecimento na mente humana.

Nessa perspectiva, Piaget se recusou a acreditar que o conhecimento era transmitido para o sujeito ou que o sujeito nascia com pré-disposições para aprender coisas específicas. Para o pesquisador, a aprendizagem ocorre por uma série de modificações de estruturas mentais que vão assimilando ou acomodando novos conhecimentos. A obra Piagetina é bastante complexa e possui dezenas de conceitos. Grosso modo, podemos dizer que, segundo Piaget

(1978), o conhecimento não se dá nem por transmissão, nem por predisposição, mas sim por construção. Construção é uma metáfora que Piaget usou para designar as inúmeras modificações cognitivas que ocorrem durante o processo de aprendizagem.

O primeiro passo para entender a aprendizagem, segundo essa perspectiva, é, como dizia Paulo Freire (1996), entender que o conhecimento não pode ser transmitido a ninguém. Entender a aprendizagem como uma construção muda todo o processo pedagógico, pois nessa epistemologia a preocupação se dá com o ato de aprender, e não de ensinar. Como o aluno é o protagonista da aprendizagem, o professor tem a função de mediar os processos de aprendizagem. Piaget é o pai do Construtivismo e, nessa perspectiva, aprender é muito mais importante do que ensinar. Esses dois conceitos podem parecer intimamente ligados e quase que sinônimos, mas na realidade são distintos. Ensinar é um processo externo, enquanto que aprender é um processo interno.

Por um tempo, o construtivismo subjacente à Piaget foi interpretado como uma pedagogia do deixar fazer (LOPES, 2010). Muitos professores ou escolas entenderam que, como a aprendizagem era algo relativo ao aluno, passaram a eximir-se da responsabilidade de “ensinar”. Essa é uma interpretação perigosa e errônea, que pode ter sido a causa de alguns fracassos do construtivismo. O verdadeiro papel da escola e do professor em uma perspectiva construtivista é, em primeiro lugar, entender que o professor não é o personagem principal no palco da aprendizagem, mas que isso não o exime de responsabilidades. Nessa epistemologia, o professor tem responsabilidades que são muito mais complexas do que o simples “passar conteúdo” ou “resolver exercícios no quadro”. O professor, nesse modelo epistemológico, torna-se orientador, sistematizador de experiências, aquele que não dá as respostas, mas possibilita que o aluno reflita, experimente e construa suas hipóteses, a fim de construir seu conhecimento.

Esse papel se intensifica no cenário permeado pelas tecnologias, pois elas permitem uma gama de exploração e experimentação que antes não seriam possíveis. Alguns fracassos escolares, contudo, são de conhecimento de todos e, muitas vezes, as tecnologias na educação são usadas apenas para reproduzir os modelos pedagógicos já utilizados antigamente. É a comum substituição das velhas tecnologias pelas novas tecnologias; quadro por *datashow*, cartaz por apresentador de *slides* e o caderno pelo editor de textos.

Minha visão sobre o papel do professor no contexto tecnológico vai à direção de um professor que seja capaz de utilizar a tecnologia para possibilitar que os alunos tornem-se sujeitos criadores e agentes de suas aprendizagens. Isso quer dizer que os alunos possam aprender a aprender e irem além de “ser usuário” para uma perspectiva de ser criador. É nesse

contexto que a aprendizagem do pensamento computacional e da linguagem de programação se inscrevem.

Em meados dos anos 1960, um teórico cujo trabalho é de extrema importância para nós imaginou que os computadores poderiam ser usados para ensinar. Seymour Papert estava muito à frente de seu tempo e acreditava que aprender uma linguagem de programação possibilitaria às crianças o desenvolvimento de competências que iriam além das ferramentas pedagógicas disponíveis na educação regular da época. Papert era um visionário, já que a tecnologia dos anos 1970 era muito rudimentar e o computador pessoal ainda estava por ser inventado. A visão de Papert serviu de base para a criação de inúmeras tecnologias para o desenvolvimento do pensamento computacional, tecnologias estas que são usadas cada vez mais em escolas do mundo inteiro para o ensino-aprendizagem de programação e conceitos de Ciência da Computação. Essa trajetória aprofundaremos no próximo capítulo.

Dentre as ferramentas mais usadas, de acordo com a literatura, estão os Robôs LEGO, o *Scratch*, o *Code.Org*, o *Kodu*, o *Tynker* e o *AppInventor*; embora uma infinidade de boas opções estejam presentes e à disposição daqueles que desejam aprender.

Essas tecnologias têm uma questão em comum, todas elas foram baseadas na perspectiva de que a aprendizagem ocorre a partir da interação do indivíduo com o meio. Elas foram concebidas para que fossem intuitivas e permitissem a aprendizagem mesmo sem o ensino tradicional. Em minha visão, essas plataformas não foram feitas para serem ensinadas, elas foram feitas para serem aprendidas. Ou seja, qualquer movimento de ensino na modalidade “transmissiva” não converge com o modelo de aprendizagem pelo qual elas foram concebidas. Isso significa que aula expositiva não faz parte desse modelo pedagógico, é preciso mais do que isso para que essas ferramentas sejam bem aproveitadas.

Nesse cenário, chamamos atenção ao papel do professor, que é fundamental para a natureza desta dissertação. O professor, nesse contexto, atua como um problematizador, alguém que, por meio de questionamentos e atividades problema, possibilita os movimentos de aprendizagem do aluno. No caso, a aprendizagem por projetos, por exemplo, configura-se como uma metodologia profícua, uma vez que, a partir de uma situação problema, o aluno pode desenvolver suas hipóteses e construir solução para questões do mundo.

Dessa forma, é necessário que nos distancieemos dos modelos de ensino com os quais estamos acostumados e, como professores, que tenhamos a competência de problematizar em vez de dar respostas, de questionar em vez de transmitir e de propor uma prática pedagógica que vá além explicar no quadro, ouvir e fazer exercícios.

As plataformas disponíveis para o desenvolvimento do Pensamento Computacional permitem uma infinidade de atividades que dispensam a ação do professor tradicional, mas que exigem uma postura epistemológica diferente e que, sem dúvida, vai exigir do professor muito mais energia do que na clássica aula expositiva.

Essas breves considerações nos permitem entender que há uma necessidade iminente de formação de professores, em especial para a atuação com o pensamento computacional, já que é preciso discutir que modelos de ensino-aprendizagem convergem com essas tecnologias e como queremos utilizar essas tecnologias em sala de aula.

Diversas iniciativas estão mundialmente difundidas para a inclusão do pensamento computacional (PC) na sala de aula, porém, no Brasil, por exemplo, ainda há pouca preocupação acerca da inclusão do PC como competência fundamental a ser aprendida no ensino regular e, além disso, há poucas iniciativas que fomentem a formação de professores. O Desse modo, o objetivo geral desta investigação insere-se nessa problemática, pois busca **investigar as concepções de ensino e aprendizagem presentes em uma plataforma *on-line* para o desenvolvimento do pensamento computacional**, por meio da análise de recursos disponibilizados pela plataforma que incluem atividades e postagens de professores presentes no fórum. Além disso, este estudo busca subsídios para que possamos refletir acerca da formação de professores também em nosso contexto. Para atingir o objetivo geral, elencamos os seguintes objetivos específicos:

- a) Identificar os princípios epistemológicos presentes na plataforma.
- b) Analisar as estratégias para ensino e aprendizagem do pensamento computacional.
- c) Analisar as dificuldades relatadas por professores na prática pedagógica com o pensamento computacional.
- d) Construir indicadores para a atuação docente com o pensamento computacional.

Assim, esta dissertação está organizada da seguinte maneira: O Capítulo 2 trata das questões acerca dos aspectos epistemológicos da aprendizagem. O Capítulo 3 aborda o conceito de pensamento computacional, os aspectos históricos e o desdobramento de pesquisas a respeito do tema no cenário nacional. O Capítulo 4 apresenta a metodologia utilizada para a investigação proposta nesta dissertação. O Capítulo 5 expõe e discute os resultados do estudo, traçando indicadores para a atuação docente com o pensamento computacional e, por fim, o Capítulo 6 apresenta as considerações finais e as perspectivas para novos trabalhos que se inscrevem em problemáticas similares a desta dissertação.

2 ALGUMAS QUESTÕES SOBRE APRENDIZAGEM

“Ensinar não é transferir conhecimento, mas criar as possibilidades para a sua própria produção ou a sua construção.”

Paulo Freire

Dedico este primeiro capítulo a introduzir algumas questões acerca da aprendizagem, buscando aprofundamento de alguns aspectos discutidos na apresentação desta dissertação. Em primeiro lugar, discorro a respeito dos modelos pedagógicos vigentes e a relação do professor nesses aspectos. Em segundo lugar, apresento a aprendizagem conforme Piaget e o percurso teórico desenvolvido por ele por julgar necessário explicitar a epistemologia com a qual pesquiso e entendo a aprendizagem. Nessa seção, apresento a aprendizagem sob a ótica piagetiana, introduzindo os principais conceitos elaborados pelo autor. Em seguida, discuto o pensamento concreto e o pensamento formal, conceitos que são importantes para o entendimento dos processos de aprendizagem. Em terceiro lugar, trago as ideias de Seymour Papert, ideias essas que não poderiam deixar de fazer parte de uma pesquisa que se inscreve sob a ótica do pensamento computacional. Dessa forma, este capítulo busca evidenciar também a trajetória de Seymour Papert e suas relações com o pensamento de Piaget, apresentando desde os primeiros caminhos do autor em sua infância até a concepção de uma nova teoria de aprendizagem, o Construcionismo. Além disso, discuto a originalidade das ideias de Papert e a gênese do pensamento computacional em sua obra. Por fim, apresento os caminhos do Construcionismo e as suas contribuições para esta pesquisa. Em síntese, trago neste capítulo uma epistemologia que me permite discutir, ao fim desta dissertação, meu ponto de vista acerca do pensamento computacional e do modelo de aprendizagem necessário para pensar a formação de professores.

2.1 MODELOS PEDAGÓGICOS

A Epistemologia é o ramo da Filosofia que trata do conhecimento e suas origens. Desde a origem da humanidade, o ser humano tem se preocupado com as origens do conhecimento. No decorrer da história, o pensamento sofreu modificações e embasou determinadas concepções do conhecer.

As primeiras inquietações sobre a origem do conhecimento nascem na Grécia Antiga a partir dos grandes pensadores da Antiguidade. Platão acreditava que o conhecimento humano era inato, ou seja, estava presente no sujeito desde o nascimento. Dessa forma, o “aprender”

seria apenas a ação de colocar para fora o que já estava presente no ser desde o seu nascimento. Na visão apriorista de Platão, nada se poderia ensinar a alguém, pois o conhecimento já estaria no indivíduo desde o nascimento. Haveria, portanto, nessa visão, uma série de conhecimentos já inscritos no nascimento que, ao longo da vida, poderiam ser trazidos à tona. Durante a Idade Média, os neoplatonistas, como Santo Agostinho, dão novas interpretações ao pensamento de Platão. Para ele, o conhecimento também é inerente ao ser humano, porém é necessário que a inspiração divina traga à consciência este conhecimento intrínseco na alma do sujeito desde o nascimento (STRATHERN, 1997).

Com Aristóteles, temos uma visão distinta e oposta a Platão. Para ele, o ser humano nasce sem nenhum conhecimento e todo o processo de conhecer é efetivado por meio dos sentidos. Nessa perspectiva, nada se conhece até que se possa absorver pelos sentidos, por estímulos do mundo real. O pensamento Aristotélico foi a base para a filosofia moderna, tendo como grande representante dessa vertente o filósofo inglês John Locke. Para este pensador, o conhecimento está ligado principalmente à experiência sensorial. O empirismo relaciona-se às concepções de ciência moderna e a máxima de que o conhecimento científico é aquele que pode ser experimentado. Para Locke (1999), o ser humano é uma tábula rasa e, a partir das experiências, esse indivíduo, naturalmente vazio, é preenchido pelo conhecimento que entra pelos sentidos. Diz ele que,

Os sentidos inicialmente tratam com idéias particulares, preenchendo o gabinete ainda vazio, e a mente se familiariza gradativamente com algumas delas, depositando-as na memória e designando-as por nomes. Mais tarde, a mente, prosseguindo em sua marcha, as vai abstraíndo, apreendendo gradualmente o uso dos nomes gerais. Por este meio, a mente vai se enriquecendo com idéias e linguagem. (LOCKE, 1999, p. 41).

Essas duas visões do conhecer embasam os modelos pedagógicos tradicionais. O Inatismo se insere na educação por meio da pedagogia não-diretiva (Becker, 1999). Nesse paradigma, parte-se do pressuposto platônico de que o indivíduo já dispõe dos conhecimentos e da capacidade de aprender desde o nascimento. Assim, a função do professor é trazer à tona as capacidades inatas do indivíduo. O professor inatista, segundo Becker (1993), é aquele que entende que a sua função é de simplesmente deixar o aluno fazer, e toda a aprendizagem irá ocorrer única e exclusivamente por meio da capacidade do aluno. Ou seja, o aprender é responsabilidade do aluno e o professor exime-se de qualquer responsabilidade.

O aluno já traz um saber que ele precisa, apenas precisa trazer à consciência, organizar, ou, ainda, recheiar de conteúdo. O professor deve interferir o mínimo possível. Qualquer ação que o aluno decida fazer é, a priori, boa, instrutiva (BECKER, 1999, p. 93).

A relação do conhecer, nesse modelo, fundamenta-se unicamente no sentido sujeito para o objeto do conhecimento. Assim, o conhecimento está presente na herança genética, o aluno irá aprender se estiver geneticamente condicionado para a aprendizagem e nada que o professor fizer poderá interferir nesse movimento. Becker (1993), em “A Epistemologia do Professor”, demonstra que professores com uma visão apriorista consideram que alguns alunos nunca serão capazes de aprender certos conhecimentos.

Já a pedagogia diretiva é, segundo Becker (1999), completamente oposta ao modelo anterior. Baseado na Epistemologia Empirista, a pedagogia diretiva entende que toda aprendizagem ocorre de fora para dentro, na direção objeto para o sujeito.

Nesse modelo, o professor entende que o aluno, assim como propunha a filosofia de Locke, é tábula rasa, ou seja, o sujeito é entendido como uma folha de papel em branco, e a função da escola e do professor é incutir no aluno os conhecimentos prontos. O professor baseado nessa filosofia compreende que o aluno nada sabe, e a sua função é inserir na mente do aluno os conceitos já prontos. Assim, o aluno tem um papel passivo no aprender, tendo a responsabilidade de prestar atenção e repetir. O ensino no modelo diretivo baseia-se em conteúdo, conceitos prontos e repetição. De acordo com Becker (1999), o modelo de educação empirista é geralmente o modelo mais convencional de ensino. Baseado em exposições de conteúdo, repetição e exercícios de fixação. É o “ensino bancário” criticado por Freire (1996), em que são adicionados depósitos de conhecimento para que, em algum momento necessário, o aluno possa resgatá-los e utilizá-los.

Para Becker (1999), a ação do professor no sentido de tentar incutir no aluno conceitos prontos baseia-se na crença de que o conhecimento é algo que pode ser transmitido e, a partir daí, a prática pedagógica reverte-se em ações que validem essa crença.

O professor considera que seu aluno é tabula rasa não somente quando ele nasceu como ser humano, mas frente a cada novo conteúdo estocado na sua grade curricular, ou nas gavetas de sua disciplina. A atitude, nós a conhecemos. O alfabetizador considera que seu aluno nada sabe em termos de leitura e escrita e que ele tem que ensinar tudo. Mais adiante, frente à aritmética, o professor, novamente, vê seu aluno como alguém que nada sabe sobre somas e subtrações. (BECKER, 1999, p. 90).

É importante ressaltar que nesse modelo de aprendizagem fala-se em ensino, pois a ação de conhecer baseia-se, exclusivamente, na ação de ensinar, e não pela ação mútua de

ensino e de aprendizagem. Dessa forma, o professor nunca aprende ao ensinar e o aluno jamais ensina ao aprender. Freire (1996) critica essa forma de conceber a educação, uma vez que, na visão do autor, o ensino e a aprendizagem são ações indissociáveis.

O pensamento de Freire (1996) se insere, nesse sentido, em outro modelo pedagógico chamado por Becker (1999) de pedagogia relacional. Nesse modelo, o ato de conhecer não é nem responsabilidade exclusiva do sujeito, tampouco do objeto. Há uma relação mútua de interações e trocas que possibilitam o conhecer.

O modelo relacional deriva do Construtivismo Piagetiano, em que se acredita que o conhecimento é fruto da construção do indivíduo a partir da interação, tanto com o meio físico como com o meio social. Becker exemplifica a ação de um professor nesse modelo da seguinte maneira:

O professor e os alunos entram na sala de aula. O professor traz algum material - algo que, presume, tem significado para os alunos. Propõe que eles explorem este material - cuja natureza depende do destinatário: crianças de pré-escola, de primeiro grau, de segundo grau, universitários, etc. Esgotada a exploração do material, o professor dirige um determinado número de perguntas, explorando, sistematicamente, diferentes aspectos problemáticos a que o material dá lugar. Pode solicitar, em seguida, que os alunos representem - desenhando, pintando, escrevendo, fazendo cartunismo, teatralizando, etc. - o que elaboraram. A partir daí, discute-se a direção, a problemática, o material da(s) próxima(s) aula(s). (Becker, 1999, p. 95).

No exemplo apresentado pelo autor, o professor esgota os recursos, a exploração e permite que os sujeitos se apropriem do material da aula e interajam, a fim de construir conhecimento. O conceito é algo que será construído à medida que se estabelece uma relação entre os sujeitos que aprendem e os objetos do conhecimento.

Refutando os modelos pedagógicos anteriores, a pedagogia relacional entende que o conhecimento não pode ser transmitido e tampouco é fruto de herança genética; assim, a prática pedagógica contempla a interação, sendo que a escola é um espaço de trocas e de aprendizagem mútua, com espaços que privilegiem sempre a construção do conhecimento em detrimento da transmissão de conceitos prontos e de conteúdo. A seguir, discutiremos essa concepção de aprendizagem a partir dos conceitos elaborados na obra Piagetiana.

2.2 APRENDIZAGEM EM PIAGET

Piaget nasceu em Neuchâtel, na Suíça, em 1896. Biólogo de formação, desenvolveu, desde muito cedo, inúmeros estudos no campo da Biologia por meio do estudo de moluscos. Piaget possuía uma inclinação à pesquisa e a busca pelo conhecimento humano. Logo, seus

estudos deslocaram-se para o campo da Psicologia, em que buscou uma relação entre o desenvolvimento psicológico e o biológico. A sua teoria do conhecimento, batizada de Epistemologia Genética, objetivou descobrir como o conhecimento era construído pelos seres humanos e, especificamente, pelas crianças (PIAGET, 1978).

A epistemologia de Piaget não se preocupa com a validade do conhecimento, mas com sua origem e desenvolvimento. Ele se preocupa com a gênese e evolução do conhecimento, e salienta esse fato ao descrever seu campo de estudo como “epistemologia genética” (PAPERT, 1988. p. 195).

Os primeiros estudos de Piaget foram realizados com base na observação de seus filhos e, mais tarde, com crianças hospitalizadas. O primeiro laboratório em que Piaget pode observar as crianças diretamente em seu meio foi na Casa das Crianças, do Instituto Jean Jacques Rousseau (MUNARI, 2010). Esses estudos foram realizados por meio de testes empíricos e de observação e, mais tarde, no Instituto de Epistemologia Genética em Genebra, por meio do seu próprio método, chamado por ele de Método Clínico.

O método clínico piagetiano constitui-se como uma importante contribuição, não só como um método de investigação, mas também como um recurso para o professor que, através dele, pode levar o aluno a repensar suas hipóteses e construir conhecimentos.

[...] consiste sempre em conversar livremente com o sujeito, em vez de limitá-lo às questões fixas e padronizadas. Ele conserva assim, todas as vantagens de uma conversação adaptada a cada criança e destinada a permitir-lhe o máximo possível de tomada de consciência e de formulação de suas próprias atitudes mentais (PIAGET, 1982, p. 176).

Nesse sentido, o método clínico possibilita ao professor criar desequilíbrios no sujeito, buscando que o mesmo repense suas hipóteses e construa o conhecimento a partir da modificação de suas estruturas mentais.

Embora Piaget tenha apenas 3% de sua obra dedicada à educação, e ele mesmo tenha deixado claro que sua área de interesse era outra, o estudioso não poderia evitar a influência que a educação traria para seus estudos (MUNARI, 2010).

O fato de trabalhar no Instituto Jean-Jacques Rousseau, dedicado inteiramente ao desenvolvimento e ao aperfeiçoamento de sistemas de educação e de práticas educativas, e não mais em estabelecimentos hospitalares ou laboratórios médicos interessados na criança enferma ou deficiente, não podia deixar de exercer certa influência na consciência que Piaget tinha adquirido sobre a problemática da educação (MUNARI, 2010, p. 15).

A pesquisa de Piaget deixa uma importante contribuição à educação, por inscrever-se sob a gênese da aprendizagem. Nesse sentido, Piaget tornou-se um importante teórico da pedagogia moderna, ao fornecer aos educadores subsídios para o entendimento da aprendizagem e do desenvolvimento da inteligência humana.

Para Piaget (1978), a concepção de aprendizagem cunhada pela filosofia tradicional não era capaz de explicar por si só a aprendizagem. Assim, ele refuta as teorias Inatistas e Empiristas, compreendendo que o conhecimento não pode estar nem inerente ao ser humano, necessitando apenas de um estímulo para ser desenvolvido, tampouco transmitido diretamente para o aprendiz. Para Piaget, o conhecimento não é fruto da transmissão, sequer do acúmulo de informações, mas sim da ação do indivíduo com o meio, tanto físico como social.

Assim, a criança aprende por meio da ação sobre o mundo. Para Valentini (2003), Piaget, entende que:

Essa implicação do sujeito com o mundo é que permite a construção do conhecimento. Nesse olhar o objeto não é compreendido de forma independente do sujeito cognoscente, mas é o sujeito que vai significar o objeto a partir de suas estruturas (VALENTINI, 2003, p. 28).

Piaget, em oposição à transmissão, usa a metáfora da construção. Desse modo, o conhecimento, em sua visão, não pode ser transmitido, pois ele é fruto de uma leitura dos objetos exteriores que modificam as estruturas do pensamento.

Piaget configura-se como um teórico complexo, com vasta gama de conceitos que se inter-relacionam na busca pela explicação sobre a gênese do conhecimento. Por sua formação em biologia, ele criou uma metáfora a partir do que observava na natureza. Segundo o pesquisador, um organismo adapta-se a partir das modificações que o ambiente lhe proporciona. Piaget teve essa percepção, principalmente, ao observar que os moluscos adaptavam-se ao ambiente em que viviam. Partindo desse pressuposto, ele constrói o conceito de adaptação para o campo do conhecimento.

Os estudos de biologia fizeram-no suspeitar de que os processos de conhecimento poderiam depender dos mecanismos de equilíbrio orgânico; por outro lado, Piaget convenceu-se de que tanto as ações externas quanto os processos de pensamento admitem uma organização lógica (MACEDO, 1978, p. 8).

Para Piaget, a inteligência constitui-se por funções invariantes que iniciam no nascimento, ou antes dele, e permanecem ao longo de toda a vida, que ele denomina de invariantes funcionais da inteligência (PIAGET, 1970). As funções invariantes básicas são a

adaptação e a organização. A adaptação constitui-se da assimilação e da acomodação, sendo o aspecto mais externo da inteligência, ou seja, em que se evidencia a interação do sujeito com o mundo. A assimilação consiste em um processo cognitivo pelo qual o sujeito busca compreender a realidade a partir de suas estruturas internas (esquemas). Quando isso é insuficiente, é necessário o movimento de acomodação, que consiste em modificar seus esquemas ou criar novos para que possa dar conta do que o sujeito busca compreender. A assimilação e acomodação são complementares e, para Piaget, o ato inteligente é quando há um equilíbrio entre assimilação e acomodação.

Ainda menos pode dar azo a dúvidas que a vida mental também é acomodação ao meio ambiente. A assimilação nunca pode ser pura, visto que, ao incorporar os novos elementos nos esquemas anteriores, a inteligência modifica incessantemente os últimos para ajustá-los aos novos dados. Mas inversamente as coisas nunca são conhecidas em si mesmas, porquanto esse trabalho de acomodação só é possível em função do processo inverso de assimilação. Veremos, assim como a própria noção de objeto está longe de ser inata e necessita de uma construção ao mesmo tempo assimiladora e acomodadora. (PIAGET, 1970, p.18).

Assim, a aprendizagem consiste na modificação das estruturas mentais por meio da adaptação, essa que ocorre através de processos de assimilação e acomodação. Como já mencionado, para Piaget (1970), o processo de apropriação dos elementos do mundo consiste na assimilação. Em outras palavras, através de um processo cognitivo, o sujeito retira do meio as informações e as organiza, utilizando as estruturas mentais que já possui. Todo movimento de aprendizagem requer que o sujeito extraia do ambiente informações para modificar-se. Nas palavras de Macedo:

As estruturas da inteligência mudam através da adaptação a situações novas e têm dois componentes: a assimilação e a acomodação. Piaget entende o termo assimilação como a aceção ampla de uma integração de elementos novos em estruturas ou esquemas já existentes. A noção de assimilação, por um lado, implica a noção de significação e por outro expressa o fato fundamental de que todo conhecimento está ligado a uma ação e de que conhecer um objeto ou acontecimento é assimilá-lo a esquemas de ação. Em outros termos conhecer para Piaget consiste em operar sobre o real e transformá-lo, a fim de compreendê-lo em função do sistema de transformação a que estão ligadas todas as ações (MACEDO, 1978, p. 11).

Desse modo, aprender significa modificar os esquemas cognitivos a partir da leitura do novo. Para Piaget, esquemas são estruturas mentais que se modificam continuamente a partir da leitura do mundo, em outras palavras, esquemas são estruturas hipotéticas e dinâmicas que demonstram, em tese, a capacidade de reorganizar informações externas em novos conhecimentos.

Toda vez que um esquema não for suficiente para responder a uma situação e resolver um problema, surge a necessidade de o esquema modificar-se em função da situação (PIAGET, 1978, p. 11).

Portanto, esquemas e os processos de adaptação, que se dão por meio de assimilação e acomodação, são processos intimamente ligados por meio de uma relação de complexidade. A organização, como explicitado anteriormente, é uma das funções invariantes funcionais da inteligência, inseparável da adaptação, mas se constituindo como o aspecto interno do mecanismo da inteligência.

Diante desses conceitos, Piaget (1975) propõe um mecanismo regulador que ele explica a partir da Teoria da Equilibração.

Equilibração é processo de passagem do desequilíbrio para o equilíbrio. Este é um processo auto-regulador cujos instrumentos são assimilação e acomodação. A equilibração permite que a experiência externa seja incorporada na estrutura interna (esquemas). Quando ocorre o desequilíbrio, ele proporciona motivação para a criança buscar o equilíbrio – para depois assimilar ou acomodar. O desequilíbrio ativa o processo de equilibração e o esforço para retornar ao equilíbrio. O equilíbrio é uma condição necessária pela qual o organismo luta, constantemente. O organismo finalmente assimila todos os estímulos (ou eventos) com ou sem acomodação. Isto resulta em equilíbrio. Então o equilíbrio pode ser visto como um estado de “balanço” cognitivo que é alcançado no momento da assimilação (WADSWORTH, 1996, p. 25).

Em outras palavras, a equilibração é um processo constante e inerente aos processos de aprendizagem e que é fundamental para a construção de novos conhecimentos. Nessa perspectiva, o fundamental para promover novas aprendizagens é o desequilíbrio que impele o sujeito a ultrapassar seu entendimento atual. O processo de adaptação, através da acomodação e assimilação, está presente na Teoria da Equilibração. Nesse processo a acomodação, diferentemente da assimilação, consiste em um processo de transformação das estruturas mentais existentes em novas estruturas, ou na criação de uma nova estrutura, sendo essa determinada pela atividade do sujeito em relação ao objeto. Em um movimento dialético vemos que para que ocorra a acomodação, é preciso que antes ocorra o que Piaget (1975) chamou de desequilibração:

Em uma perspectiva da equilibração, deve-se procurar nos desequilíbrios uma das fontes de progresso no desenvolvimento dos conhecimentos, pois só os desequilíbrios obrigam um sujeito a ultrapassar seu estado atual e procurar seja o que for em direções novas (PIAGET, 1975).

A metáfora da construção do conhecimento na obra de Piaget se dá a partir desses movimentos que, de maneira alguma, referem-se à transmissão de conceitos e sim à criação ou ampliação de esquemas mentais. Esse movimento é constante e vai construindo uma vasta gama de estruturas mentais que se relacionam ao que o sujeito compreende do mundo. Os movimentos de assimilação e acomodação são constantes e complementares, fundamentais para o processo de construção do conhecimento.

Embora breve, trazemos a teoria da equilibração por entender que ela explicita uma forma de aprender em que o sujeito é provocado pelos desequilíbrios e que necessita da ação do sujeito para que a aprendizagem aconteça.

Ainda a partir da abordagem Piagetiana, podemos compreender a importância da cooperação no processo de aprendizagem. Para Piaget, a cooperação pode ser entendida como moral e intelectual, e essas são indissociáveis. Dito de outra forma, para cooperar é necessário colocar-se no ponto de vista do outro, denominado por Piaget de descentração e, também, é necessária a reciprocidade, entendida como considerar o ponto de vista do outro.

Para Piaget (1998), a cooperação está relacionada, entre outros aspectos, com o desenvolvimento intelectual e moral. “O segundo depende do desenvolvimento do primeiro, tendo como uma condição necessária” (CAMARGO E BECKER, 2012, p. 528). A cooperação na obra piagetiana é um constructo bastante peculiar, pois, para Piaget, no âmbito das relações sociais, a cooperação depende de prerrogativas que estão intimamente ligadas à moral. Na perspectiva piagetiana cooperar é uma relação de trocas entre iguais e, nesse sentido, só pode ocorrer entre sujeitos com um mesmo prestígio social. Assim, para que haja cooperação, é preciso que não exista uma relação de coerção, ou seja, uma relação unilateral de respeito. Piaget (1994) propõe a necessidade de distinção entre coação e cooperação. “É preciso distinguir, em todos os domínios dois tipos de relações sociais. A coação e a cooperação, a primeira implicando em um elemento de respeito unilateral, de autoridade, de prestígio, a segunda uma simples troca de indivíduos iguais” (PIAGET, 1994 p. 58). Em outras palavras, a cooperação é uma relação de reciprocidade entre sujeitos iguais. Camargo e Becker (1993, p. 532) definem a cooperação como:

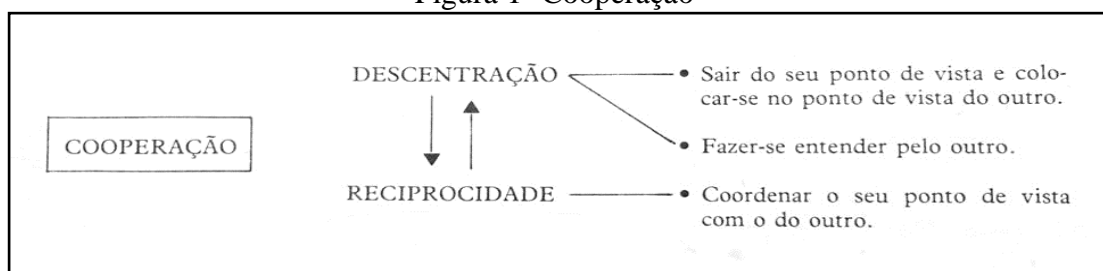
1. Uma relação que se fundamenta na reciprocidade, por considerar os sujeitos envolvidos na interação como iguais (assumindo-se sem hierarquia).
2. Uma situação ideal de relação que exige dos sujeitos um desenvolvimento intelectual e moral capaz de possibilitar a dissociação do pensamento de si do pensamento do outro com quem se relaciona. Essa dissociação, e o respeito mútuo que se possibilita a partir dela, é a lógica das relações de cooperação.

Para Piaget (1998), cooperar significa abrir mão do ponto de vista do sujeito para que então se possa conhecer o ponto de vista dos demais. A ideia de cooperação nos é importante, pois permite entender de que forma os sujeitos compreendem e se relacionam com o outro. Kebach (2007) explica essa relação da seguinte forma:

Por exemplo, somente quando as crianças conseguem realmente compreender as regras de um jogo simples, como o de dominó, é que estarão aptas para realizar condutas sociais baseadas nos princípios de coordenação de pontos de vista e cooperação, portanto de descentramento, buscando coordenar suas ações com a de outras crianças. Antes disso, por falta de compreensão do sistema de regras, agem por conta própria, pensando que suas ações estão sendo compreendidas pelos outros (KEBACH, 2007. p. 51).

Nesse sentido, a capacidade de cooperar está relacionada com a capacidade de descentração, ou seja, de compreensão do ponto de vista do outro. Essa capacidade ocorre a partir do momento em que a criança é capaz de se colocar na situação do outro, para que assim possa entender um problema sob uma ótica extrínseca a ela. Rangel (1992. p. 68) explica que “descentração é a capacidade da criança poder ‘sair’ do seu ponto de vista, inicialmente egocêntrico, e se colocar no ponto de vista dos outros; é a capacidade da criança criar novos argumentos capazes de se fazer entender pelo outro.” A imagem a seguir demonstra essa relação:

Figura 1- Cooperação



Fonte: Rangel, 1992, p. 69.

Rangel (1992) ainda esclarece que os sujeitos constroem suas regras e valores dentro de um grupo social e que essas trocas baseiam as relações de construção do conhecimento a partir dos conflitos surgidos por meio de trocas com o grupo e da modificação das hipóteses.

Nesse caminho, Piaget ainda postula dois tipos de respeito, o unilateral e o mútuo. Enquanto o primeiro diz respeito diretamente à relação de maior prestígio de um em detrimento de outro, o segundo se configura em uma relação entre iguais. Nas palavras de Piaget:

[...] pode-se igualmente falar de respeito num segundo caso, quando dois indivíduos se respeitam mutuamente. Este respeito “mútuo”, longe de implicar um

constrangimento espiritual, como o primeiro tipo, constitui o ponto de partida da “cooperação” (PIAGET, 1998, p. 60).

Ao explicar essa relação, o autor propõe a existência de dois conceitos correspondentes, a solidariedade externa e a solidariedade interna. A primeira corresponde ao respeito unilateral, em que os sujeitos aceitam uma regra necessariamente externa, ou seja, partem que a regra foi criada para ser obedecida e está relacionada com o prestígio de quem a criou. Enquanto a segunda, a solidariedade interna, corresponde à capacidade dos sujeitos em criar as suas próprias regras. Ou seja, modificar as regras é uma capacidade de sujeitos que são capazes de cooperar, pois conseguem entender o ponto de vista do outro e, assim, compartilhar um ponto de vista comum no grupo. Piaget esclarece essa questão ao diferenciar o comportamento das crianças:

Totalmente diferente é a forma de se comportar das crianças mais velhas. Nestas, o prestígio dos mais velhos cede lugar à necessidade de acordo mútuo. A solidariedade já não reside na participação comum numa realidade transcende, mas na vontade comum de respeitar decisões tomadas. (PIAGET, 1998, P. 61).

Destacamos a cooperação, considerando importante esse conceito para entender as relações de interação e trocas que envolvem a construção do conhecimento. Os conceitos de Piaget trazidos até aqui são importantes para encaminhar, do ponto de vista teórico, a análise que será realizada nos capítulos finais. Dessa forma, iniciamos com Piaget e seguimos um percurso teórico que se articula com outros construtivistas, como Papert, Resnick e Fagundes.

2.3 PAPERT VIDA E OBRA

Seymour Papert viveu em um período de transformações, tanto sociais como tecnológicas e políticas. Nasceu em primeiro de março de 1928, em Pretória, capital executiva da África do Sul, hoje chamada de Tshwane, onde seu pai, um biólogo especialista em insetos, trabalhava. Papert obteve uma educação excêntrica, uma vez que a família Papert passou boa parte de sua infância vagando pela costa leste do sul da África, à procura de insetos (BOYLE, 2004).

Como muitos meninos da sua idade, demonstrava curiosidade e interesse pelo funcionamento de carros.

Antes dos meus dois anos eu já me interessava por automóveis [...] sentia-me muito orgulhoso por conhecer os componentes do sistema de transmissão, caixa de câmbio e especialmente o diferencial. Isso aconteceu, é claro, muito antes de eu entender como as engrenagens funcionavam; mas assim que passei a conhecê-las brincar com elas passou a ser meu passatempo favorito (PAPERT, 1988, p. 11).

Papert descreve que o interesse por engrenagens o fez entender mais tarde a matemática. Essa curiosidade certamente foi o germe do que lhe tornaria um grande pesquisador no futuro. Nos tempos de infância, ele se deparou com os conflitos da sociedade africana, tendo problemas em aceitar, desde pequeno, o *Apartheid*. Mais tarde, essa infância refletiu em suas ações, tornando-se um ativista contra o movimento de segregação racial. Esse ativismo lhe rendeu, inclusive, a reprovação do visto americano por 10 anos (BOYLE, 2004).

Na vida acadêmica, graduou-se em filosofia no ano de 1949 e, em seguida, em 1952, cursou um doutorado em Matemática, ambos na universidade de Witwatersrand, na África do Sul. Algum tempo depois, mudou-se para o Reino Unido, onde ganhou uma bolsa de estudos da Commonwealth para St. John's College em Cambridge e lá cursou seu segundo doutorado, também em matemática, concluído em 1959. Esteve na França por alguns anos e teve uma das grandes oportunidades de sua vida ao trabalhar no Centro Internacional de Epistemologia Genética, na Universidade de Genebra, ao lado de Jean Piaget (SCHOFIELD, 2016). Durante os mais de quatro anos que se passaram, o contato com o grande epistemólogo lhe proporcionaria, mais tarde, a construção de sua própria teoria de aprendizagem, a qual Papert chamaria de Construcionismo. Escreveu ele:

Todo mundo interessado em como a criança pensa tem um débito geral e imenso para com Jean Piaget. Eu também tenho um débito especial. Se Piaget não tivesse entrado na minha vida eu seria agora um “matemático de verdade” em vez de ser o que quer que seja que eu tenha me tornado. Piaget investiu muita energia e muita fé em mim. Espero que ele reconheça que o que eu tenho contribuído para o mundo da criança está dentro do espírito do empreendimento de sua vida (PAPERT, 1988, p. 252).

Além do legado na educação, Papert desenvolveu alguns trabalhos no campo da Inteligência Artificial. Ele conheceu Marvin Minsky em 1960, em uma conferência em Londres e, no ano de 1963, mudou-se para os EUA. Com Minsky, escreveu o livro “*Perceptrons*”, obra que trata do desenvolvimento de inteligência artificial com base em redes neurais. Devido à mudança, passou a trabalhar no Instituto de Tecnologia de Massachusetts e, em 1985, foi co-fundador do Media Lab, ao lado de Minsky e Negroponte (SCHOFIELD, 2016), lugar em que ele desenvolveu a maior parte de seus projetos em informática na educação, trabalhando com grandes pesquisadores. Ele escreveu:

Este livro e toda a minha vida profissional ganharam muito com a oportunidade de trabalhar no MIT Media Lab, cuja criação, por Nicholas Negroponte, representa uma experiência significativa e original na construção de um ambiente apropriado para o florescimento de uma disciplina, ainda embrionária [...] (PAPERT, 2008, p. 8).

Embora Papert tenha transitado por diversas áreas, desde a filosofia à matemática, sua maior contribuição foi, sem dúvida, a introdução dos computadores na educação. Sua obra máxima foi publicada em 1980, sob o título de *Mindstorms: Children, Computers e Powerful Ideas*, nos EUA, publicado no Brasil em 1985 com o título *Logo: Computadores e Educação*. Nela, Papert (1980) apresentou a sua forma inovadora de aprender com o computador e deu luz à sua teoria Construcionista. Em seu livro, o autor introduz a sua visão sobre o uso do computador como um recurso de aprendizagem, discutindo de que forma o computador poderia modificar a maneira como as pessoas pensam e se relacionam com o conhecimento. Além disso, percebe que a maneira que o computador estava sendo inserido na educação - apenas como uma máquina de programar os usuários - estava muito aquém das verdadeiras contribuições que o computador poderia trazer para a educação. Assim, nessa obra, critica a maneira que o computador vinha sendo utilizado na educação, em especial, nas escolas dos Estados Unidos. Em suas palavras:

Na minha perspectiva, é a criança que deve programar o computador e, ao fazê-lo, ela adquire um sentimento de domínio sobre um dos mais modernos e poderosos equipamentos tecnológicos e estabelece um contato íntimo com algumas das ideias mais profundas da ciência, da matemática e da arte de construir modelos intelectuais. (PAPERT, 1988, p. 18).

A crítica de Papert baseava-se no fato de nos anos 1980 o computador estar sendo incluso nas escolas como um instrumento de ensino em uma perspectiva instrucionista, ou seja, de instruir, de condicionar. Foi por conta desta oposição que Papert batiza sua teoria de aprendizagem de Construcionismo.

No ambiente LOGO a relação é inversa: a criança, mesmo em idade pré-escolar, está no controle – a criança programa o computador. E ao ensinar o computador a “pensar”, a criança embarca numa exploração sobre a maneira como ela própria pensa. Pensar sobre modos de pensar faz a criança tornar-se um epistemólogo, uma experiência que poucos adultos tiveram (PAPERT, 1988, p. 35).

Nesse caminho, ele percebe algo que não havia sido percebido por outros, a potencialidade do computador como um recurso para a criança aprender, não apenas a programar o computador, mas aprender a aprender e desenvolver competências em outras ciências por meio da programação. Papert (1988) percebeu no computador um recurso de

aprendizagem e de viabilização da aprendizagem formal, uma vez que, por meio dos modelos computacionais, o aprendiz poderia vivenciar questões que antes não seriam possíveis, formando uma ponte entre o pensamento concreto e o pensamento formal.

Por ser matemático de formação, uma das preocupações de Papert (1988, 2008), em suas obras, sempre foi a aprendizagem da matemática. Ele observou que as crianças poderiam aprender matemática naturalmente se estivessem em um ambiente que lhes propiciasse essa aprendizagem. Assim, em sua visão, a metodologia LOGO proporcionaria a aprendizagem da matemática como se a criança estivesse na Matelândia. Na visão de Papert (1988, 2008), aprender matemática no mundo da matemática seria naturalmente mais fácil, assim como aprender francês na França, por exemplo.

Papert veio ao Brasil várias vezes e construiu interlocuções com grandes estudiosos brasileiros como Paulo Freire, José Armando Valente e Léa da Cruz Fagundes. Em uma de suas visitas ao Rio de Janeiro, surpreendeu-se com o modelo de aprendizagem desenvolvido pelas escolas de samba, um modelo natural, sem distinções de faixas etárias ou currículos e comparou esse modelo de aprendizagem - é claro, com algumas ressalvas - ao modelo de aprendizagem LOGO (PAPERT, 1988).

Os estudos de Papert foram base para projetos em diversos países que visavam à introdução do computador na educação. Em especial, no Brasil ele motivou uma série de projetos em parceria com a UNICAMP e UFRGS. Papert veio ao Brasil algumas vezes e, inclusive, participou do Congresso Internacional de LOGO no Brasil em 1986, ao lado de seus seguidores brasileiros. A figura 2 mostra Léa Fagundes, José Armando Valente, Seymour Papert e outros participantes do congresso.

Figura 2 - Léa Fagundes e Seymour Papert



Fonte: <http://www.educadigital.org.br/biodaleafagundes/portfolio-view/por-mirian-gregori-congresso-internacional-logo-no-brasil-1986>.

Nesse caminho, Papert ainda contribuiu significativamente para a robótica educacional que conhecemos atualmente. Uma parceria entre os criadores da linguagem LOGO e a empresa de brinquedos LEGO deu origem, em 1998, ao *Lego Mindstorms*, uma gama de brinquedos que aliou a tecnologia dos brinquedos LEGO ao potencial de programação da linguagem LOGO. (SCHOFIELD, 2016). Do mesmo modo, o famoso ambiente de programação *Scratch* nasceu a partir da junção dos princípios dos brinquedos LEGO (conectar e montar) e a linhagem de programação LOGO, resultado da parceria de Papert com Michel Resnick. Assim, as contribuições de Papert estão vivas em inúmeras plataformas e brinquedos de robótica usados nas escolas e por crianças no mundo todo. Ao que parece, Papert tinha um objetivo maior, que seria tornar a linguagem desenvolvida por ele a base para outras tecnologias de aprendizagem em um futuro próximo. Em “Logo: Computadores e Educação”, ele escreveu: “Vejo-a como instrumento educacional válido, mas sua principal função é servir como modelo para outros objetos ainda a serem inventados”. (PAPERT, 1988, p. 26). As contribuições de Papert, pensadas ainda nos anos 1960, forneceram os primeiros passos para as tecnologias que existem atualmente. Essas tecnologias permitem às crianças possibilidades de criação nunca antes vista na história da educação.

Papert teve uma vida brilhante, porém, sua produção acadêmica cessou depois de um grave acidente em Hanói, Vietnã, em 2006, do qual nunca mais se recuperou totalmente. Morreu no dia 31 de julho de 2016. Em sua memória, o Mit Media Lab (2017) escreveu: “Seymour Papert, cujas ideias e invenções transformou como milhões de crianças ao redor do mundo criam e aprendem, morreu no domingo, 31 de julho de 2016.”.

2.4 O CONSTRUCIONISMO DE PAPERT

A teoria Construcionista de Papert (PAPERT, 2008) é subjacente ao construtivismo piagetiano, porém, sob a ótica construcionista, a construção do conhecimento aconteceria por meio do fazer. Lopes (2010, p. 53) acrescenta que o construcionismo de Papert avançou em uma direção afetiva: “Pode-se afirmar que Papert, em sua preocupação com o afetivo, procurou ir além de Piaget no sentido de procurar os elementos da aprendizagem que vão além da cognição.”.

Como Piaget, Papert percebe que o conhecimento não é, nem empírico, nem inato, mas sim construído à medida que o indivíduo conhece o mundo e modifica as suas estruturas mentais. Segundo ele:

[...] nossa interpretação da teoria piagetiana estabelece três pontos. Primeiro, fornece uma teoria psicológica específica, altamente competitiva, por sua parcimônia e poder de explicação, com outras na área. Segundo ela nos mostra o poder de um princípio computacional específico, neste caso a teoria de procedimentos puros, ou seja, modular. Terceiro, ela concretiza meu argumento sobre como linguagens diferentes podem influenciar as culturas que crescem ao redor dessas linguagens (PAPERT, 1988, p. 204).

Dentro da epistemologia Piagetina, como já discutido anteriormente, há duas formas de pensar o mundo, quais sejam: em primeiro lugar, o pensamento concreto, que corresponde ao que a criança pensa sobre o concreto e, em segundo lugar, o pensamento formal, que tem suas origens no que se pensa de maneira abstrata, ou seja, o pensamento formal constitui-se a partir de abstrações cognitivas, de algo que não necessariamente existe no mundo concreto.

Nesse campo de complexidade, Papert (1988, 2008) percebe uma maneira de transformar o formal em concreto, pois ele enxerga no computador um potencial de criação de modelos que permitem ao aprendiz conhecer a partir da transformação do formal em concreto. Se trouxermos o que Papert pensou para um contexto atual, um exemplo seria entender uma molécula de qualquer substância por meio de computação gráfica e de animações que permitam uma experiência que antes dos computadores não seria possível. Assim, pensar sobre uma molécula é um pensamento formal, mas por meio do computador esse pensamento converte-se em pensamento sobre o concreto, uma vez que a molécula passa a existir e ser internalizada de outra maneira. Nas palavras de Papert:

Minha suposição é que o computador pode concretizar (e personalizar) o formal. Sob este prisma, o computador não é somente mais um instrumento educacional poderoso. Ele é o único a nos permitir aos meios para abordar o que Piaget e muitos outros identificam com o obstáculo que deve ser transposto para a passagem do pensamento infantil para o pensamento adulto. Eu acredito que o computador pode nos permitir mudar os limites entre o concreto e o formal. Conhecimentos que só eram acessíveis através de processos formais podem agora ser abordados concretamente. (PAPERT, 1988, p. 37).

Nesse caminho, o Construcionismo percebe no computador um importante recurso de aprendizagem, uma vez que é capaz de viabilizar simulações e a criação de relações sobre o pensamento formal, que seriam exclusivos da abstração do indivíduo. No campo da computação, por exemplo, diversos conceitos, como os de estruturas de dados, são estritamente formais e de difícil aprendizagem. O conceito de pilhas e filas, e até mesmo vetores ou ainda variáveis, são conceitos que, sem o uso de simulações e de modelos computacionais, são estritamente formais.

Além disso, Papert (1980, 2008) entende a aprendizagem como um movimento do indivíduo, um percurso de construção que ocorre por meio da vontade do sujeito em conhecer, explorar e descobrir. Nas palavras do autor, “A atitude construcionista no ensino não é, em absoluto, dispensável por ser minimalista – a meta é ensinar de forma a produzir maior aprendizagem a partir do mínimo de ensino” (PAPERT, 2008, p. 134). Nesse sentido, o construcionismo entende o papel do professor como um mediador do processo de aprendizagem. Papert (2008) usa uma metáfora ao explicar que o professor deve ensinar a pescar, e não dar os peixes. Ele argumenta que a aprendizagem ocorre em inúmeros espaços e que a aprendizagem escolar é artificial. Em sua visão, as crianças aprendem naturalmente diversos tipos de conhecimento; um exemplo são os videogames que possuem desafios de grande complexidade e os jovens aprendem a jogar sem a necessidade de um ensino formal ou de um professor especialista em jogos.

O construcionismo proposto por Papert compartilha da visão proposta pelo construtivismo de Piaget – de aprendizagem a partir da construção de estruturas do conhecimento através de que isso acontece especialmente e felizmente num contexto no qual o aprendiz está conscientemente engajado em construir uma entidade pública, seja um castelo de areia numa praia ou uma teoria sobre o universo (LOPES, 2010, p. 55).

Nesse sentido, Papert (2008) esclarece sua visão de aprendizagem ao compartilhar da posição defendida por Piaget sobre o conhecimento não poder ser transmitido, mas sim construído:

As metáforas “Transmissão” versus “construção” são temas que permeiam um movimento educacional maior e mais diversificado dentro do qual situo o construcionismo e resalto isso pelo jogo de palavras no nome. Para muitos educadores e para todos os psicólogos cognitivos minha palavra evocará o termo Construtivismo, cujo uso educacional contemporâneo em geral remete à concepção de Piaget que o conhecimento simplesmente não pode ser “transmitido” ou “transferido pronto” para outra pessoa. Mesmo quando parece estarmos transmitindo com sucesso informações dizendo-as, se pudéssemos ver os processos cerebrais em funcionamento, observaríamos que nosso interlocutor está “reconstruindo” uma versão pessoal das informações que pensamos estar “transferindo” (PAPERT, 2008, p. 137).

Assim, o Construcionismo (PAPERT, 2008) é uma reinterpretação da teoria construtivista de Piaget, uma vez que o autor propõe uma concepção mais concreta sobre a maneira como se constrói o conhecimento. Ao definir Construcionismo, Papert (2008, p. 137) explica o seguinte: “Ele atribui especial importância ao papel das construções no mundo como um apoio para o que ocorre na cabeça tornando-se assim uma concepção menos mentalista.”.

Nesse sentido, o construcionismo tem uma dimensão de significativa importância para esta pesquisa, uma vez que, a partir dele é possível entender como a aprendizagem se desenvolve por meio da ação. Logo, entendemos que o desenvolvimento do pensamento computacional, o ensino de programação e a aprendizagem de Ciência da Computação se beneficiam dessa visão de aprendizagem, já que permitem a aprendizagem por meio da ação e da exploração.

3 PENSAMENTO COMPUTACIONAL

“Quando você aprende através da escrita de códigos, e escreve códigos para aprender, você está aprendendo em um contexto significativo, e esta é a melhor forma de aprender as coisas.”

Mitchel Resnick

Neste capítulo, apresento um breve histórico dos computadores na educação e do ensino de programação como prática de informática na educação desde a invenção dos primeiros computadores. Em seguida, apresento a construção teórica acerca do termo Pensamento Computacional, buscado explicitar sua origem e os conceitos por trás do termo, bem como as convergências e divergências da comunidade científica acerca do assunto. Na sequência, exponho uma discussão a respeito dos trabalhos publicados recentemente sobre pensamento computacional, em que mostro a situação da pesquisa acerca do pensamento computacional no cenário brasileiro e a necessidade de se discutir a formação de professores. Por fim, aponto alguns ambientes para o desenvolvimento do pensamento computacional que fornecem algum recurso para o professor.

3.1 UM BREVE HISTÓRICO DO PC NA EDUCAÇÃO

Enquanto nos primeiros anos do século XX alguém poderia ouvir as notícias do mundo em um rádio, ao final desse século, um número muito superior de informações estaria disponível no computador pessoal de milhões de pessoas. Já no início do século XXI, informações do mundo todo, bem como a conexão entre pessoas, estariam disponíveis na palma da mão de um usuário de *smartphone*. Castells (2002) argumenta que o século XX foi, sem dúvida, um período de extremas transformações, sobretudo pelas mudanças tecnológicas que modificaram a vida de todos.

Chamado por Hobsbawm (1995) de “A era das Catástrofes”, esse período contemplou transformações na ciência nunca antes vistas. Foi o cenário de instabilidade entre as duas Grandes Guerras e a Guerra Fria que possibilitaram avanços em muitos campos, como a descoberta da Física Nuclear, do átomo, dos transistores e de muitas tecnologias que revolucionaram a vida humana. Escreveu Hobsbawm (1995): “Nenhum período da história foi mais penetrado pelas ciências naturais nem mais dependente delas do que o século XX” (p. 505). Na visão desse historiador, a emergência da computação permitiu à ciência experimentações que antes não seriam humanamente realizáveis.

Dependente de inúmeros estudos da matemática, desde a antiguidade, a história da computação esteve associada à história da matemática, pois a evolução de diversos conceitos matemáticos, desde os mais básicos, como o do zero, subsidiou a evolução da computação moderna (FONSECA FILHO, 2007). Assim, não há como se estudar a história da computação sem antes relacioná-la com a evolução da matemática desde o princípio.

Dentre as inúmeras transformações do século XX, o surgimento da computação e a maneira como ela permeou os diversos setores da sociedade merecem destaque. As ressonâncias da tecnologia e do aparecimento do computador pessoal também implicaram transformações no espaço escolar. Em algum tempo, o computador passou a ser pensado como ferramenta de aprendizagem, estando presente também nas escolas (VALENTE, 1999).

“A utilização de computadores na educação é tão remota quanto o advento comercial dos mesmos” (VALENTE, 1999, p. 1). Desde o início da computação, havia a intenção de usá-los na educação, contudo, a abordagem de outrora se distingue da maneira como a informática na educação se desenvolveu e chegou à atualidade. Nos anos 1950, por exemplo, o uso do computador estava ligado à armazenagem de informação e na execução de processos.

Em 1967, Seymour Papert (1988) idealizou o uso do computador na educação por meio da criação da Linguagem LOGO. Pensada como uma necessidade para todos, Papert (1988) estava muito à frente de seu tempo, sendo pioneiro em pensar uma educação mediada pelo uso do computador. Nesse período, contudo, a tecnologia que Papert necessitava ainda não estava disponível para todos e, principalmente, para as escolas. Seria necessária a invenção do computador pessoal para a popularização da informática na educação.

O aparecimento dos microcomputadores, principalmente o *Apple*, permitiu uma grande disseminação dos computadores nas escolas. Essa conquista incentivou uma enorme produção e diversificação [...], como tutoriais, programas de demonstração, exercício-e-prática, avaliação do aprendizado, jogos educacionais e simulação (VALENTE, 1999, p. 3).

Na década de 1980, as ideias de Seymour Papert passaram a ser aplicadas, e o papel do computador colocou-se em evidência. Um marco na informática na educação foi, sem dúvida, a publicação da obra *Mindstorms: Children, Computers and Powerful Ideas* (PAPERT, 1980), em que Papert trouxe a metodologia LOGO como uma nova forma de conceber a aprendizagem.

Conforme já discutido no capítulo anterior, Papert (1980) faz uma crítica sobre a maneira como os computadores faziam parte da educação estadunidense, apenas como mais um conteúdo a ser ensinado dentro de um currículo formal. Logo, os computadores estavam na

escola como uma disciplina a ser ensinada e com conteúdos a serem vencidos, não apenas nos EUA, mas também em outros países. Para Valente (1999) e Papert (2008), introduzir a computação como disciplina seria a maneira mais simples de implementar o currículo sem demandar a necessidade de se repensar a educação. Nesse sentido, esses autores propuseram o uso da informática na educação, não como um elemento do currículo, mas como um recurso que possibilitasse o aprendizado de outras ciências de maneira natural, como é o caso da matemática.

É nesse sentido que a informática na educação, no Brasil, passa a ser introduzida, como uma alternativa para contribuir com o aprendizado de outras disciplinas. À frente desse movimento, estavam o professor José Armando Valente, da Universidade Estadual de Campinas (UNICAMP), e a Professora Léa da Cruz Fagundes, da Universidade Federal do Rio Grande do Sul (UFRGS). Ambos tiveram importante papel no desenvolvimento de formações e políticas que estimularam a disseminação da informática na educação nas escolas brasileiras. Assim, o computador, ao final da década de 1980, estaria inserido em algumas escolas públicas e privadas, no cenário brasileiro. Valente (1999) explica que o surgimento da informática na educação no Brasil remonta à década de 1970, quando educadores brasileiros, motivados pelo que ocorria nos Estados Unidos e na França, buscaram introduzir a informática na educação brasileira.

Na visão de Papert (1988, 2008), conhecer uma linguagem de programação demanda também conhecimento lógico matemático. Nessa visão, o aluno, ao aprender uma linguagem de programação, desenvolveria naturalmente os conceitos matemáticos.

De acordo com Valente (1999), a evolução dos computadores permitiu que, nos anos 1990, o computador estivesse totalmente difundido nas escolas estadunidenses, ainda mais depois do advento da *internet*, que possibilitou uma maior gama de possibilidades no campo da aprendizagem.

Entretanto, a introdução do ensino de códigos de programação como alternativa interdisciplinar e de desenvolvimento de habilidades do currículo ficou, de certa forma, esquecida. Valente (2016) argumenta que, nos anos 1980, a linguagem LOGO era praticamente a única forma de introdução do computador nas escolas, porém, com a grande evolução dos *softwares* e a popularização da informática, que ocorreu a partir dos anos 1990, os novos recursos deixaram a antiga linguagem de programação no passado. Afinal, um mundo de tecnologia se abriu, com *softwares* de autoria, multimídia e inúmeros recursos que eram muito mais atrativos aos alunos e professores. Nesse caminho, o processo de introdução da informática nas escolas brasileiras se desenvolveu com a distribuição de computadores e

construção de laboratórios de informática e, recentemente, com a distribuição de *laptops* em projetos experimentais, por meio de políticas públicas como, por exemplo, o “Um Computador por Aluno” (UCA).

Recentemente, uma série de projetos voltados ao ensino de códigos de programação tem levantado discussão sobre a necessidade de todos em aprender conceitos de Ciência da Computação. Um movimento iniciado por Jeannette Wing (2006) fomenta a disseminação do “Pensamento Computacional” como uma necessidade de todos e uma competência a ser desenvolvida pelo cidadão do século XXI, desde o ensino básico.

3.2 PERCURSO DO CONCEITO DE PENSAMENTO COMPUTACIONAL

A invenção do computador moderno, por volta dos anos 1940, provocou uma revolução no campo da informação, desencadeando mudanças significativas na maneira como as pessoas viviam. Esse movimento transformou as formas de trabalho e também a maneira de ensinar e aprender. A computação permeou as inúmeras áreas do conhecimento da agricultura à medicina, da metalurgia à engenharia, das salas de aula às corporações.

Nesse contexto de mudanças, essa ciência ocupou um lugar privilegiado no mundo do trabalho e, atualmente, ganhou espaço de discussão na escola, trazendo a necessidade de uma nova competência para o cidadão do século XXI: O Pensamento Computacional (PC).

Conforme já discutido, desde os anos 1960 Seymour Papert (1980, 1988, 2008) percebeu que as linguagens de programação não poderiam servir apenas para cientistas da computação; elas deveriam fazer parte da educação de todos.

Papert influenciou uma série de práticas de introdução da informática na educação na perspectiva do ensino de códigos. No Brasil, diversas escolas trabalharam com essa abordagem, por meio da Linguagem Logo, com *softwares* que permitiam a manipulação de uma tartaruga a partir de códigos de programação. “Embora essa realidade tenha sido bastante específica, uma vez que o acesso aos computadores demandava investimentos e políticas públicas” (VALENTE, 1999; ALMEIDA, 2008).

Ao que parece, Papert foi pioneiro em utilizar o termo “*Computational Thinking*” em seu livro *Mindstorms, Children, Computers and Powerful Ideas* (PAPERT, 1980, p. 182), trazendo uma visão bastante semelhante à proposta atualmente difundida sobre o pensamento computacional. Embora o termo tenha, ao que tudo indica, aparecido na literatura neste tempo, sabe-se, contudo, que o estudo do pensamento computacional existe desde a invenção da computação moderna.

[...] inventei maneiras de tirar vantagens educacionais da oportunidade de dominar a arte de deliberadamente pensar como um computador, de acordo, por exemplo, com o estereótipo de um programa que é executado de maneira seriada, literal e mecânica. Há situações em que esse estilo de pensamento é apropriado e útil. Algumas das dificuldades das crianças em aprender matérias formais como gramática ou matemática devem-se à sua incapacidade de entender a utilidade deste estilo de pensamento (PAPERT, 1988, p. 44).

Nesse sentido, ao Papert explicitar que a criança poderia “Pensar como o Computador” e, mais tarde, referir-se a este pensamento como “Pensamento Computacional” (PAPERT, 1980, p. 182) não há como negar a sua influência como precursor do conceito de Pensamento Computacional. Contudo, em sua obra, o autor não trouxe o atual destaque ao termo e, desse modo, ele não tem aparecido na literatura como um autor de referência nas discussões atuais sobre PC.

Ao trazer a programação como um recurso de pensar como um computador, Papert (1988) ainda acrescenta que há vantagens ao se “pensar como um computador”, uma vez que esse se torna um recurso cognitivo para a resolução de problemas.

Neste livro tenho argumentado claramente que o pensamento de procedimento é uma ferramenta intelectual poderosa e até sugeri fazer analogia de si mesmo com o computador como uma estratégia para ajudar a conseguir isto. As pessoas temem frequentemente que usar modelos computacionais para pessoas levará a um pensamento mecânico ou linear (PAPERT, 1980, p. 186).

O autor ainda argumenta que pensar como o computador não significa pensar apenas mecanicamente, e defende a abordagem de um pensamento que acrescente às pessoas maior capacidade na resolução de problemas, por meio do pensamento computacional. Em suas palavras:

O conselho “pense como um computador” pode ser entendido como significando que sempre se deva pensar sobretudo como um computador. Isto seria restritivo e limitativo. Mas o conselho poderia ser entendido num sentido muito diferente, não excluindo nada, mas fazendo um poderoso acréscimo à coleção de ferramentas mentais de uma pessoa. [...] Na minha experiência, o fato de eu pedir a mim mesmo para “pensar como um computador” não exclui outras epistemologias. Simplesmente abre outros caminhos para abordar a reflexão (PAPERT, 1980, p. 187).

Assim, entendemos que o termo “pensamento computacional” nasceu nos anos 1980 a partir dos estudos de Papert e tomou novas dimensões recentemente. O conceito recente de pensamento computacional emergiu por meio do texto “Computational Thinking”, em que Jeannet Wing (2006) apresentou a importância da discussão do PC como uma competência para

todos, e não apenas para cientistas da computação. De acordo com a autora, enquanto a Ciência da Computação trata do que pode ser computado e como se computa, o pensamento computacional corresponde a uma série de ferramentas mentais que refletem a dimensão do campo da Ciência da Computação, que dizem respeito a pensar basicamente na resolução de problemas em múltiplos níveis de maneiras recursiva e paralela (Wing, 2006).

Posteriormente, Wing (2008) definiu que pensamento computacional consiste em processos de pensamento envolvidos na criação e resolução de problemas que possam ser solucionados por um ser humano ou máquina. Para Wing (2008), pensar computacionalmente é, em outras palavras, pensar como um cientista da computação. Wing (2006, 2008, 2010) também argumenta que o pensamento computacional é uma atitude e uma habilidade para todos em qualquer lugar, não apenas para cientistas da computação, uma vez que os avanços das tecnologias têm mudado a maneira como as pessoas pensam e se relacionam com o mundo. A Ciência da Computação contribui significativamente para inúmeras áreas, por meio de estruturas computacionais que possibilitam a resolução de problemas complexos. Segundo Wing (2006, 2008, 2010), o pensamento computacional será uma habilidade indispensável para qualquer profissional do futuro, assim como atualmente a leitura e a escrita são competências necessárias para qualquer um, em qualquer lugar. Pensar computacionalmente consistirá em uma necessidade para estar incluso na sociedade, na escola e no mundo do trabalho - é o que conclui Blikstein:

Portanto, a habilidade de transformar teorias e hipóteses em modelos e programas de computador, executá-los, depurá-los, e utilizá-los para redesenhar processos produtivos, realizar pesquisas científicas ou mesmo otimizar rotinas pessoais, é uma das mais importantes habilidades para os cidadãos do século XXI. (BLIKSTEIN, 2008, não paginado).

Enquanto os seres humanos possuem inúmeras habilidades no processamento de alguns tipos de informações, como imagens e gráficos, os computadores são capazes de processar um número inestimável de dados e algoritmos de maneira infinitamente mais rápida do que a mente humana. Entretanto, não é necessária a existência de computadores para se pensar computacionalmente, mas, se o pensamento humano for aliado à alta capacidade de processamento das máquinas, a capacidade de processamento pode ser muito maior (WING, 2008).

Se o pensamento computacional será usado em todos os lugares, então ele vai tocar a todos, direta ou indiretamente. Isto levanta um desafio educacional. Se o pensamento computacional é adicionado ao repertório de habilidades de pensamento, então como e quando as pessoas devem aprender esse tipo de pensamento e como e quando devemos ensiná-la? (WING, 2008, tradução nossa) ¹.

Dentre as competências necessárias para o desenvolvimento do PC, Wing (2008) define que o processo de mais alto nível consiste na abstração. Nas palavras de Barnes e Kölling (2004), a abstração e a decomposição podem ser entendidas como compreender um problema complexo, extrair os dados fundamentais e dividi-lo em pequenos problemas de menor complexidade, até que problemas menores sejam suficientemente simples para se trabalhar individualmente, tratando os problemas em blocos independentes.

Em outras palavras, a capacidade de decompor um problema em partes menores possibilita que o mesmo seja compreendido mais facilmente e sistematizado em forma de código de programação. Assim, temos que a abstração é um conceito necessário para a resolução de problemas, pois, à medida que um problema se torna demasiadamente grande, torna-se mais difícil trabalhar com todas as suas características ao mesmo tempo.

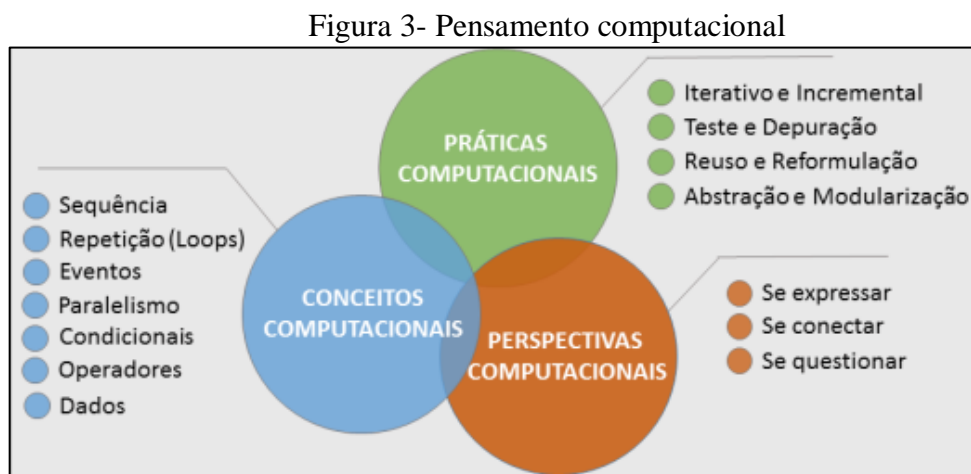
No campo da programação orientada a objetos, por exemplo, o conceito de abstração também se constitui com fundamental importância, uma vez que possibilita a implementação de classes e componentes com funções independentes.

Para nos ajudar a manter uma visão geral em programas complexos, tentamos identificar subcomponentes que podemos programar como entidades independentes. Então tentamos utilizar esses subcomponentes como se fossem partes simples sem nos preocupar com suas complexidades internas. (BARNES E KÖLLING, 2004, p. 50).

Brennan e Resnick (2012) afirmam que, embora o PC esteja em constante discussão, ainda há pouco consenso sobre quais as competências efetivas que são englobadas pelo conceito. Nesse sentido, os autores propõem que o pensamento computacional engloba três grandes dimensões: Em primeiro lugar, os conceitos de sequência, repetição, condição, eventos, paralelismo, operadores e dados. Em segundo, as práticas de programação que correspondem a pensamento incremental e interativo, testes e depuração, abstração, reutilização e modularização. E, por fim, apresentam as seguintes perspectivas: expressão (por meio da computação o programador pode expressar seus sentidos, sentimentos e a sua vontade de criar),

¹ If computational thinking will be used everywhere, then it will touch everyone directly or indirectly. this raises an educational challenge. If computational thinking is added to the repertoire of thinking abilities, then how and when should people learn this kind of thinking and how and when should we teach it?

conexão (criar com outros indivíduos e a partir de outros projetos) e questionamento (poder questionar a tecnologia por meio dela própria). O conceito proposto por esses autores pode ser entendido a partir do gráfico de Cavalcante, Costa e Araujo (2016), exposto a seguir.



Fonte: Cavalcante, Costa e Araújo (2016).

Nesse caminho, Blikstein apresenta uma definição mais sintética. Em suas palavras:

Pensamento computacional é saber usar o computador como um instrumento de aumento do poder cognitivo e operacional humano – em outras palavras, usar computadores, e redes de computadores, para aumentar nossa produtividade, inventividade e criatividade. (BLISKTEIN. 2008, não paginado).

No campo da educação, inicialmente os conceitos de pensamento computacional foram percebidos e introduzidos nos currículos de cursos superiores, tendo em vista a necessidade do conhecimento de computação e de linguagens de programação, em especial nas áreas de engenharia. Hoje, contudo, a introdução da Ciência da Computação tem acontecido mundialmente desde o ensino básico. O Reino Unido foi pioneiro na introdução do ensino de computação nas escolas, seguido pelos Estados Unidos e, mais recentemente, por outros países da Europa, embora Valente (2016) argumente que poucos países fizeram, de fato, a inclusão do PC nos currículos escolares.

Essa introdução segue duas perspectivas importantes: de um lado, a carência mundial de profissionais da área de Tecnologia da Informação (TI); de outro, a contribuição dos conceitos de computação para o pensamento lógico matemático e o desempenho escolar (BRACKMANN, 2016).

A informática na educação, introduzida em especial como uma ferramenta de trabalho para produção de projetos na escola, tomou uma dimensão que é normalmente passiva, ou seja,

os alunos costumam aprender atividades que já faziam antes, porém, utilizando o computador. É o caso das pesquisas no *Google*, que substituíram as enciclopédias físicas e das apresentações de *slides*, que substituíram os cartazes. Fagundes (1999), Valente (1999), Demo (2011) e outros têm criticado essas abordagens em que a tecnologia é incluída nas práticas pedagógicas sem um rompimento epistemológico. Na contramão dessa realidade, o pensamento computacional se opõe ao uso passivo da informática, deslocando o aluno de receptor para produtor de tecnologia. É o que explica Blikstein (2008, não paginado): “não dá para redesenhar uma linha de produção, ou decodificar o DNA, copiando e colando textos da internet”. O Pensamento Computacional (PC) traz uma dimensão inovadora no campo da aprendizagem e da produção de conhecimento, uma vez que o aluno passa a ser agente de seu processo de aprendizagem e desenvolvedor de novos produtos.

Em geral, o conceito de nativo digital (PRENSKY, 2001) explica que as crianças nascidas após a revolução digital possuem grandes habilidades no uso das ferramentas tecnológicas, contudo, essas habilidades são ainda utilizadas como meio de desenvolver excelentes usuários e não desenvolvedores. “Estamos ensinando nossos alunos que a tecnologia serve para recombinar informações já existentes, e não para criar conhecimento novo.” (BLIKSTEIN, 2008, não paginado). Assim, emerge uma questão fundamental que corresponde a possibilitar que no futuro uma grande parte da população possua as competências necessárias para o desenvolvimento de tecnologias que contribuam para o desenvolvimento social, já que apenas recombinar informações e organizá-las por meio de um computador não são suficientes para a criação de novas tecnologias.

Resnick (2012) critica a educação tradicional, em que as crianças aprendem apenas conceitos que não são significativos para elas. É o caso do ensino de variáveis a uma criança do sexto ano, por exemplo. Como ela sabe qual é sua efetiva função? Que relações ela pode fazer com seu cotidiano? O ensino de programação e o desenvolvimento do pensamento computacional podem possibilitar uma compreensão de significados na prática, que permitem à criança compreender estruturas que antes eram extremamente abstratas.

Nesse sentido, é comum que olhemos para a escola, os professores e o sistema educacional em geral para que possamos nos perguntar qual a melhor forma de desenvolver essas habilidades. Acreditamos que a introdução do PC nas práticas escolares demanda uma mudança da escola e dos professores, necessitando, nesse sentido, de um processo de formação continuada que crie significados entre a prática e a epistemologia do professor.

3.3 PENSAMENTO COMPUTACIONAL: TRABALHOS INVESTIGADOS

O Advento da tecnologia em rede, assim como transformou inúmeros segmentos da sociedade, trouxe significativas mudanças às pesquisas científicas. Atualmente, milhões de estudos são criados e publicados na rede mundial de computadores e estão ao alcance de todos.

Embora um mundo de informação permita ao pesquisador um deleite na rede do conhecimento, uma prática comum na pesquisa, como elencar tudo o que foi produzido referente a um determinado assunto, tem se configurado cada vez mais difícil.

Metodologias “Estado da Arte” têm sido desafiadoras e exigem recortes significativos para que toda a produção científica de uma área possa ser analisada. Longe desta tentativa e no intuito de contextualizar os estudos que convergem com esta dissertação, optamos por realizar uma revisão sistemática de literatura. Assim, elencamos os artigos publicados em dois eventos do Congresso Brasileiro de Informática na Educação (CBIE). São eles: Os *Workshops* do Congresso e o Simpósio Brasileiro de Informática na Educação (SBIE), nos anos de 2014, 2015, 2016 e 2017, com os seguintes descritores: Pensamento Computacional, Formação de Professores e Ensino de Programação. Foram encontrados 160 artigos relacionados com o objeto de estudo, que foram organizados nas seguintes categorias: 1 - Análise de Plataforma, 2 - Educação Básica, 3 - Formação Docente, 4 - Ensino Superior. 5 - Outra.

A partir das métricas estabelecidas por Wohlin et al. (2012), no que diz respeito à revisão sistemática de literatura, esta pesquisa foi definida em três etapas. Na primeira, foram definidos critérios de inclusão dos artigos, que são os seguintes: 1- Ter relação com o desenvolvimento do pensamento computacional. 2 - Apresentar estudo ou análise de ferramenta para o desenvolvimento do pensamento computacional e 3 - Apresentar relação entre pensamento computacional e formação de professores.

A segunda etapa consistiu na exclusão dos artigos que foram listados, mas que não possuíam relevância com o tema ou que se tratavam de revisões sistemáticas de literatura. Assim, foram elencados os seguintes critérios de exclusão: 1 - Estar relacionado ao ensino de algoritmos em nível superior em cursos de computação. 2 - Não possuir relação com o tema da investigação. 3 - Tratar-se de artigos de revisão sistemática de literatura.

O quadro a seguir mostra o total de artigos por categoria relacionados com os termos de busca.

Quadro 1 - Artigos Analisados

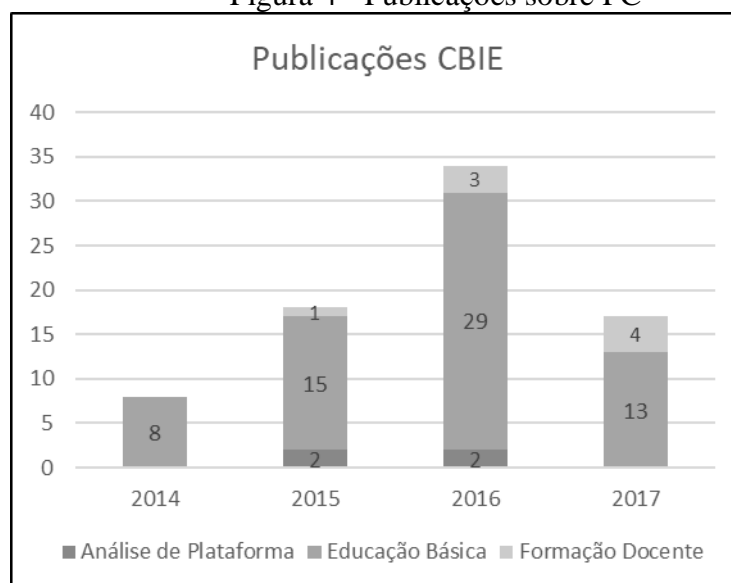
Base de Dados	Excluir	Incluir	Total Geral
CBIE	71	66	137
SBIE	12	11	18
Total Geral	83	77	160
%	51,88%	48,13%	100%

Fonte: Autor.

A partir do Quadro 1, podemos perceber que há uma quantidade significativa de artigos presentes em ambas as bases de dados e que possuem alguma relação com o tema pensamento computacional, o que pode indicar que o assunto tem ganhado maior relevância no cenário nacional.

De acordo com os critérios de seleção estabelecidos, selecionamos 77 (48,13%) artigos para a análise, conforme a figura abaixo.

Figura 4 - Publicações sobre PC



Fonte: Autor.

A partir do gráfico representado pela Figura 4, podemos perceber que há um crescente de publicações sobre pensamento computacional e que a maior parte dos estudos tem alguma relação com o PC na educação básica (84,42%). Além disso, foi possível observar que, a partir do ano de 2015, foi criado no CBIE um espaço especial para trabalhos sobre pensamento computacional, incluindo estudos que antes estariam em categorias diversas ou relacionadas a

metodologias no ensino superior. Nesse sentido, justifica-se a exclusão de mais de 50% dos artigos desta base.

Paradoxalmente, publicações no campo da formação de professores obtiveram um crescimento pouco significativo. Comprovando uma carência de pesquisas nessa área, conforme discutido por Farias, Andrade e Alencar (2015), Barcelos et al. (2015), Barcelos, Bortoletto, Andriolli (2016), Souza, Rodrigues e Andrade (2016). Para esses autores, embora muito se tenha falado sobre pensamento computacional nos últimos dez anos, ainda há poucos estudos que privilegiem as interlocuções entre PC e formação inicial e continuada de professores.

Os trabalhos mencionados trazem diferentes perspectivas de formação de professores para a atuação com o PC. Farias, Andrade e Alencar (2015) buscam um olhar sobre o pensamento computacional e a formação docente, focalizando no profissional licenciado em computação. Na visão desses autores, a inserção do profissional da licenciatura em computação é fundamental para a implementação do pensamento computacional no ensino básico.

Barcelos, Bortoletto e Andriolli (2016) entendem que o pensamento computacional como uma competência escolar pode ser desenvolvido pelos professores de todas as disciplinas e apontam para a necessidade iminente de formação de professores. Os autores trazem o relato de um curso de formação de professores de matemática em modalidade *on-line*, em que foram desenvolvidas estratégias para a formação docente por meio do desenvolvimento de jogos digitais abordando e discutindo os conceitos de matemática.

Souza, Rodrigues e Andrade (2016) buscam evidenciar a necessidade de introdução do pensamento computacional para o uso de recursos de robótica, uma vez que identificaram que as dificuldades de aprendizado de alunos e professores em oficinas de robótica estão, dentre outros aspectos, relacionadas à ausência de conhecimentos básicos de pensamento computacional em alunos e professores, vindo, nesse sentido, ao encontro da necessidade de formação docente. Já Barcelos et al. (2015) analisam a questão da formação de professores relacionada entre o pensamento computacional e as interlocuções com o ensino de matemática, concluindo que há poucos estudos neste campo. Nesse caminho, uma publicação mais recente de Cavalcante et. al (2017) buscou analisar que conceitos de pensamento computacional estão claros para os professores do Ensino Médio. A partir de um estudo de caso exploratório, os autores investigaram o conhecimento e a aplicação das nove competências relacionadas ao PC, de acordo com a CSTA. Desse modo, foi identificado que os professores do estudo não possuíam conhecimento acerca do uso e da aplicação de conceitos de Pensamento Computacional na prática pedagógica. Nesse sentido, fica evidente que, ao menos para os

professores analisados, os conceitos de pensamento computacional não são explorados e utilizados de acordo com o potencial que possuem. Em um caminho semelhante, Leite et al (2017) identificam que, apesar de existirem políticas públicas que fomentem o letramento digital e a formação de professores para o uso de tecnologias, ainda não há iniciativas, pelo menos não de cunho governamental, de formação que fomentem o estudo e desenvolvimento do pensamento computacional nos processos de formação de professores.

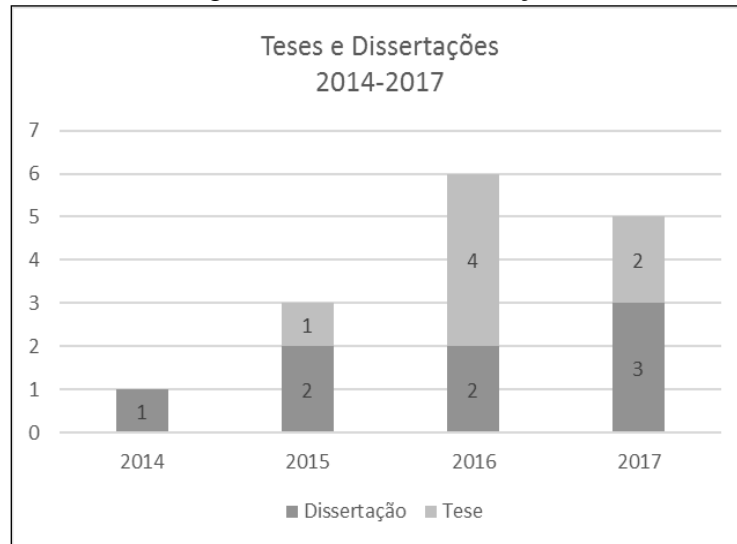
Além dessas questões mencionadas, podemos perceber uma carência de trabalhos que tragam análise ou discussão de plataformas para o desenvolvimento do pensamento computacional – representam apenas 5%, como os de Gomes e Alencar (2015), Gomes et al. (2015), Fernandez e Silveira (2016) e Cavalcante, Costa e Araujo (2016) - este último apresenta a análise da plataforma *Code.Org*, um ambiente que tem crescido, mas que ainda possui pouca menção no mundo acadêmico. Uma revisão sistemática de literatura de Bombasar et. Al (2015) mostra as plataformas utilizadas em nível internacional para o desenvolvimento do pensamento computacional, mas não faz menção ao *Code.Org*, enquanto outra revisão realizada por Fernandez e Silveira (2016) aponta que, dentre as plataformas *on-line* para o desenvolvimento do PC avaliada pelos autores, nenhuma utiliza a *Code.org*, embora a plataforma tenha sido apenas mencionada em dois dos estudos avaliados.

No intuito de corroborar com esta investigação, uma segunda etapa desta revisão consistiu em buscar dissertações e teses publicadas nos últimos quatro anos (2014, 2015, 2016 e 2017) com o tema Pensamento Computacional. Inicialmente, foi realizada uma pesquisa no banco de teses e dissertações da Capes, tendo como diretriz de busca o termo “Pensamento Computacional”. Contudo, os resultados da pesquisa não foram significativos, uma vez que o algoritmo de busca retornou milhares de produções que não se relacionavam ao termo. Desse modo, optamos por uma busca no *Google Acadêmico*, seguindo as mesmas métricas anteriores, porém adicionando-se os termos “Tese” ou “Dissertação”². Desse modo, foram encontrados 196 resultados, analisados individualmente e, deste montante, foram extraídos 14 trabalhos. Esse número relativamente baixo se justifica, pois, a pesquisa do motor de busca não permite a restrição pelo tipo de produção, acarretando em uma série de resultados que não se enquadravam nas categorias “Tese” ou “Dissertação”.

² A pesquisa no Google Acadêmico foi realizada com os termos "tese" & "pensamento computacional" & "programa de pós-graduação" e em seguida com os termos "Dissertação" & "pensamento computacional" & "programa de pós-graduação", ambas realizadas no dia 20/02/2018.

Após a mineração desses resultados, identificamos 7 teses e 8 dissertações que apresentavam o termo “Pensamento Computacional”. O gráfico abaixo demonstra a distribuição dessas produções em função do tempo.

Figura 5 - Teses e dissertações



Fonte: Autor.

Como se pode observar, em virtude de o termo ser ainda recente no cenário brasileiro, existem poucas dissertações e teses que tratam do assunto. Nesse contexto, Brackmann (2017) investigou questões acerca da computação desplugada como uma estratégia do desenvolvimento do pensamento computacional no ensino básico. Apesar do tema de pesquisa não estar diretamente vinculado à formação de professores, o autor se preocupa em problematizar essa questão, afirmando que pouco tem se pensando em formação de professores no campo da prática pedagógica com o pensamento computacional. Além disso, aponta a computação sem o uso do computador como alternativa que possibilita o desenvolvimento do pensamento computacional em escolas que não dispõem da estrutura necessária para o uso de *softwares* para esta finalidade.

Em proposta semelhante, Carbajal (2016) buscou investigar um caminho para o desenvolvimento do pensamento computacional, por meio da construção de um ambiente de programação tangível. Neste estudo, o autor demonstra que é possível desenvolver conceitos de PC pelo uso de artefatos concretos e propõe uma ferramenta de baixo custo para ser usada como recurso pedagógico para o desenvolvimento do PC.

Bouchinha (2017) buscou evidenciar a correlação entre o raciocínio lógico e o desenvolvimento do pensamento computacional por meio de um estudo exploratório em

oficinas oferecidas em duas escolas de ensino básico. Como conclusão, identificou que existe incremento do pensamento lógico através do desenvolvimento do pensamento computacional.

Já os trabalhos de Stella (2016) Sperb (2014), Bozolan (2016) Bressan (2016) Oliveira Júnior (2017) se dedicam a investigar o desenvolvimento do pensamento computacional por meio de plataformas como o *Scratch* e o *Etoys*. Esses tipos de estudo são comuns na literatura, embora o termo Pensamento Computacional, por ser relativamente novo, não seja geralmente encontrado em trabalhos anteriores como, por exemplo, em Alves (2013), Dullius (2008) e outros.

Já Amaral (2015) analisa e discute as questões do aprendizado de programação por uma proposta que rompe o ensino de programação a partir da introdução de linguagens estruturadas para o paradigma de blocos. Nesse estudo, o autor demonstra que a compreensão dos alunos é muito maior nesse paradigma do que se comparada às iniciativas tradicionais para o ensino de programação e algoritmos.

Dargains (2015) analisa a robótica pedagógica como recurso para o desenvolvimento do pensamento computacional, buscando sustentar seus estudos a partir do construcionismo e da Taxonomia de Bloom. Em seu estudo, conclui que a robótica pedagógica é eficaz como recurso para o desenvolvimento do pensamento computacional em detrimento das metodologias e processos tradicionais para o ensino de algoritmos e programação. No mesmo caminho da robótica pedagógica, Silva (2017) apresenta um estudo sistemático sobre os trabalhos desenvolvidos nos últimos anos acerca deste tema, assim a autora conclui que o desenvolvimento do pensamento computacional pode ocorrer através de atividades com robôs em ambientes de livre criação e inspirados nas perspectivas da aprendizagem construcionista.

Couto (2017) apresenta um estudo sistemático, buscando as produções acadêmicas acerca do pensamento computacional com o recorte temporal de 1965-2016 e, a partir desse estudo, busca aproximar o conceito de PC com os constructos teóricos da educação libertadora proposta por Paulo Freire.

Os estudos aqui analisados nos mostram que as produções acadêmicas (teses e dissertações) envolvendo o tema pensamento computacional são crescentes e o tema pode ser explorado por meio do desenrolar de outros temas subjacentes ao pensamento computacional. É o caso da formação de professores, um tema que ainda tem muito a ser explorado. Com o objetivo de investigar a produção acadêmica neste campo, utilizamos o *Google Acadêmico* para pesquisar por teses e dissertações com os descritores “Pensamento Computacional e Formação

Docente” e “Pensamento Computacional e Formação de Professores”³, porém o mecanismo de busca encontrou poucos resultados. Essa busca nos mostra que, como já mencionado, há uma produção científica que busca discutir a formação de professores ainda muito restrita a eventos e periódicos e que talvez, pelo ineditismo do assunto, ainda são encontrados poucos trabalhos densos sobre o mesmo.

3.4 PC: PLATAFORMAS VIRTUAIS E FORMAÇÃO DE PROFESSORES

A emergência de discussões no campo do pensamento computacional e do ensino de Ciência da Computação na educação básica nos últimos anos trouxe a necessidade de se pensar nos processos de formação de professores para essas demandas.

Recentemente, a Base Nacional Comum Curricular (BNCC) trouxe, em sua última versão, diversas menções acerca do conceito de Pensamento Computacional como sendo fundamental para o desenvolvimento de algumas habilidades na área de Matemática e suas tecnologias, tanto no que tange o Ensino Fundamental como o Ensino Médio.

A área de Matemática, no Ensino Fundamental, centra-se no desenvolvimento da compreensão de conceitos e procedimentos em seus diferentes campos, visando à resolução de situações-problema. No Ensino Médio, na área de Matemática e suas Tecnologias, os estudantes devem utilizar conceitos, procedimentos e estratégias não apenas para resolver problemas, mas também para formulá-los, descrever dados, selecionar modelos matemáticos e desenvolver o pensamento computacional, por meio da utilização de diferentes recursos da área (BRASIL, 2017, p. 470).

O desenvolvimento dessas habilidades está intrinsecamente relacionado a algumas formas de organização da aprendizagem matemática, com base na análise de situações da vida cotidiana, de outras áreas do conhecimento e da própria Matemática. Os processos matemáticos de resolução de problemas, de investigação, de desenvolvimento de projetos e da modelagem podem ser citados como formas privilegiadas da atividade matemática, motivo pelo qual são, ao mesmo tempo, objeto e estratégia para a aprendizagem ao longo de todo o Ensino Fundamental. Esses processos de aprendizagem são potencialmente ricos para o desenvolvimento de competências fundamentais para o letramento matemático (raciocínio, representação, comunicação e argumentação) e para o desenvolvimento do pensamento computacional (BRASIL, 2017, p. 264).

Esses fragmentos da BNCC apontam para o crescimento e para o reconhecimento da importância do conceito ligado às competências da matemática durante todo o ensino básico. É interessante salientar, inclusive, que BRACKMANN (2016), em um estudo recente, mostrou que até 2016 ainda não havia de forma declarada os conceitos de Pensamento Computacional

³ A pesquisa no Google Acadêmico foi realizada com os termos "pensamento computacional" & "formação de professores" & "tese" & "programa de pós-graduação" no dia 20/02/2018.

nas bases anteriores, embora se pudesse inferir a partir deles uma correlação com o conceito de PC. Essa evolução no que diz respeito à compreensão e importância do conceito para a área de Matemática e Suas Tecnologias corroboram ainda mais para uma necessidade de se pensar a formação do professor nesse contexto.

Durante os primeiros passos da filosofia LOGO, muitas iniciativas de formação proporcionaram aos professores o contato com a linguagem LOGO e sua filosofia, e esse movimento que se insere na problemática da informática na educação formou inúmeros professores no mundo todo (VALENTE, 2016).

Atualmente, a necessidade de novas competências para o professor do século XXI ligadas ao PC nos conduz a pensar em iniciativas e em conceitos a serem desenvolvidos no campo de formação de professores. Para Tardif, a formação de professores perpassa saberes que são específicos e inerentes à prática pedagógica, “[...] isto é, dos saber-fazer, das competências e das habilidades que servem de base ao trabalho dos professores no ambiente escolar” (TARDIF, 2011, 227).

De acordo com Valente (2016), o PC tem sido inserido de três maneiras distintas. A primeira diz respeito à inclusão de uma nova disciplina que contemple os conceitos de Ciência da Computação e Pensamento Computacional, a segunda corresponde à inclusão de oficinas que ofereçam PC como atividade complementar e a última corresponde à inclusão do PC como um recurso transdisciplinar.

É preciso, pois, esclarecer que cada uma dessas maneiras traz significados diferentes. A inclusão do PC como uma disciplina ou como uma atividade complementar não exige modificações paradigmáticas efetivas. Papert (1988) critica esse movimento que havia ocorrido nas escolas desde os primeiros passos da inclusão de linguagens de programação na escola, argumentando que esse tipo de “solução” não promove modificações significativas na filosofia da educação. Valente (2016) argumenta que muitas escolas ao redor do mundo resolveram o problema adicionando novas disciplinas ou oficinas e que, efetivamente, não há verdadeiras modificações na estrutura curricular das escolas que implementaram ensino de Ciência da Computação.

Modificações nas estruturas curriculares e nas formas de ensino estão intimamente ligadas a políticas públicas. Alguns países como a Inglaterra, por exemplo, alteraram seus currículos a fim de contemplar as novas competências do século XXI oriundas da Ciência da Computação. Contudo, de acordo com Valente (2016), a maioria dos países ainda não criaram políticas públicas que possibilitassem verdadeiras alterações nesse sentido. A exemplo disso, temos os Estados Unidos que, embora muitas escolas estejam adicionando o PC computacional de

alguma maneira às suas grades curriculares ou extracurriculares, essas iniciativas acontecem principalmente por instituições não vinculadas ao governo, como é o caso da Computer Science Teacher Association (CSTA). A CSTA é uma instituição americana de nível global com mais de 25.000 membros em 145 países. A associação “oferece oportunidades para que os professores do ensino fundamental e médio e seus alunos compreendam melhor a ciência da computação e preparem-se com mais êxito para ensinar e aprender”. (ABOUT CSTA, 2018, não paginado). A organização definiu os pilares do pensamento computacional que contribuem para o ensino de Ciência da Computação. São eles: Decomposição, Reconhecimento de Padrões e Algoritmos. Brackmann (2016) buscou esclarecer esses conceitos:

O Pensamento Computacional envolve identificar um problema complexo e quebrá-lo em pedaços menores e mais fáceis de gerenciar (DECOMPOSIÇÃO). Cada um desses problemas menores pode ser analisado individualmente com maior profundidade, identificando problemas parecidos que já foram solucionados anteriormente (RECONHECIMENTO DE PADRÕES), focando apenas nos detalhes que são importantes, enquanto informações irrelevantes são ignoradas (ABSTRAÇÃO). Por último, passos ou regras simples podem ser criados para resolver cada um dos subproblemas encontrados (ALGORITMOS). Seguindo os passos ou regras utilizadas para criar um código, é possível também ser compreendido por sistemas computacionais e, conseqüentemente, utilizado na resolução de problemas complexos eficientemente, independentemente da carreira profissional que o estudante deseja seguir (BRACKMANN, 2016, p. 33).

Essas definições permitem que pessoas no mundo inteiro busquem se aproximar das perspectivas da prática pedagógica com o Pensamento Computacional e contribuem para outras instituições que fomentam a disseminação dessas competências. A exemplo disso, temos a *Code.org* que leva em consideração essas definições para propor suas atividades e planos de aula. Conforme já mencionado, a *Code.org* também possui um importante papel na inclusão do PC e na disseminação de uma cultura relativa à Ciência da Computação nos Estados Unidos, em especial por dispor de grandes financiamentos e apoio de pessoas famosas no mundo todo.

A *Code.Org* tem um importante papel, uma vez que se dedica, não apenas a levar o PC para todos ou de dispor ambientes para o aprendizado de programação a partir de interfaces intuitivas. Ela vai muito além ao preocupar-se com a formação de professores e propor formações e materiais que dão subsídios às práticas pedagógicas.

Essa demanda por formação também ocorre na realidade brasileira, pois o Brasil não dispõe de um programa de políticas que contemplem a inclusão do PC nas escolas e a formação de professores para tal. Nesse sentido, essas demandas são supridas à medida que organizações

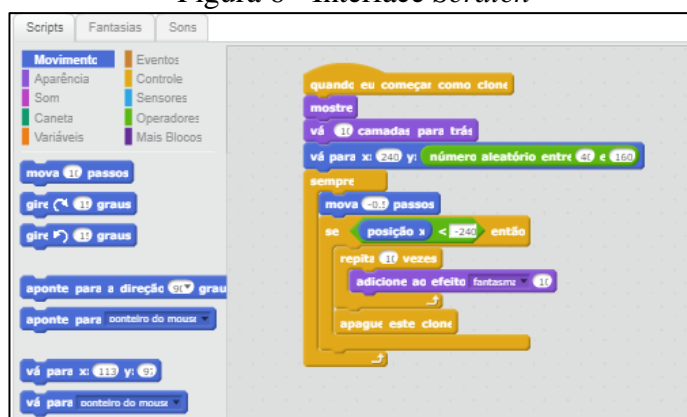
como o PROGRAMAÊ!, por exemplo, movimentam-se para mostrar à sociedade a importância do PC e a necessidade de formações.

Entretanto, muitas vezes essas formações ocorrem de maneira puramente técnica, sem interseções com uma epistemologia convergente ao modelo com o qual essas tecnologias foram concebidas. Nesse sentido, é importante salientar as iniciativas que visam disponibilizar recursos e formação de professores para a implementação do PC no currículo ou como atividades extracurriculares. Há muita literatura analisando e comparando os diversos ambientes para o desenvolvimento de pensamento computacional. Trabalhos como os de Bombasar et. al (2015) apontam plataformas como o *Scratch*, *Code.org*, *Kodu* e tantas outras disponíveis para o aprendizado e desenvolvimento do pensamento computacional. Contudo, muitas dessas plataformas possuem também ambientes que subsidiam a prática pedagógica com o pensamento computacional, mas essas questões não são discutidas na literatura. Para delinear este estudo, optamos por descrever brevemente esses ambientes, bem como caracterizar essas ferramentas no intuito de poder tecer considerações acerca das possibilidades do desenvolvimento do PC e do ensino e da aprendizagem dessas habilidades nos processos de formação docente.

3.4.1 Scratch

O *Scratch* é uma linguagem de programação visual também chamada de VPL (*Visual Programming Language*), criada pelo *Massachusetts Institute of Technology* (MIT) no ano de 2007. Esta ferramenta é fruto da união de princípios da filosofia LOGO com as características dos brinquedos LEGO. O resultado foi um ambiente de programação intuitivo em que qualquer pessoa pode aprender a programar através da organização de blocos.

Figura 6 - Interface *Scratch*



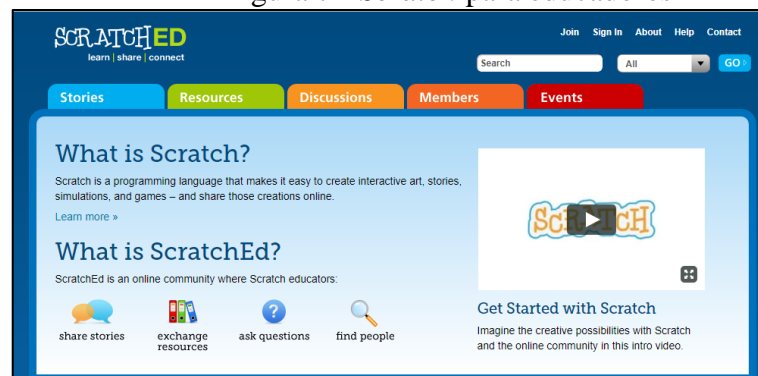
Fonte: <https://scratch.mit.edu/projects/155107004>.

No *Scratch*, cada bloco tem funções específicas e o programador pode utilizá-los para criar *games*, animações e histórias animadas. Essa plataforma, por ser inspirada na filosofia LOGO, permite que o usuário crie projetos totalmente do zero, explorando funcionalidades sem que haja nenhum tipo de conhecimento prévio.

Além do ambiente instigador, a ferramenta conta com uma vasta comunidade em que os *scratcher*, (criadores de projetos *Scratch*) podem compartilhar seus projetos com a comunidade. O *Scratch* é uma ferramenta bastante utilizada e mencionada no mundo acadêmico, diversos trabalhos trazem estudos que nos apresentam potencialidades do uso do *software* como ferramenta para o desenvolvimento do pensamento computacional.

O projeto também conta com um local específico para as interações e trocas, em especial de educadores que implementam estratégias para o ensino e a aprendizagem por meio do uso do *Scratch*. No *Scratch* para Educadores, disponível no *link* <https://scratch.mit.edu/educators>, também estão disponíveis recursos para o encontro presencial de professores, tutoriais e dicas e o Guia Curricular da Computação Criativa, uma iniciativa que traz uma metodologia para o ensino e a aprendizagem da programação por meio de conceitos da computação criativa.

Figura 7 - *Scratch* para educadores



Fonte: <https://scratch.mit.edu/educators>.

A página conta com ambiente disponível para o compartilhamento de experiências, disponibiliza recursos para os professores, discussões e eventos disponíveis para encontros e formação. Todos os recursos do *Scratch*, assim como a comunidade e os eventos disponíveis são totalmente gratuitos e podem ser utilizados sem a necessidade de qualquer tipo de licença para o aprendizado, desenvolvimento e implementação do ensino e da aprendizagem de programação.

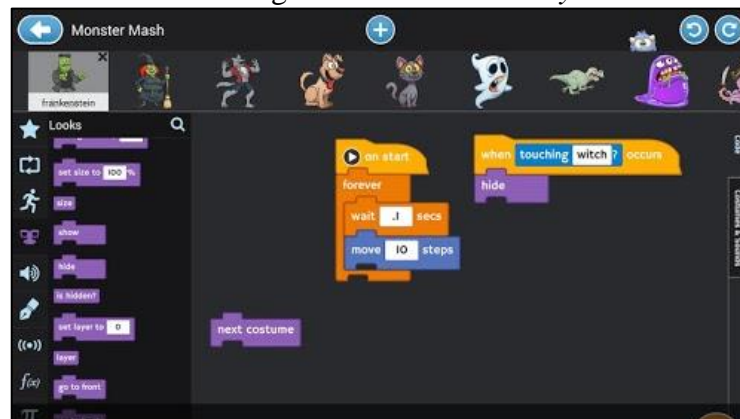
3.4.2 Tynker

A *Tynker* é uma empresa do vale do Silício, fundada em 2012, que tem por finalidade oferecer uma série de recursos para o desenvolvimento de competências ligadas à Ciência da Computação e ao PC. Assim como as outras iniciativas disponíveis no mercado, a empresa acredita que a capacidade de programar é uma competência necessária para todos. “Acreditamos que a capacidade de codificar permite que as crianças concretizem suas ideias.” (TYNKER, 2018).

A *Tynker* conta com diversas empresas investidoras, também residentes no Vale do Silício, que acreditam na disseminação do PC. Diferentemente de outras iniciativas, como o *Scratch* ou a *Code.Org*, a *Tynker* disponibiliza a maior parte de seus recursos, que vão desde a programação em blocos até games e recursos avançados para o uso das ferramentas em sala de aula, por meio de planos em que as escolas precisam aderir para ter acesso ao material. Para o acesso livre, a *Tyker* oferece alguns tutoriais e o ambiente de programação com algumas limitações.

A programação disponível na plataforma se assemelha bastante à disponível na interface do *Scratch*, como se pode visualizar na imagem abaixo:

Figura 8 - Interface do *Tynker*



Fonte: <https://play.google.com/store/apps/details?id=com.tynker.TynkerPremium>.

O *Tynker*, por ser uma plataforma voltada e comercializada para escolas, conta com inúmeros materiais disponíveis para o desenvolvimento da Ciência da Computação nas escolas a partir de uma proposta de grade curricular, planos de aula e atividades voltadas para o professor e para a implementação do ensino e da aprendizagem dos conceitos de PC na escola.

3.4.3 Kodu

O *Kodu* é uma ferramenta *Microsoft* criada em 2009 para o aprendizado e desenvolvimento de *games* 3D. A ferramenta consiste, basicamente, em um *game* para criar *games*. A partir da programação visual, qualquer pessoa, criança ou adulto pode desenvolver os seus próprios jogos por meio da criação de mundos 3D e da utilização de personagens disponíveis na plataforma. O *Kodu*, diferentemente das outras plataformas descritas nesta dissertação, precisa ser instalado no computador e está disponível para computadores pessoais e para o console *XBOX*. O *Kodu*, apesar de utilizar programação visual, não utiliza blocos, mas sim elementos visuais que possibilitam ao usuário programar ações para os personagens da plataforma.

Figura 9 - Programação *Kodu*



Fonte: <https://www.microsoft.com/en-us/research/project/kodu/>.

No site oficial da plataforma kodugamelab.com, estão disponíveis um fórum geral para usuários e, no link “resources”, uma série de atividades e materiais disponíveis para professores que desejam utilizar a ferramenta como recurso para o desenvolvimento do PC e da aprendizagem de programação.

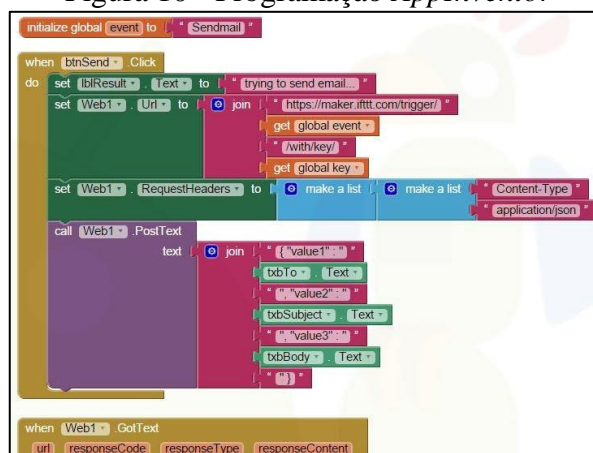
3.4.4 AppInventor

O *AppInventor* é uma plataforma de programação em blocos criada pela *Google* no ano de 2010 e, atualmente, mantida e atualizada pelo MIT. Essa plataforma tem como objetivo introduzir o aprendizado de programação por meio do desenvolvimento de aplicativos para o sistema operacional *Android*.

Baseado em um paradigma semelhante ao *Scratch*, o *AppInventor* também é um VPL que permite ao usuário criar aplicativos a partir da programação em blocos e, além disso,

familiarizar-se com uma linguagem de programação nativa, o *Java Script*. Essa plataforma também conta com um ambiente semelhante ao de um ambiente de programação profissional, em que o usuário irá se familiarizar com a estrutura de desenvolvimento de aplicativos para o Sistema Operacional *Android*.

Figura 10 - Programação *AppInventor*



Fonte: <https://www.hackster.io/taifun/trigger-ifttt-to-send-an-email-using-app-inventor-9df505>.

Assim como as outras plataformas, o *APP Inventor* conta com uma área para educadores, com recursos para o ensino e a aprendizagem da ferramenta, área de discussão e links importantes e tutoriais em vídeo, em que o professor poderá fazer uso para a prática em sala de aula.

3.4.5 Code.org

A *Code.org* é uma organização não governamental e sem fins lucrativos, que tem por objetivo divulgar e ensinar computação para pessoas de todas as idades, com foco específico em crianças que não possuem acesso a essas tecnologias. A plataforma *on-line* conta com um significativo acervo de tutoriais, planos de aula, e recursos que possibilitam aprender, ensinar, refletir e compartilhar informações entre os participantes. Qualquer usuário pode ter acesso e utilizar o *site* para aprender programação, entretanto, o maior foco da plataforma está nos professores e profissionais da educação. A escolha dessa plataforma se deu devido à forte presença de recursos para professores. Em virtude de a *Code.org* ser o cenário de estudo desta dissertação, detalharemos com mais precisão nesta seção os elementos do ambiente.

A instituição Americana foi fundada em 2013 e possui parceria com grandes empresas da tecnologia da informação, como a *Microsoft*, a *Apple* e o *Facebook*. De abrangência mundial,

a *Code.org* tem levado a cultura da computação para o mundo todo por meio de incentivos de grandes empresas e ícones no mundo da tecnologia. Disponível no endereço <http://www.code.org>, a página conta com depoimentos de grandes nomes da tecnologia, como Bill Gates (fundador da *Microsoft*) e Mark Zuckerberg (fundador do *Facebook*), que apresentam suas histórias e estimulam o uso e o desenvolvimento do aprendizado de computação para todos.

3.4.6 Programaê!

A PROGRAMAÊ! é uma plataforma brasileira que tem por finalidade levar a Ciência da Computação e o pensamento computacional para todos no Brasil. Disponível no endereço <http://programae.org.br/>, a plataforma é um grande repositório que, em parceria com outras plataformas, oferece uma compilação de atividades e recursos para quem quer aprender ou ensinar. O ambiente do PROGRAMAÊ! é todo em português, inclusive as áreas disponíveis para professores, o que facilita para os profissionais brasileiros da educação que desejam levar esses conhecimentos para a sala de aula. Os planos de aula de outras plataformas como a *Code.org* estão presentes no PROGRAMAÊ!, traduzidos e organizados de maneira mais simples, com vídeos e outros recursos que podem ser utilizados pelo professor. Embora a plataforma seja um excelente recurso para professores, não dispõe de fórum e ambiente que possibilite interações e trocas entre os professores que estão implementando os planos disponíveis, como ocorre no *Code.org*, por exemplo.

4 MÉTODO

“A consciência da complexidade nos faz compreender que não poderemos escapar jamais da incerteza e que jamais poderemos ter um saber total: a totalidade é a não verdade”.

Edgar Morin

Este capítulo apresenta o percurso metodológico desta dissertação, assim como o contexto de pesquisa, as fontes de análise e as escolhas realizadas para a construção dos dados.

4.1 DELINEAMENTO METODOLÓGICO

Desde os tempos passados, a natureza humana esteve ligada ao conhecer. Seja pela invenção da escrita, da impressa, das máquinas a vapor ou do computador moderno, o homem concretizou suas aspirações de produzir conhecimento. É desta capacidade de se pensar que emergem as questões de pesquisa.

A escrita é, talvez, o principal meio que a humanidade encontrou, desde os tempos pretéritos, para perpetuar suas ideias através dos tempos. Assim, os registros humanos, passíveis de interpretação, constituem-se como elementos a serem desvendados, sendo do pesquisador, não apenas a responsabilidade de fazer emergir desses registros o conhecimento, mas trazê-los à luz do entendimento.

Nesse caminho, esta dissertação apresenta o seguinte problema de pesquisa: **Como se evidenciam as concepções de ensino e aprendizagem presentes em uma plataforma *on-line* para o desenvolvimento do pensamento computacional?** Para responder a esta questão de investigação, elegemos como método uma pesquisa qualitativa, de caráter exploratório, tendo seu delineamento metodológico como estudo de caso, em que a unidade de análise consiste na proposta de formação de professores contida na plataforma *Code.Org* para o desenvolvimento do pensamento computacional.

No entendimento de Moraes (2003), a pesquisa qualitativa não pretende validar hipóteses, mas trazer subsídios para o entendimento de fenômenos, por meio de uma análise rigorosa e criteriosa. Assim também são os estudos de caso que, na visão de Yin (2010), são:

[...] uma investigação empírica que investiga um fenômeno contemporâneo em profundidade e em seu contexto de vida real, especialmente quando os limites entre o fenômeno e o contexto não são claramente evidentes. (YIN, 2010, p.39).

O autor ainda explica que o estudo de caso é utilizado para o entendimento de fenômenos da vida real de maneira profunda e que engloba uma série de outras relações contextuais e a utilização de múltiplas fontes de evidência. Além disso, questões “como” e “por que” são mais explanatórias, favorecendo o uso de estudos de caso.

Nesse sentido, entendemos que o estudo de caso caracteriza-se como uma metodologia profícua para esta dissertação, uma vez que o problema de pesquisa se constitui por uma questão “Como” e, além disso, trata-se de um caso contemporâneo e real, tendo em si inter-relações.

Yin (2010) ainda explica que o estudo de caso é uma metodologia que implica um olhar profundo de um ou mais fenômenos e que o pesquisador tem a responsabilidade de olhar para esses movimentos à luz de um referencial teórico previamente construído.

4.2 CONTEXTO E ORGANIZAÇÃO DOS DADOS

Entendemos que a pluralidade do ambiente fornece inúmeras fontes de informações passíveis de análise. A escolha do contexto da *Code.Org* se justifica, em especial, pelo fato de ser um projeto de nível mundial, que possui um foco em recursos destinados à formação de professores. Além disso, conforme já discutido, a *Code.org* ainda possui poucos trabalhos que se dedicam a discutir esta plataforma e suas possibilidades na literatura no Brasil. Diante das inúmeras possibilidades que a plataforma nos apresenta, elencamos três fontes de evidências. São elas as seguintes:

- a) O “*Teacher Forums*” – apresenta interações, relatos, dúvidas e anseios de professores que estão implementando as estratégias do ensino de programação em escolas e oficinas de codificação, sendo o fórum um ambiente vivo, que fornece informações do ponto de vista dos próprios professores, suas dificuldades e motivações para o ensino de programação.
- b) As Atividades – a plataforma *Code.org* traz uma grande variedade de tutoriais que são a principal forma de interação do aluno com a plataforma. Esses tutoriais buscam conduzir o aluno, desde os primeiros passos da programação e da lógica até estágios mais elaborados do pensamento computacional. Além disso, esses elementos permitem também ao professor o aprendizado e a familiarização com a plataforma e conceitos do pensamento computacional.
- c) Os Planos de Aula – acompanham um planejamento prévio disponível na plataforma *Code.Org* e dão subsídio ao professor para o ensino do pensamento computacional por meio do uso dos tutoriais (atividades *on-line*) e de atividades que vão além das atividades disponíveis em meio eletrônico.

Essas três fontes de evidência nos permitem inferir, não apenas sobre a estrutura de formação proposta pela *Code.Org*, mas também sobre os professores envolvidos no uso da plataforma, os desafios a serem vencidos e os possíveis promotores do processo de aprendizagem. A análise dessas fontes foi realizada de forma integrada, de modo a possibilitar a emergência das categorias de análise e permitir a construção de um metatexto que busque responder ao problema de pesquisa e aos objetivos propostos por esta dissertação.

4.2.1 Fórum

Para este estudo, escolhemos os tópicos “*Teaching Computer Science – Your Plans*”, em que os professores relatam principalmente os objetivos e as práticas pedagógicas realizadas ou que serão feitas no futuro e o tópico “*What strategies will you use when facilitating computer science learning in your classroom*”, que traz, basicamente, as estratégias que os professores pensam em utilizar para a implementação do ensino de Ciência da Computação e Pensamento computacional no ensino regular. Ambos pertencentes à subcategoria “General Teaching and Learning (All Grades)” e à categoria “*Code.Org Teachers Professional Learning Community*”.

Após a escolha dos tópicos, as postagens foram copiadas para um arquivo de texto e, a partir desse material, utilizamos um algoritmo em linguagem VBA desenvolvido pelo autor para organizar os dados. O algoritmo utilizado organizou as postagens em pastas de trabalho correspondentes aos tópicos analisados. Em cada pasta de trabalho, as informações foram dispostas por usuário, data e conteúdo das postagens. O algoritmo considerou como critério de seleção postagens com mais de 100 caracteres, pois entendemos que as postagens menores não seriam relevantes para a análise textual discursiva.

Para fins de organização e clareza dos dados apresentados, optamos por apresentar os comentários analisados a partir da notação de citação direta, seguido de um identificador de cada tópico e postagem. Assim, apresentaremos a origem do tópico a partir da notação T, seguida de um número que representa o primeiro ou o segundo tópico analisado. O tópico “*Teaching Computer Science – Your Plans*” será apresentado por meio da notação T1 e o tópico “*What strategies will you use when facilitating computer science learning in your classroom*”, por meio da notação T2. Em seguida, definiremos por P seguido de um número a identificação correspondente à ordem temporal da postagem. Dessa forma, a postagem 1 será indicada por P1, a postagem 2 por P2 e assim sucessivamente. Ou seja, cada postagem será identificada por meio da notação Tn-Pn.

No momento desta pesquisa, o fórum apresentava nove categorias, sendo elas: *Code.Org Teachers Professional Learning Community*, *Computer Science Principles (CSP)*, *Computer Science Discoveries (CSD)*, *Marketing and Recruitment*, *Computer Science Fundamentals (CSF) (K-5)*, *Exploring Computer Science*, *CS in Algebra*, *CS in Science e Uncategorized*. Em seguida, cada categoria apresentava inúmeras subcategorias e cada subcategoria, diversos tópicos relacionados à categoria acima.

4.2.2 Atividades

As atividades são tutoriais *on-line* que compõem a principal parte da *Code.Org*. A partir delas, usuários do mundo todo têm acesso aos primeiros passos de codificação, por meio de programação em blocos. Este conjunto de atividades (tutoriais) e aulas é denominado “*Code Studio*”. No momento desta pesquisa, por meio do acesso ao menu principal “Alunos”, estavam disponíveis cinco cursos completos, um curso de atividades *off-line*, mais de dez tutoriais “Hora do Código” e diversos laboratórios, como o “Game Lab”, voltados à criação de aplicações não direcionadas.

Para este estudo, por razões de viabilidade de tempo, optamos por realizar a análise completa do Curso 2, voltado a crianças já alfabetizadas. A justificativa de escolha deste curso se dá por ser o primeiro curso introdutório à Ciência da Computação disponível na plataforma voltada a crianças alfabetizadas. O descarte do Curso 1 se deu por tratar-se de um curso focado ainda em questões muito básicas relacionadas ao letramento digital, como, por exemplo, o “arrastar e soltar”. Desse modo, consideramos que a análise se tornaria prejudicada, uma vez que boa parte deste curso foge ao foco desta dissertação.

A exclusão dos cursos 3 e 4 se deu porque verificamos que a estrutura dos cursos se mantém comum ao Curso 2. Por este motivo, esse curso se tornou elegível para análise, por representar de maneira significativa as características dos outros cursos.

O Curso 2 é voltado a crianças do segundo ao quinto ano escolar. Ele é organizado em 19 aulas, cada aula com no mínimo uma e no máximo dezesseis fases em formato de tutorial. Essas fases somam um total de 152 (cento e cinquenta e dois) níveis. Dessas dezenove aulas, todas contam com um plano de aula específico, além disso, as aulas são acompanhadas de um arquivo voltado a atividades sem o uso do computador. O Curso 2 conta com oito atividades *off-line* disponíveis para o professor, tendo a função de introduzir questões concretas antes das atividades direcionadas pelos tutoriais.

Para fins de organização, definimos que as aulas serão organizadas nesta dissertação pela notação A (Aula) seguida de um número correspondente à ordem da aula. Assim, a primeira aula será definida por A1, a segunda por A2 e assim sucessivamente. As fases disponíveis em cada aula serão intituladas por F (fase) seguido do nível de cada atividade representado por um número. Por exemplo, a atividade 5 do tutorial da aula 4 será denominado A4-F5.

4.2.3 Área do professor e planos de aula

Como já mencionado anteriormente, o Curso 2 conta com dezenove planos de aula. Nesta dissertação, usaremos a notação P (Plano) seguida do número da aula. Assim, o plano de aula correspondente à primeira aula será definido pela notação P1, o segundo por P2 e assim sucessivamente.

4.3 PROCEDIMENTOS DE ANÁLISE

Como procedimento de análise, elegemos a Análise Textual Discursiva (MORAES, GALIAZZI, 2011), considerando ser uma metodologia de pesquisa qualitativa que transita entre a análise de conteúdo e análise de discurso.

[...] a análise textual qualitativa pode ser compreendida como um processo auto-organizado de construção de compreensão em que novos entendimentos emergem de uma sequência recursiva de três componentes: desconstrução dos textos do *corpus*, a *unitarização*; estabelecimento de relações entre os elementos unitários, a categorização; o captar do novo emergente em que a nova compreensão é comunicada e validada (MORAES, 2003, p. 191).

A análise textual discursiva dá suporte à construção de significados por meio de um conjunto de textos em que o pesquisador deve atribuir sentido e significado ao material textual. O autor ainda acrescenta que, metaforicamente, essa análise é uma “Tempestade de Luz”, em que a partir da desordem emergem os achados do pesquisador. Nesse caminho, seguindo os procedimentos estabelecidos por Moraes (2003, p. 191), elencamos as seguintes etapas de análise: 1 - Seleção do *Corpus*, 2 - Desmontagem dos Textos (unitarização), 3 - Estabelecimento de Relações (categorização) e 4 - Captando o emergente (compreensão renovada do todo). Este último movimento possibilitará a construção do metatexto. Nas palavras de Moraes (2003, p. 191), o metatexto é o resultante de um processo que procura

“explicitar a compreensão que se apresenta como produto de uma nova combinação construídos ao longo dos passos anteriores”.

Dentro da proposta metodológica, as categorias emergentes correspondem aos achados do pesquisador que emergem a partir da análise e relacionamento das narrativas investigadas.

Categorizar, nessa perspectiva conceitual, significa o momento em que o pesquisador assume a função de autor de seus argumentos e inicia o metatexto. A partir da análise, emergiram três categorias. São elas: **Concepções de aprendizagem, Dificultadores do Processo de Aprendizagem e Promotores do Processo de Aprendizagem.**

A primeira categoria corresponde às concepções de aprendizagem, ou seja, como aparece na plataforma a epistemologia subjacente na relação pedagógica proposta. A segunda corresponde às evidências que trazem algum tipo de problema na prática pedagógica com o ensino de pensamento computacional e Ciência da Computação. Como, por exemplo, as questões de tempo de aula que acabam prejudicando o desempenho das atividades de sala de aula. Por fim, a última categoria corresponde às evidências que trazem intervenções e técnicas que os professores e plataforma utilizam em sala de aula para possibilitar a aprendizagem dos conceitos trabalhados. Como, por exemplo, atividades que envolvem resolução de problemas em grupo.

Diante das categorias, emergiram subcategorias que são descritas no quadro abaixo e que norteiam a análise proposta por este capítulo:

Quadro 2- Categorias e Subcategorias

CATEGORIA	SUBCATEGORIA	DESCRIÇÃO DAS SUBCATEGORIAS CATEGORIA
Concepções de aprendizagem	Empirismo Construtivismo	Concepções de aprendizagem, ou seja, a teoria do conhecimento que embasa as relações pedagógicas.
Dificultadores do processo de aprendizagem	Tempo	Desafios relacionados às restrições de tempo de aula.
	Condições Estruturais	Problemas de rede, hardwares, disponibilidade de equipamentos e outros problemas que impactam o desempenho das aulas nesse sentido.

(Continua)

(Conclusão)

	Dificuldades com TI	Desafios que envolvem dificuldades com letramento digital, falta de conhecimento da plataforma e problemas com a linguagem de programação usada pela plataforma.
Promotores do processo de aprendizagem	Interação entre Pares	Estratégias que contemplam ações de interação entre os alunos de forma cooperativa ou colaborativa.
	Atividades Off-line	Estratégias que contemplam atividades sem o uso do computador como recurso para aprendizagem.
	Atividades On-line	Estratégias que contempla o uso das atividades on-line (tutoriais) como recursos para a aprendizagem.

Fonte: autor.

Os trechos das postagens dos professores transcritos estão disponíveis em versão traduzida pelo autor e disponibilizadas na forma de notas de rodapé para melhor esclarecer possíveis dúvidas.

5 ANÁLISE E DISCUSSÃO

Trazemos, neste capítulo, os resultados obtidos da pesquisa proposta por esta dissertação. A análise está norteada por dois grandes campos que se articulam. O primeiro corresponde ao referencial teórico adotado, que permitiu iluminar os achados desta pesquisa. O segundo corresponde ao rigor metodológico, que permitiu sistematizar os dados para a análise e interpretá-los de modo a gerar um produto final, de acordo com os pressupostos da Análise Textual Discursiva.

A análise está organizada em três tópicos, baseados nos objetivos específicos que foram propostos no projeto desta dissertação, aliados às categorias emergentes da análise textual discursiva. O primeiro item apresenta evidências das concepções epistemológicas ou concepções de aprendizagem encontradas na plataforma analisada. O segundo busca discutir os achados acerca dos aspectos promotores do processo de aprendizagem do pensamento computacional. O terceiro e último discute os aspectos dificultadores do processo de aprendizagem evidenciados pelos professores com a prática do pensamento computacional em sala de aula.

A discussão está organizada de forma a contemplar as múltiplas fontes de evidência, fórum, atividades e planos de aula, norteada pelos objetivos. Por fim, apresentamos uma seção que visa, a partir do referencial teórico construído nesta dissertação, levantar indicadores para o professor na atuação com o pensamento computacional, tendo o intuito de trazer à discussão possibilidades que possam ser consideradas para a formação de professor neste contexto de PC.

Antes de partirmos para a análise das fontes de evidência, apresentaremos uma análise descritiva da plataforma *Code.org*, em sua organização e possibilidades para melhor situar o leitor.

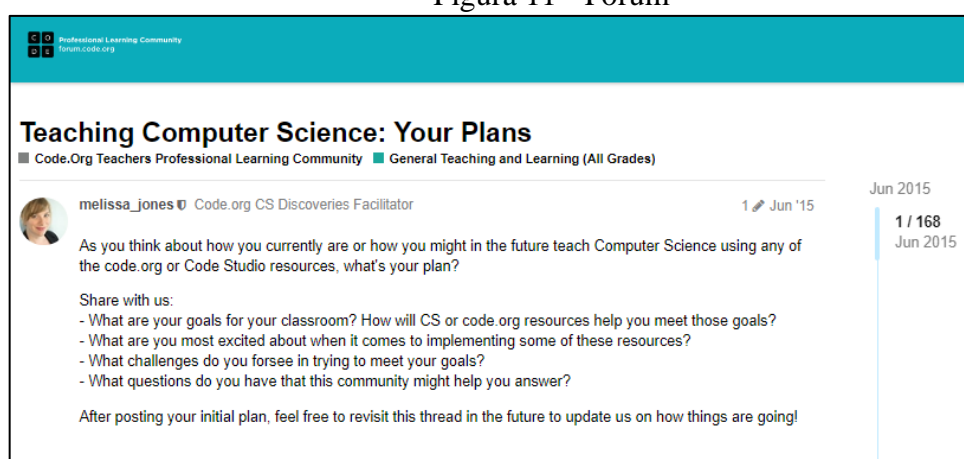
5.1 A PLATAFORMA *CODE.ORG*: UMA VISÃO GERAL

Conforme já mencionado, a *Code.org* é uma instituição não governamental e sem fins lucrativos, que tem por objetivo levar os conhecimentos de Ciência da Computação para todos, em qualquer lugar do mundo. A base da interação na plataforma são os tutoriais (atividades), ambientes que conduzem o participante por etapas no desenvolvimento de conceitos de computação, a partir de atividades direcionadas. Esses ambientes vinculam programação em blocos com temas atuais como games de computador ou filmes que envolvem os participantes em atividades divertidas.

Essa plataforma conta com uma série de recursos exclusivos para o professor, que são utilizados para levar os conceitos de ciência da computação para a sala de aula por meio das iniciativas de formação realizadas pela *Code.org*.

A partir do *site* é possível acessar os tutoriais disponíveis nos cursos e os planos de aula para cada atividade, além disso, o ambiente conta com um fórum em que os professores do mundo todo se comunicam, trocando informações sobre a implementação das atividades do *Code Studio* em sala de aula. Os tópicos desse fórum estão organizados a partir da provocação de um moderador, e, em seguida, uma série de outros comentários dos usuários da plataforma que, em geral, são professores implementando o ensino de pensamento computacional e Ciência da Computação na escola ou professores e outras pessoas interessadas no tema. A imagem abaixo mostra um fragmento da organização do fórum iniciando pela fala do moderador.

Figura 11 - Fórum

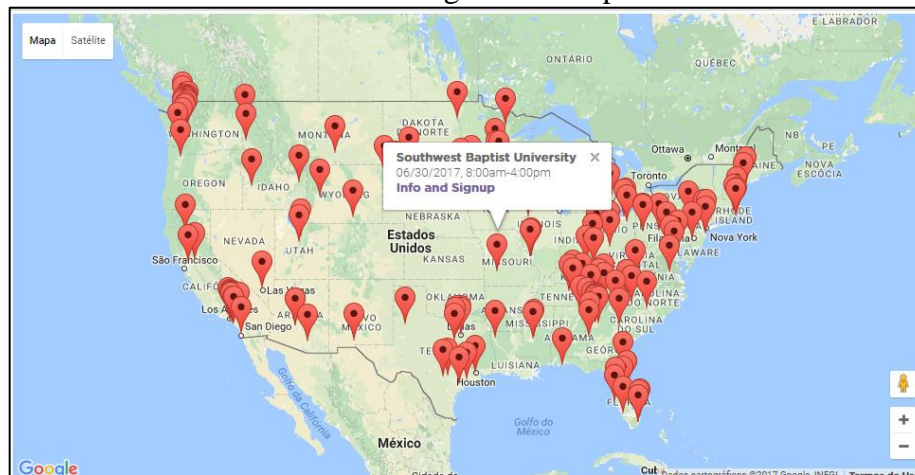


Fonte: <https://forum.code.org/t/teaching-computer-science-your-plans/412>.

É importante destacar que as postagens desses tópicos são basicamente oriundas de *workshops* e formações desenvolvidas pela comunidade *Code.org* nos Estados Unidos, mas que não excluem a participação e interação no fórum de professores, pesquisadores e profissionais da computação interessados em levar o conhecimento de programação para todos.

O mapa representado pela Figura 12 abaixo demonstra a área de localização dos próximos *workshops*. Infelizmente, não há como verificar as formações já realizadas, as quais possibilitaram a maior parte das interações nos tópicos analisados.

Figura 12- Mapa

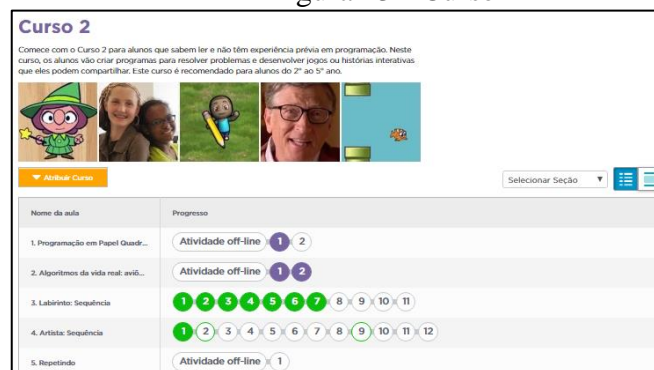


Fonte: <https://code.org/professional-development-workshops>.

Para acessar o “*Teacher Forums*”⁴, o professor necessita estar devidamente cadastrado na plataforma. Após o cadastro, o acesso é liberado por meio do *link* “Educadores”, disponível na página inicial do site *Code.Org*. O acesso está disponível no “Painel do Professor”. Este fórum faz parte do item “Recursos”, disponível na parte inferior da página. O acesso é realizado por meio do título “*Teacher Fóruns*”.

O fórum é dividido em categorias, subcategorias e tópicos para facilitar a busca e a interação com outros participantes. Os cursos possuem recomendação de faixa etária e iniciam com atividades concretas ou de baixa dificuldade e vão aumentando à medida que o usuário completa os desafios. Esse processo se assemelha ao de um game, assim dando a sensação à criança de que está jogando, e não realizando um curso. Parte da estrutura do Curso 2 é demonstrada na imagem a seguir.

Figura 13 - Curso 2



Fonte: <https://studio.code.org/s/course2>.

⁴ Disponível no endereço <http://forum.code.org/c/plc>.

Conforme se pode perceber na figura 13, os cursos também contam com atividades *off-line*. Estas atividades são mediadas pelo professor e têm por objetivo iniciar a aprendizagem por meio de experiências concretas.

Cada aula é introduzida por um vídeo motivacional e, em seguida, por um desafio. As aulas possuem uma série de desafios que aumentam a dificuldade à medida que o usuário avança.

Figura 14 - Fase 4



Fonte: <https://studio.code.org/s/course2/stage/6/puzzle/4>.

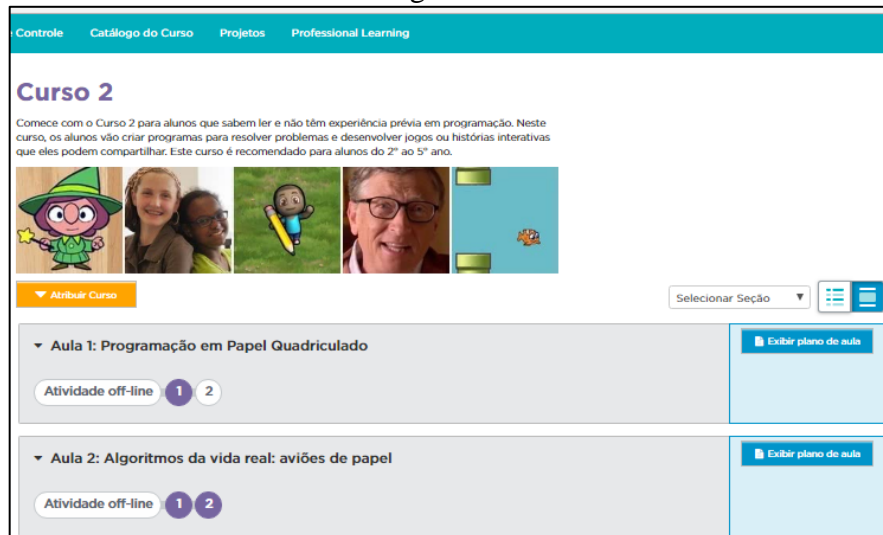
As atividades usam personagens conhecidos pelas crianças e vão introduzindo conceitos de computação com o objetivo de desenvolver o raciocínio lógico e o aprendizado da Ciência da Computação. Do mesmo modo que os cursos, os tutoriais “Hora do Código” buscam ensinar a criança por meio da resolução de problemas e da experiência de programar personagens conhecidos, porém, neste caso, esses tutoriais foram desenvolvidos para serem resolvidos, em média, em uma hora. Diferentemente do curso completo, que possui várias aulas e fornece um progresso mais lento, em uma modalidade extensiva.

Todos os cursos podem ser acessados por qualquer criança na *web*, porém, por sua estrutura, fica evidente que a proposta da *Code.Org* é fornecer aos professores subsídios para implementarem o ensino de programação e Ciência da Computação na sala de aula, uma vez que, mesmo na tela do aluno, as fases mostram as atividades *off-line*.

A área do professor consiste em um *dashboard* em que o professor pode criar, gerenciar ou editar turmas, além de poder visualizar o progresso dos alunos de cada turma cadastrada. Para acessar a área do professor da *Code.org*, o professor deverá realizar um simples cadastro na plataforma que habilitará as funcionalidades específicas do painel do professor.

Nele, o professor também poderá acessar todos os cursos disponíveis pela *Code.org*. Porém, além da visualização normal, o professor poderá encontrar planos de aula para cada uma das aulas disponíveis no tutorial. A Figura 15 mostra parte do detalhamento do Curso 2 a partir da visualização do professor.

Figura 15 - Curso 2



Fonte: <https://studio.code.org/s/course2>.

Ao acessar o plano de aula, o professor tem acesso aos tutoriais da mesma forma que os alunos, porém conta com a possibilidade de exibição do plano de aula. Os planos de aula seguem um padrão, oferecendo ao professor estratégias para o desenvolvimento da aula, tendo a estrutura descrita no Quadro 3.

Quadro 3 - Estrutura dos planos de aula

Seção	Conteúdo
Título	Fornece o título da aula seguido do tempo necessário sugerido.
Visão Geral da Lição	Apresenta de maneira panorâmica a atividade que será desenvolvida na aula.
Lições Objetivas	Apresenta os tópicos referentes aos objetivos de cada aula.
Começando	Fornece uma sugestão de introdução à aula, buscando relações com atividades anteriores ou conhecimentos prévios dos alunos.
Atividade	Fornece o nome da atividade proposta, on-line ou off-line.
Aprendizagem Estendida	Fornece outra atividade necessariamente <i>off-line</i> , que o professor poderá utilizar para trabalhar melhor os conceitos desenvolvidos na aula.
Conexões e Informações básicas	Nessa seção, o plano de aula traz uma série de conexões com outras áreas do conhecimento, como ciências e matemática, linguagem das artes e padrões estabelecidos por organizações como a CSTA, fornecendo subsídios para que o professor possa pensar em propostas que vão além da estabelecida no plano de aula.

Fonte: Autor.

5.2 CONCEPÇÕES DE APRENDIZAGEM EMERGENTES NA PLATAFORMA

Conforme apresentado nos capítulos anteriores, podemos dizer que a maneira de se entender a gênese da aprendizagem está baseada em três modelos epistemológicos. São eles: o empirismo, o inatismo e o construtivismo (Becker, 1999). Essas formas de conceber a aprendizagem estão refletidas na maneira como a prática pedagógica é conduzida e na forma como professor e aluno se relacionam na produção do conhecimento.

Uma visão pedagógica empirista fundamenta-se na crença de que o professor é o sujeito conhecedor e o aluno é tábula rasa. Nessa visão, a função do professor é a mais importante no processo educativo, pois é dele a responsabilidade de fazer o aluno aprender. Nessa perspectiva, o aluno nada sabe, e o professor, por meio da exposição e de exercícios repetitivos, irá inserir no aluno os conhecimentos já prontos trazidos por ele, pois todo conhecimento seria unicamente fruto da experiência dos sentidos. Conforme Becker (1993), a epistemologia empirista se traduz em um modelo pedagógico diretivo, ou seja, nesse modelo a relação do conhecimento ocorre do objeto para o sujeito. “O conhecimento é algo que vem do

mundo do objeto (meio físico ou social); portanto o mundo do objeto é determinante do sujeito, e não o contrário” (BECKER, 1993, p. 12). A relação pedagógica em um modelo empirista é geralmente pautada por condutas heterônomas, em que o professor age de maneira autoritária, e o aluno de maneira submissa.

Em outra perspectiva, a visão pedagógica baseada na epistemologia inatista pressupõe que o conhecimento é fruto exclusivo do sujeito e depende da sua bagagem hereditária para que a aprendizagem ocorra. Ao contrário da visão anterior, a prática pedagógica, nessa perspectiva, entende que o professor pouco tem a fazer, além de fazer emergir uma capacidade que é inerente ao sujeito. Uma pedagogia baseada nesse modelo reflete a ideia de o professor estabelecer, por exemplo, no início da sua disciplina, os alunos que, em sua visão, irão ser aprovados ou reprovados. Becker (1993) refere-se à prática pedagógica baseada nesse modelo como pedagogia não diretiva, ou seja, a relação do conhecimento ocorre do sujeito para o objeto. Nas palavras de Becker (1993),

Podemos dizer que aprioristas são todos aqueles que pensam que as condições de possibilidade do conhecimento são dadas na bagagem hereditária: de forma inata ou submetida ao processo maturacional, mas, de qualquer forma predeterminadas ou a priori – isto é estão aí, dadas como condição de possibilidade. (BECKER, 1993, p. 15).

Ou seja, uma prática pedagógica baseada nesse modelo geralmente isenta o professor da responsabilidade de “ensinar” e reforça a ideia de que o centro da aprendizagem é o aluno. Nas palavras de Becker (1993), a pedagogia não diretiva é uma espécie de aprendizagem pelo “deixar fazer”.

Uma terceira concepção de aprendizagem se refere ao construtivismo. Diferentemente das duas anteriores, neste caso o conhecimento não está nem no sujeito, nem no objeto, mas é fruto da relação dos dois. Becker (1993) define essa relação pedagógica como pedagogia relacional. Em uma pedagogia relacional, valoriza-se a interação entre os sujeitos, as relações de troca e a ideia de que o professor não é o centro do conhecimento, mas é, por outro lado, um sujeito que provocará a construção do conhecimento.

O professor construtivista recusa a ideia de que o conhecimento pode ser transmitido e entende que o conhecimento é fruto da interação do sujeito com o objeto do conhecimento. Nessa visão epistemológica, refuta-se o autoritarismo e entende-se que o professor aprende ao ensinar e o aluno ensina ao aprender. É uma relação constante de respeito mútuo entre as partes que estão dispostas a conhecer.

A partir da retomada desses conceitos, podemos entender a análise das fontes de evidência e a construção dos dados. Assim, nesta seção buscamos identificar os modelos epistemológicos encontrados nas fontes de evidência.

Becker (1993) argumenta que a ação do professor traz consigo uma concepção de aprendizagem, seja ela consciente ou não. Nosso papel, nesse sentido, foi verificar a partir das fontes as concepções epistemológicas subjacentes na plataforma.

5.2.1 Planos de aula

Os planos de aula nos fornecem diversos indícios da epistemologia subjacente à plataforma *Code.org*. Nesse sentido, convém ressaltarmos inicialmente os seguintes extratos:

Algoritmo - Diga comigo: Al-go-rit-mo. Uma lista de etapas que você pode seguir para finalizar uma tarefa (P1).

Programa - Diga comigo: Pro-gra-ma. Um algoritmo que foi codificado em algo que pode ser executado por uma máquina (P1).

Na aula seguinte, o plano de aula P2, sugere que o professor deveria retomar essa questão e solicita que o professor faça os alunos repetir mais uma vez.

Esta lição tem uma palavra de vocabulário que é importante rever: Algoritmo - Diga comigo: Al-go-rit-mo. Uma lista de etapas que você pode seguir para finalizar uma tarefa (P2).

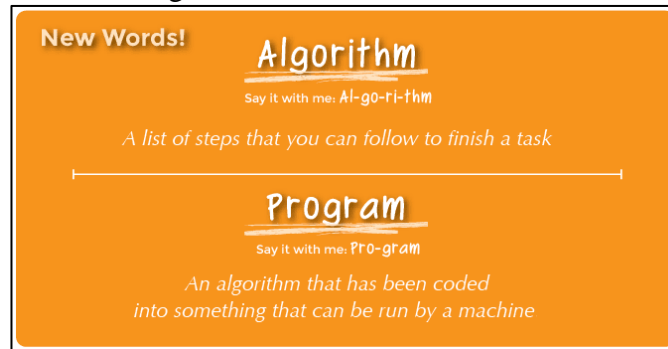
Esta lição tem uma palavra nova e importante: Binário - Diga comigo: Bi-ná-rio. Uma maneira de representar informações usando apenas duas opções (P14).

Observamos que essa estratégia de introdução de uma nova palavra e, em seguida, a repetição pela turma ocorre nos planos de aula, P1, P2, P5, P9, P11, P15, P16, P19, referente às palavras algoritmos, programas, *loops*, *debugging*, condicionais, binário, evento e pegadas digitais. Em todos os casos, os planos de aula sugerem que o professor apresente o nome e o conceito e, em seguida, trabalhe ações para que os alunos possam aprender o “conteúdo” proposto para a aula.

A partir dessa estratégia de apresentar a palavra e a sua definição, identificamos que há indícios de uma abordagem empirista, em que o aluno não sabe e o professor transmite a ele a informação, do mais simples ao mais complexo, para garantir que todos saibam a mesma coisa. Não aparece a possibilidade de os estudantes construírem seus próprios conceitos ou explicações sobre o que estão fazendo ou aprendendo.

A figura a seguir demonstra como o vocabulário da aula é apresentado nos planos de aula.

Figura 16 - Vocabulário da aula



Fonte: P1

Essa abordagem presente nesses planos de aula evidenciam uma abordagem epistemológica que se aproxima do empirismo, pois pelo plano o professor deverá apresentar o conceito que será trabalhado com os alunos a partir de exercícios.

Seguindo nessa linha que se aproxima de uma pedagogia diretiva, apontamos ainda o seguinte trecho.

Finalizar a revisão perguntando sobre as coisas favoritas dos alunos ajuda a deixar uma impressão positiva do exercício anterior, aumentando a excitação para a atividade que você está prestes a apresentar (P5).

Nesse trecho, podemos salientar a palavra “exercício” e também o fragmento “para a atividade que você está prestes a apresentar”. Embora não se tenha certeza da maneira como o professor irá interpretar, ou qual foi o intuito na elaboração desse plano, podemos perceber certa linearidade e dependência em relação aos exercícios e às atividades, todas propostas pelo professor. Atividades estas que o professor vai dando sequência à medida que se avança por um “conteúdo”.

Outras evidências, nesse sentido, podem ser encontradas no trecho a seguir que se refere à atividade de avaliação proposta ainda no plano de aula P5:

Entregue a folha de trabalho intitulada "Getting Loopy" e permita que os alunos completem a atividade de forma independente após as instruções terem sido bem explicadas. Isso deve ser familiar, graças às atividades anteriores (P5).

Nessa orientação, também percebemos uma perspectiva empirista quando propõe que o professor apresente detalhadamente as instruções, em linearidade, considerando que os

conhecimentos anteriores garantirão a aprendizagem dos próximos conteúdos. Ainda vemos que o aluno tem um papel passivo de cumprir a ordem do professor, que deve ser detalhadamente explicada.

É preciso esclarecer certa incoerência em se pensar planos de aula e um modelo de formação que evidencie uma pedagogia diretiva. A história do conceito do pensamento computacional, resgatada a partir dos primeiros constructos teóricos de Papert (1980), mostram que o aprender em ambientes como o *Code.org* deve privilegiar o aprendizado pela exploração e pela construção do conhecimento.

Convém aqui trazermos o argumento inicial em que Papert (1980) propunha que a filosofia logo possibilitaria o aprendizado de matemática naturalmente, como se a criança estivesse na Matelândia. O *Code.org* traz estruturas que evidenciam, assim como o *Scratch* e outros, similaridades que são subjacentes ao percurso da filosofia LOGO. Contudo, a partir da análise desses extratos, notamos um distanciamento entre uma proposta de aprendizagem inicialmente trazida na filosofia LOGO e a que podemos observar nos planos da *Code.org*.

Outra questão importante a salientar, além das atividades que sempre aparecem em níveis, é o fato de alguns planos de aula proporem avaliações escritas como forma de verificar a aprendizagem, conforme a seguinte:

Entregue a planilha de avaliação e permita que os alunos completem a atividade de forma independente após as instruções serem bem explicadas. Isso deve ser familiar, graças às atividades anteriores (P15).

O fragmento acima evidencia que a avaliação, no sentido que aparece, diz respeito a um instrumento de aplicação individual, em que o professor explicará bem as questões antes de o aluno fazer. É claro que, nesse modelo, a avaliação se refere a uma espécie de prova e não há indicadores de ser um instrumento de retroalimentação do processo de ensino e aprendizagem.

Vasconcellos (2008) critica a avaliação no modelo tradicional (prova individual), argumentado que a prova é um instrumento de coerção do aluno ao passo que precisaria ser um instrumento que favorecesse o aprender a aprender, a valorização do erro como um percurso para a construção do conhecimento. No sentido de perceber a avaliação como um meio e não um fim, Kebach (2007), à luz da teoria de Piaget, argumenta que a avaliação é um instrumento do professor pesquisador que busca avaliar, de fato, o que o aluno construiu.

Se utilizarmos, não as provas, mas a forma de aplicação das mesmas em sala de aula, ou seja, propusermos uma situação a ser pensada coletivamente, procurarmos desequilibrar as crianças com contra-argumentos e novas investidas de organização sobre os objetos, estaremos proporcionando momento de real pesquisa em sala de aula. (KEBECH, 2007, p. 47).

Convém ainda refletir sobre o extrato P15, em que o professor é orientado a apenas entregar a planilha de avaliação e solicitar que os alunos a façam individualmente. Ao que nos parece, esse fragmento, principalmente por estar presente em um plano de aula, reforça a ideia de que esse processo de avaliação é apenas um fim e não um recurso que o professor irá utilizar em sala de aula para criar novas problemáticas, investigar as hipóteses dos alunos e, a partir daí, criar novas problemáticas. Esse tipo de avaliação aparece em todos os planos de aula como uma retomada às atividades.

Pelo fato de a atividade ser entregue no final da aula, ser feita individualmente e ainda ter um tempo para o término, nos parece que essa atividade se configura como uma avaliação tradicional, sem dar outras opções aos alunos e professores. A Figura 17, extraída do plano de aula P1, mostra o estilo das questões propostas.

Figura 17 - Avaliação

Name: _____ Date: _____

U
Unplugged **Graph Paper Programming**
Assessment Worksheet

You have just learned how to create algorithms and programs from drawings, and how to draw an image from a program that someone gives to you. During the lesson, you worked with other people to complete your activities. Now you can use the drawings and programs below to practice by yourself.

Use the symbols below to write a program that would draw each image.

→ Move One Square Forward ← Move One Square Backward ↑ Move One Square Up ↓ Move One Square Down ✎ Fill-In Square with Color

Start Here

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20

Fonte: P1

É importante perceber que essas questões estão exatamente no mesmo formato que as presentes no plano de aula P1. É como se fosse a atividade prática convertida em um instrumento de avaliação formal, semelhante a uma prova. Os planos não informam ao professor que métricas estabelecerem ou se, de alguma forma, essas questões são revertidas em notas, ou mesmo discute ou orienta como o professor pode analisar essas avaliações. Convém ressaltar que os planos de aula trazem inúmeras atividades e propostas que devem ser realizadas em grupo, conforme trataremos mais adiante, porém a avaliação não privilegia momentos

similares. Dessa forma, há certa incoerência ao se propor atividades que são realizadas em grupo durante todos os planos de aula analisados, mas que apresentam avaliação individual.

Os fragmentos apresentados até aqui evidenciam que a maneira como as atividades são propostas convergem para um modelo de aprendizagem baseado em uma pedagogia diretiva.

No que se refere aos planos de aula, por exemplo, observamos que a organização das aulas se dá por meio da apresentação de conceitos, repetição e prática. Um modelo pedagógico pautado nessas premissas vem ao encontro de uma pedagogia clássica em que o professor é o protagonista da aprendizagem, enquanto o aluno é receptor. Nesse modelo de aprendizagem, o aluno não tem o que construir, pois os conceitos já estão prontos e serão absorvidos pelo aluno, por meio da experiência ou da exposição. Assim, as informações são passadas aos estudantes a partir da explicitação do conceito para depois oferecer exercícios de fixação. Em todos os planos de aula em que é apresentado um novo conceito, observamos que o professor sempre apresenta o vocabulário da aula, explica o conceito e só depois os alunos têm a possibilidade de experimentar por meio de atividades práticas.

A partir dessas questões levantadas, podemos argumentar que essa abordagem poderia ser deslocada para um enfoque que se aproximasse, do ponto de vista teórico, das propostas de Papert (1980, 2006) e Piaget (1978). Ou seja, os planos de aula poderiam sugerir ao professor que propusesse experimentos práticos e atividades antes da apresentação do conceito e, a partir da ação do aluno sobre os objetos, permitir que o conceito fosse construído. Além disso, poderiam oferecer maior flexibilidade e menos linearidade no que diz respeito às atividades propostas, pois da forma como estão apresentadas há um padrão a ser seguido, o que pode excluir a criatividade e a liberdade do professor. Essa sequência de passos de exposição e repetição remete a um professor que possivelmente acredita que o conhecimento pode ser transmitido e, nesse sentido, esforça-se para fazê-lo. É preciso considerar, portanto, que ensinar não é transmitir conhecimentos, mas criar as possibilidades para que ele seja construído (FREIRE, 1996).

O professor tradicional geralmente pauta sua aula em exposições seguidas de exercícios, porque ele acredita que não é necessário criar significados, mas apenas depositar o conhecimento pronto na mente do aluno para que, mais tarde, esses conhecimentos sem lastros possam ser utilizados ou postos em prática em uma situação futura. É uma espécie de ensino bancário, como argumenta Freire (1996), pois os conhecimentos são depositados, para que mais tarde possam ser sacados. Nas palavras de Becker:

O professor considera que seu aluno é tabula rasa não somente quando ele nasceu como ser humano, mas frente a cada novo conteúdo estocado na sua grade curricular, ou nas gavetas de sua disciplina. A atitude, nós a conhecemos. O alfabetizador considera que seu aluno nada sabe em termos de leitura e escrita e que ele tem que ensinar tudo. Mais adiante, frente à aritmética, o professor, novamente, vê seu aluno como alguém que nada sabe sobre somas e subtrações. (BECKER, 1999, p. 90).

Do mesmo modo, a avaliação poderia então privilegiar um momento de retroalimentação do processo de aprendizagem e, de forma alguma, excluir a atividade em grupo. Convém salientar que aprendizagem em um modelo construtivista ou construcionista não estipula etapas e níveis a serem vencidos, pois a aprendizagem nesse modelo epistemológico não é linear, ela acontece em uma perspectiva complexa. Do contrário, a aprendizagem norteada por uma epistemologia empirista pressupõe que todo conhecimento deve ocorrer por meio de fases, em que o sujeito, necessariamente, deve passar por todas, das mais simples às mais complexas. É como imaginar que o processo de aprendizagem é uma escada com degraus iguais e que todo o aluno tem o mesmo tamanho de passos, a mesma velocidade e capacidade para subir a escada exatamente da mesma forma.

No que se refere a uma epistemologia construtivista, encontramos nos planos de aula alguns breves fragmentos que nos permitem reconhecer um esforço para aproximar o professor de uma prática pedagógica em um modelo relacional, como o seguinte.

A sala de aula pode estar repleta de sugestões até este ponto. Se a turma tiver a essência do exercício, este é um bom lugar para discutir maneiras alternativas de preencher a mesma grade. Se ainda houver confusão, guarde essa peça para outro dia e trabalhe com outro exemplo. (P1).

Trechos como esse, expressos no plano de aula P1, mostram um movimento importante que instiga o professor a privilegiar momentos de interação e trocas, o que sinaliza aproximação com um modelo de aprendizagem que não se fundamenta apenas na transmissão, mas acredita que o conhecimento também é gerado a partir da interação dos alunos.

Você conhece melhor a sua sala de aula. Como professor, decida se os alunos devem fazer isso individualmente ou se os alunos devem trabalhar em duplas ou em pequenos grupos. (P2).

Este fragmento, presente no plano de aula P2, fornece sugestões para que o professor possa escolher uma forma de trabalhar, permitindo que os alunos trabalhem em duplas ou individualmente. Essa proposta também aposta na autonomia e no conhecimento do professor. Seguindo nessa linha temos o seguinte:

Este é um ótimo momento para revisar a última lição que você passou com sua turma. Você pode fazer isso como o grande grupo ou pedir aos alunos que discutam com um parceiro de cotovelo. (P9).

No extrato presente em P9 observamos, mais uma vez, uma relação que permite ao professor escolher a melhor forma de trabalho, sugerindo que o profissional discuta com a turma toda ou que os alunos trabalhem em pares, discutindo as atividades propostas. Podemos entender, a partir da análise desses fragmentos, que há uma preocupação em as atividades propostas não estarem exclusivamente centradas no professor e que o professor permita momentos de construção de conhecimentos em uma relação aluno – aluno.

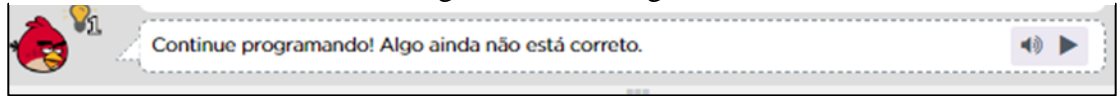
Por outro lado, observamos que os planos de aula parecem não apresentar inovação ou novas possibilidades de aprendizagem frente à tecnologia. Eles estão praticamente pautados em explicação do professor, problematização de questões, exercícios, avaliação e retomadas, mesmo que existam inúmeras atividades desconectadas, práticas e momentos de trabalhos em grupo.

Essas abordagens poderiam ser pautadas em metodologias que privilegiassem a interação, a exploração, a construção de hipóteses e que, principalmente, o professor pudesse ser deslocado do protagonismo da sala de aula. Há indícios de que o professor, nas aulas propostas a partir dos planos de aula analisados, tem um papel de protagonismo, ao passo que poderia se transferir esse protagonismo por ações que privilegiassem a participação dos alunos e, principalmente, a construção do conceito pelo aluno, antes das explicações ou, melhor, das intervenções dos professores.

5.2.2 Atividades/Tutoriais

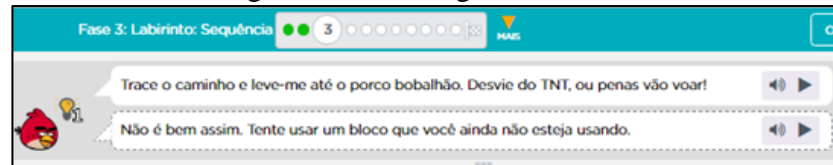
Seguindo nessa linha, analisamos as Atividades (Tutoriais) e percebemos que as aulas do Curso 2 estão organizadas de forma a começarem por atividades desconectadas e, em seguida, o professor deverá utilizar as atividades guiadas para a introdução dos conceitos de programação. Nesse sentido, observamos que a primeira evidência epistemológica se dá pelas mensagens de erro que a plataforma apresenta ao usuário à medida que ele vai executando códigos que não estão corretos na totalidade.

Figura 18 - Mensagem de erro 1



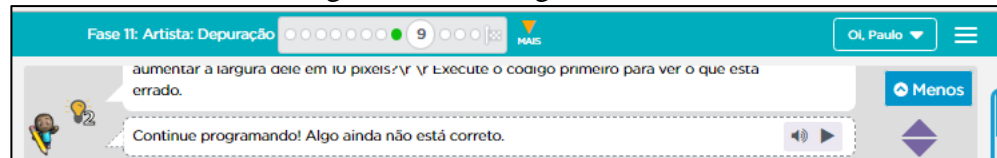
Fonte: A3-F1.

Figura 19 - Mensagem de erro 2



Fonte: A3-F3.

Figura 20 - Mensagem de erro 3

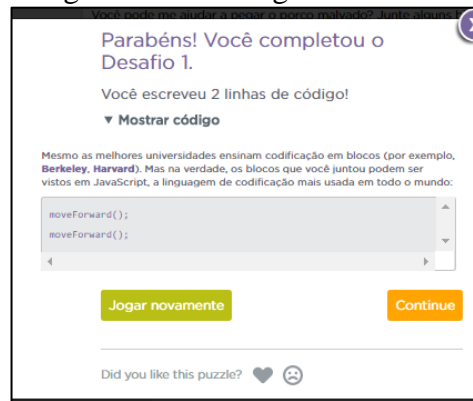


Fonte: A11 - F9.

Essas mensagens simplesmente informam ao usuário que há um erro, porém não permitem um processo meta-reflexivo que possibilite ao sujeito criar novas hipóteses. Nesse caso o acerto se dá provavelmente por tentativa e erro, e não pelo processo de acomodação, como na visão proposta por Piaget (1978). As mensagens, nesse sentido, não convidam o sujeito a pensar em novas hipóteses e a olhar o erro como parte de um processo de aprendizagem ou uma oportunidade para reorganizar o pensamento.

Já no caso dos acertos, há um reforço positivo que instiga o aluno a sempre acertar. O erro é sempre visto através de uma abordagem negativa, enquanto o acerto é visto como o foco principal da atividade, aproximando, ao que tudo indica, um embasamento empirista da plataforma.

Figura 21 - Mensagem Desafio 1

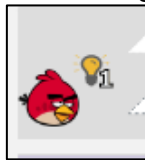


Fonte: A3-F1

Em uma perspectiva construtivista, o erro faz parte do processo de aprendizagem e permite que a criança crie novas possibilidades através da interação com o objeto do conhecimento.

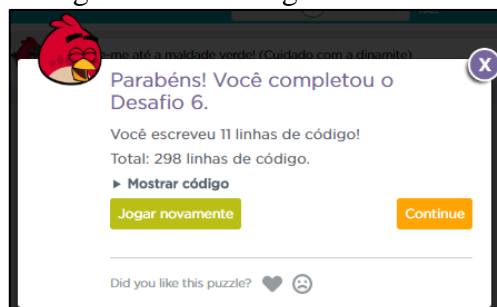
Essas questões de erro e acerto também são expressadas pela representação do personagem em cada um dos casos, o que, mais uma vez, nos remete a uma abordagem empirista de reforço, pela expressão do personagem e pelo texto da mensagem.

Figura 22 - Personagem - erro



Fonte: Curso 2.

Figura 23 - Mensagem de Acerto



Fonte: A3-F6.

Na figura 22, o personagem deste tutorial aparece com um aspecto infeliz, pois se trata de uma mensagem de erro, enquanto a figura 23 mostra o personagem feliz, porque o usuário acertou o que estava sendo solicitado.

Além disso, os tutoriais são rígidos e não permitem a livre exploração do sujeito, como em outras plataformas disponíveis para o mesmo fim. O sujeito é sempre conduzido por um tutorial para só no fim poder criar o seu próprio game ou história. Isso de maneira geral obstrui a livre criação da criança, pois pressupõe que ela não conhece suficientemente os recursos para criar seus produtos sem a prévia instrução. Essa relação pode ser observada a partir do trecho extraído do plano de aula P17 e que se refere a uma atividade *on-line*:

Esta é a atividade plugada mais livre do curso. Na fase final, os alunos têm a liberdade de criar uma história própria. Você pode fornecer diretrizes estruturadas sobre o tipo de história a ser escrita, especialmente para alunos sobrecarregados por muitas opções (P17).

Ou seja, no trecho anterior observamos que o sujeito só terá oportunidade de realizar suas criações depois de percorrer um longo caminho (16 aulas) guiado pelo professor e por atividades prévias que não permitiam em nenhum momento a livre criação. Na visão de Papert (1980, 2006), ambientes como esse deveriam ser utilizados para instigar a criatividade e a capacidade do aluno em criar e explorar, e não em uma perspectiva instrucionista.

O tutorial subestima a capacidade de criação do sujeito. Se compararmos essa estrutura com a maneira como outras plataformas são organizadas, percebemos que no *Scratch* ou no *AppInventor*, por exemplo, o sujeito pode criar qualquer coisa desde o primeiro contato com a ferramenta, sem que antes tenha que passar por etapas previamente definidas em um tutorial.

Na *Code.org* as atividades são lineares, não possibilitam múltiplas escolhas (do ponto de vista de um game) e sempre são pré-requisitos da próxima. Essa abordagem se aproxima, mais uma vez, de um ambiente de aprendizagem pautado em uma epistemologia empirista.

A partir da análise dos tutoriais disponíveis nos cursos, as mensagens de erro e o fluxo com o qual o aluno percorre os tutoriais, é possível identificar que há aspectos que nos induzem a crer que a plataforma foi desenvolvida com características sutis de gamificação.

De acordo com Fardo (2013), o termo gamificação se refere à utilização dos princípios dos *games* para proporcionar o envolvimento e a motivação de pessoas fora do contexto de um *game*. Ao analisar os cursos da plataforma, podemos observar alguns elementos de gamificação, por exemplo, a presença de níveis em que o aluno deve vencer e que ganham complexidade à medida que os desafios vão sendo resolvidos.

Um sistema de recompensa que dá ao aluno um certificado ao término do curso e mensagens que estimulam o aluno a vencer o desafio. O número da fase, que é mostrado no topo de cada tela, é um indicativo de que o usuário está avançando nos desafios. Embora os

cursos não contenham um *score*, a própria exibição do nível pode ser entendida como um tipo de pontuação que indica em que fase o usuário está. Os *feedbacks*, que também são elementos presentes nos games, são exibidos na tela sempre que o aluno executa a atividade. Esses *feedbacks*, contudo, estão restritos a informar se o usuário acertou ou errou a atividade. Conforme já discutido anteriormente, as figuras 22 e 23 restringem-se a informar o erro ou o acerto, referindo-se ao erro como algo negativo e ao acerto como algo positivo.

Nesse sentido, cabe salientar que a gamificação muitas vezes é aplicada de modo comportamentalista, ou seja, a aprendizagem ocorre por meio de condicionamentos e tentativa e erro. É comum os conceitos de gamificação serem aproximados na prática de uma epistemologia empirista, principalmente por serem utilizados de maneira ingênua e por atribuírem especial significado à pontuação e recompensas.

De maneira geral, por exemplo, a tarefa de gamificar uma aula poderia resumir-se em transformar as provas em missões e os pontos em moedas, porém essa abordagem mantém uma mesma epistemologia, sem que haja mudança de paradigma. Por outro lado, Fardo (2013) contrapõe essa abordagem ao aproximar a gamificação da perspectiva sócio-histórica de Vygotsky, o que nos dá subsídios para pensar um sistema que permita ao aluno aprender com o jogo mais do que aprenderia sem o seu auxílio.

Na visão de Fardo (2013), é possível aplicar os conceitos de *games* e utilizar a gamificação em uma situação de vida real a partir de uma perspectiva interacionista, ou seja, sem que a aprendizagem ocorra de forma exclusivamente comportamentalista. Para isso, seria necessário que os *feedbacks* pudessem, segundo o autor, atuar na Zona de Desenvolvimento Proximal, ou seja, esses *feedbacks* deveriam possibilitar ao aluno um desenvolvimento maior do que apenas por meio da realização das atividades.

Em uma perspectiva piagetiana, é possível pensar no *feedback* como uma forma de provocar desequilíbrios. Ou seja, as mensagens do sistema poderiam levar em consideração a maneira que o usuário tentou resolver os problemas e, a partir disto, fornecer outras possibilidades de resolução. Para este tipo de auxílio, bastaria que a plataforma estivesse munida de um sistema que previsse ao menos as mais prováveis possibilidades de resposta e, a partir disso, levasse a uma resposta ou intervenção mais próxima do ideal para cada situação.

A plataforma, do ponto de vista das atividades propostas, precisa avançar, tanto em estrutura como em possibilidades para os usuários. Uma questão a ser discutida, por exemplo, é a restrição das atividades na plataforma, pois à medida que o aluno avança nas atividades, os planos de aula escasseiam em estratégias e a aprendizagem fica por conta da interação exclusiva do aluno com a ferramenta. Um *software* como esse poderia ser mais útil se fosse capaz de

“aprender” com a interação do aluno com a ferramenta e, a partir disso, criar novas possibilidades e desafios aos sujeitos.

Os tutoriais e os planos de aula convergem para uma noção de aprendizagem em que é preciso passar de um nível de menor complexidade para um de maior complexidade, passando necessariamente por fases específicas e que todos os alunos deverão passar por elas da mesma forma. Esses conceitos de aprendizagem se configuram em uma prática que reproduz uma pedagogia transmissiva. Em contrapartida, as proposições de Papert (1980, 2006) sempre estiveram atreladas à introdução da linguagem de programação como um recurso que possibilitasse a transformação da educação, vencendo as perspectivas de ensino tradicional.

Tanto o construtivismo como o construcionismo entendem a aprendizagem como um movimento do indivíduo e que o aprender deve acontecer com o mínimo de ensino possível, pois a aprendizagem não depende de ensino como transmissão, mas sim das alterações das estruturas mentais de quem aprende. Nesse sentido, observamos nos planos de aula um excessivo papel do professor, que conduz, em geral, de maneira expositiva as aulas, sem necessariamente possibilitar momentos de exploração e espaços que permitam ao aluno modificar suas próprias estruturas por meio da descoberta, exploração e construção de hipóteses.

De maneira geral, embora boa parte da plataforma e dos planos de aula tenham indícios de uma epistemologia que se aproxima do empirismo, há, por outro lado, evidências de uma metodologia que privilegie também o aprendiz com o outro. Basta ver as atividades em grupo e os planos de aula que sugerem esse tipo de atividade, embora atuar em grupo não garanta a cooperação, mas é um princípio de ações e interações em um nível de menor coerção. Nessa perspectiva, identificamos um construtivismo emergente, pois o aprendizado não parece ocorrer apenas do professor para o aluno, mas também entre alunos.

5.2.3 Fórum

Não houve dados suficientes para se analisar as concepções de aprendizagem presentes a partir dos extratos do fórum, embora se possa supor que há evidências de concepções construtivistas a partir da análise dos extratos que tratam das estratégias de cooperação e colaboração apresentadas na próxima seção.

5.3 ASPECTOS PROMOTORES DO PROCESSO DE APRENDIZAGEM

A prática pedagógica é rica em possibilidades, intervenções e recursos que permitem ao professor meios para promover a construção do conhecimento. A partir das categorias emergentes, identificamos que, basicamente, os aspectos promotores da aprendizagem presentes na plataforma consistem em: “Interação entre Pares”, “Atividades *On-line*” e “Atividades *Off-line*”.

Enquanto a primeira consiste em uma mescla de relatos sobre interação entre os alunos, a segunda apresenta o uso de atividades *on-line* disponíveis na plataforma e a última apresenta as atividades *off-line*, ou *unplugged*, atividades de desenvolvimento do pensamento computacional, que não necessitam do computador para serem desenvolvidas.

5.3.1 Interação entre pares

A partir das fontes analisadas, foi possível observar, na categoria Interação entre Pares, uma série de menções que nos levam a entender que há um esforço presente nos planos de aula em fornecer estratégias diversas que promovam a interação entre os alunos da turma, como a seguinte:

Sugerimos que você alterne entre fazer perguntas a toda a classe e ter alunos falando sobre suas respostas em pequenos grupos (P2).

Ao analisamos o trecho de P2, observamos que o professor é instigado a realizar discussões em grupos. Em P3, o plano de aula sugere o uso da programação em pares, ou seja, que os alunos possam trabalhar em duplas para melhor desenvolverem o aprendizado por meio da interação, como podemos verificar:

Lembre-se e aplique as regras da programação em pares. Use a programação de pares para completar tarefas colaborativas com ou sem um computador. Identifique situações em que as regras da programação em pares não sejam seguidas (P3).

Esse movimento também se observa como uma estratégia bastante frequente na fala dos professores no fórum, as quais serão trazidas na próxima seção. Nesse caminho, as falas dos professores a seguir relatam a estratégia de programação “Ask 3” e o uso de um monitor para ajudar no desenvolvimento das atividades. Estratégias que, além de possibilitar aos alunos interação e trocas de experiências, permitem que o professor esteja liberado para ajudar em

problemas mais específicos. Esses relatos exibem, além de um exemplo de trabalho cooperativo, uma estratégia para vencer os desafios de turmas grandes. Por exemplo:

Na minha sala de aula, eu uso o "Pergunte a 3 e então para mim". Ao fazer isso, os alunos são capazes de conversar com os outros e tentar encontrar soluções em conjunto. Eu também tenho alunos que já estão programando e eles enviam seu trabalho para membros da equipe e *admin* para olhar. Eles também têm outros alunos a jogar os seus jogos. Isso lhes dá um grande impulso de confiança e mostra que todo mundo é bom em alguma coisa. (T2-P128)⁵.

No ano passado eu usei o seguinte: - Pergunte a 3 antes de mim – Programação em Pares - Cada semana eu modelo uma falha e mostro aos meus alunos como persistir através da resolução de um problema, eu também tenho "paramédicos" na minha classe. Estes alunos são alunos que muitas vezes terminaram a sua tarefa mais cedo do que todos os outros. Eles andam em torno da classe e ajudam os outros quando eu sou incapaz de chegar a cada estudante⁶. (T2-P132).

Essas estratégias aparecem em diversas postagens como estratégias do professor para promover a interação e a aprendizagem. O “Pergunte a três”, por exemplo, incentiva os alunos a não buscarem a resposta direto com o professor e promove a interação na sala de aula, além, é claro, de liberar o professor para que ele dê atenção à toda a turma.

É importante ressaltar que o fato de simplesmente perguntar para o colega não é suficiente; é importante que os alunos também não deem respostas, incentivando os colegas a pensarem. É o que é evidenciado no extrato a seguir:

Eu também tenho usado o "pedir a 3 e, em seguida, pergunte-me" estratégia na minha sala de aula. Eu também digo aos meus alunos para usar a estratégia "tip-tip-tell". Eu treino meus alunos para não apenas dizer a seus colegas a resposta porque isso não vai ajudá-los a aprender nada. Em vez disso, encorajo os alunos a apenas dar aos seus colegas uma dica para encontrar a resposta. Se derem duas dicas e os alunos ainda não entenderem como chegar à resposta, então eles são autorizados a dizer a seus colegas. Beneficia ambos os alunos a pensar criticamente. (T2-P137).

A partir do extrato supracitado, percebemos que há evidências de que o professor, neste caso, incentiva que os alunos atuem de forma instigadora, criando na turma um ambiente realmente propício à construção do conhecimento.

⁵ In my classroom, I use "Ask 3 then me". By doing this, students are able to talk it out with each other and try to find solutions together. I also have students who are already programming and they email their work to staff members and admin to look at. They also have other students playing their games. This gives them such a great boost of confidence and shows that everyone is good at something.

⁶ This past year I used the following: - Ask 3 Before Me- Pair Programming
- Each week I modeled failure and showed my students how to persist through a problem to solve it- I also have "paramedics" in my class. These students are students who often finished their assignment earlier than everyone else. They walk around the class and help others when I am unable to get to each and every student.

Dentro da perspectiva da interação dos alunos, convém salientar a importante distinção entre cooperação e coação. Para Piaget (1998), a cooperação é o mais elevado nível de interação, pois necessita de descentração e reciprocidade, necessitando de uma relação entre iguais para acontecer. Piaget, nesse sentido, distingue a cooperação da coação; ambas estão relacionadas com a moral, porém a primeira é um processo de troca entre iguais, enquanto a segunda é uma relação heterônoma, ou seja, de autoridade, de prestígio. Os extratos extraídos do fórum não nos permitem identificar que tipo de relações estão expressas na troca com o outro nesse sentido, o que podemos perceber é uma proposta de interação e trocas proporcionada pelo professor.

Em diversos planos de aula se faz menção à colaboração. De fato, não sabemos se a ideia é que os alunos possam agir de forma colaborativa ou cooperativa. Na literatura, há divergências em relação à definição conceitual desses dois termos. Para Dillenbourg (1996), cooperar consiste em uma tarefa em que cada parceiro do grupo é responsável por uma parte. Assim, para compor o todo cada parte é agrupada, sem que, necessariamente, os outros parceiros do grupo tenham algum envolvimento com a parte do outro. Colaborar, para o autor, é um processo que vai além. Em uma tarefa em grupo todos os indivíduos fazem parte de cada ação, tornando o produto final uma produção em que todos agiram conjuntamente para a realização da tarefa. Em Piaget (1998), observamos que o conceito da cooperação se aproxima do conceito de Colaboração de Dillenburg. Assim, nesta dissertação nos convém utilizar o conceito de cooperação a partir da ótica Piagetiana.

As falas a seguir demonstram como as atividades em conjunto com os colegas são usadas pelos professores como recursos promotores da aprendizagem:

[...] Colaboração - os alunos descobrem muito mais ao trabalhar em parceria. Eles veem coisas que nem sempre veriam sozinhas⁷ (T1-P161).

Eu usei programação em pares, bem como perguntar a 3 então a mim. Eu também tive alunos mostrando suas criações com pelo menos outro colega⁸ (T2-P134).

A colaboração cria confiança em meus alunos. Ajuda-os a ver quanto mais eles podem realizar se eles compartilham abertamente seus conhecimentos, habilidades, ideias e

⁷ [...] Collaboration -students discover so much more when working in partnership. they see things they would not always see alone.

⁸ I have used paired programming as well as ask 3 then me. I have also had students show off their creations with at least one other classmate.

lutas uns com os outros. Aprender é "menos assustador" particularmente quando se trata de aprendizagem CS.⁹ (T2-P164).

Não sabemos se os professores relatam o uso de atividades colaborativas ou cooperativas por terem participado de formações que privilegiassem este tipo de estratégia pedagógica ou se, por outro lado, os professores tentam aplicar as orientações dos planos de aula da plataforma. Há uma diferença entre seguir uma orientação e viver uma experiência de aprendizagem. Se o professor vive essa experiência de interação e cooperação em seu processo de formação e reflete sobre a mesma, é possível que possa realizar intervenções mais atentas ao processo dos estudantes. De qualquer maneira, é sabido que a aprendizagem é potencializada à medida que a criança interage com o outro.

Piaget (1978) afirma que a aprendizagem ocorre por meio de desequilíbrio e equilíbrio. O processo de troca com o outro pode possibilitar esse movimento, uma vez que permite que o outro questione e problematize questões que poderiam não ser levantadas pelo sujeito sem a troca com o outro. O extrato a seguir segue a ideia de atividades que contemplem a interação com o outro.

Eu serei muito específico em minhas instruções e o que é esperado dos alunos. Eu vou pedir-lhes para tentar resolver por sua conta primeiro, em seguida, perguntar a alguém ao lado e, finalmente, a mim. Quando eles chegarem a mim, eles deverão ser capazes de me dizer o que eles fizeram para tentar resolver o problema por conta própria ¹⁰ (T2-P130).

Além da cooperação ou colaboração, essas estratégias também podem possibilitar a construção da autonomia dos alunos, uma vez que os provoca a testarem suas hipóteses e encoraja a construção conjunta de conhecimentos. Além disso, pode contribuir para o desenvolvimento da oralidade e da tomada de consciência, já que precisam explicitar ao professor o que já experimentaram fazer para a resolução de sua problemática. Wadsworth (1996) acrescenta que esse tipo de atividade permite ao professor criar um conflito cognitivo que levará o sujeito a pensar sobre suas hipóteses. Nas palavras do autor:

O propósito das explorações críticas é determinar que construções (regras e generalizações) um aluno dispõe em relação ao conteúdo em discussão. O professor

⁹ Collaboration builds confidence in my students. It helps them see how much more they can accomplish if they openly share their knowledge, skills, ideas and struggles with each other. Learning is "less scary" particularly when it comes to CS learning.

¹⁰ I'll be very specific in my instructions and what is expected of students. I'll ask them to try to solve on their own first, then ask someone next to them and finally me. When they get to me they should be able to tell me what they did to try to solve the problem on their own.

pode, então, propor questões destinadas a conflitar com o raciocínio subjacente às construções infantis. (WADSWORTH, 1996, p. 152).

Ou seja, o uso de atividades que privilegiem a interação, seja ela denominada como cooperação ou colaboração, permite que a aprendizagem seja favorecida por meio das trocas e reflexões possibilitadas por esse tipo de ação.

5.3.2 Atividades *off-line*

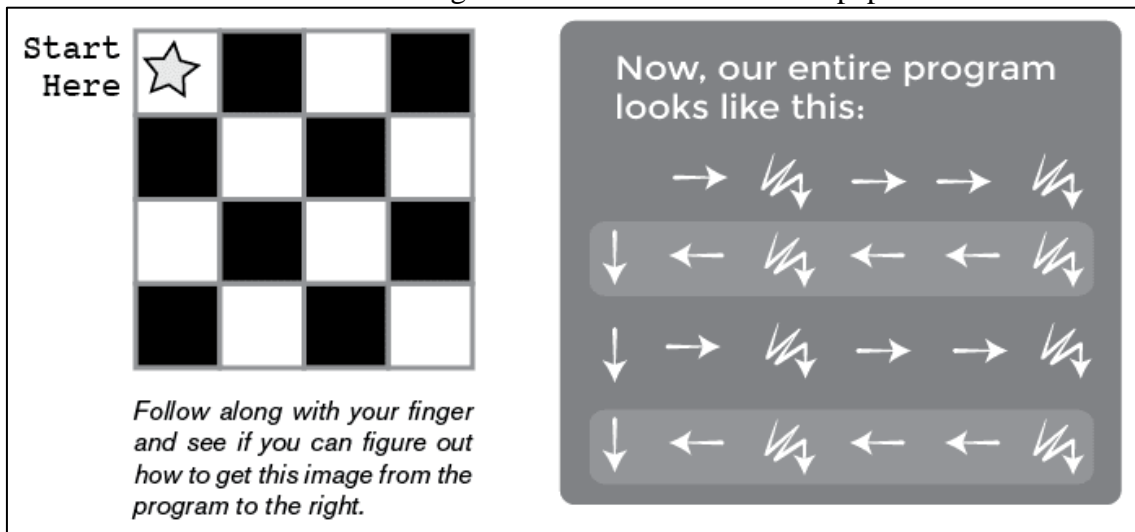
As atividades *off-line* ou, *unplugged*, estão bastante presentes nos planos de aula e são referidas por professores no fórum. O termo “*unplugged*” é proposto por Bell, Witten e Fellows (2011), em que os autores apresentam alternativas para o ensino do PC sem o uso do computador, possibilitando que o Pensamento Computacional seja desenvolvido mesmo em locais ou escolas que não dispõem de computadores, rede ou *internet*. Além disso, essas atividades permitem o entendimento dos alunos a partir das relações concretas com os conceitos de computação.

Boa parte dos planos de aula inicia a partir de uma atividade sem o uso do computador. O Plano de Aula 1 (P1), por exemplo, inicia a introdução aos conceitos de programação propondo uma atividade em que os alunos fornecerão instruções para que os colegas possam fazer desenhos. É uma espécie de algoritmo do mundo real em que os alunos interpretam comandos e desenvolvem ações. O trecho a seguir foi extraído do Plano 1.

Ao "programar" um ao outro para desenhar figuras, os alunos começarão a entender o que é realmente programação. A aula começará com alunos instruindo uns aos outros a colorir quadrados em papel milimetrado em um esforço para reproduzir uma imagem existente. Se houver tempo, a lição pode concluir com imagens que os alunos criam (P1).

A Figura 24 demonstra os princípios desta atividade. A partir dos símbolos (quadro da direita) desenhados, as ações são executadas pelos alunos na imagem à esquerda.

Figura 24 - Atividade aviões de papel



Fonte: P2.

O plano de aula 2 (P2) segue essa sistemática, propondo uma atividade com aviões de papel e construção de algoritmos da vida real, tendo como objetivo descrever as atividades de um dia e sistematizá-las por meio do uso do conceito de algoritmos. Exemplo:

Nesta lição, os alunos relacionarão o conceito de algoritmos às atividades cotidianas da vida real, fabricando aviões de papel. O objetivo aqui é começar a construir as habilidades para traduzir situações do mundo real para cenários *on-line* e vice-versa. (P2).

Apresente aos alunos a ideia de que é possível criar algoritmos para as coisas que fazemos todos os dias. Dê-lhes alguns exemplos, como fazer café da manhã, escovar os dentes e plantar uma flor (P2).

Algumas sugestões de intervenção, como a presente na aula A5, instigam o professor a utilizar atividades desconectadas como recurso para o entendimento de inúmeros conceitos da Ciência da Computação, como é o caso dos laços de repetição presentes no extrato a seguir.

Peça ao seu voluntário para caminhar ao redor da mesa (ou sua cadeira, ou um amigo). Quando terminar, instrua-os a fazê-lo de novo, usando exatamente as mesmas palavras que você fez antes. Quando terminarem, instrua novamente. Então de novo. Teria sido mais fácil para mim pedir-lhe que voltasse à mesa quatro vezes? (P5).

Exemplos como esses podem ser encontrados em inúmeros planos de aula. Assim, a partir da análise detalhada dos planos de aula, podemos concluir que há uma valorização especial às atividades desplugadas, pois as mesmas atuam como recursos que fornecem maiores

possibilidades para a compreensão dos conceitos propostos em cada curso. Nesse sentido, podemos ver mais afirmações, como as seguintes:

Espero trazer mais das atividades de aprendizado desconectadas e integrar uma abordagem mais interdisciplinar para que mais possam estar motivados para levar CS em sua sala de aula de forma mais consistente ¹¹ (T1-P7).

Atividades desconectadas - às vezes, precisamos estar longe do dispositivo para pensar em um problema e relacionar nossas descobertas de volta e aplicar ao dispositivo ¹² (T1-P61).

BRACKMANN (2016) salienta que a aprendizagem por meio de atividades desplugadas permite que os alunos aprendam a partir do concreto, sendo este um princípio do construcionismo. Nas palavras do autor:

As atividades desplugadas ocorrem frequentemente através da aprendizagem cenestésica (e.g. movimentar-se, usar cartões recortar, dobrar, colar, desenhar, pintar resolver enigmas, etc.) e os estudantes trabalham entre si para aprender conceitos da Computação. Trabalhar com objetos tangíveis do mundo real é um princípio central do construcionismo de Papert [...] (BRACKMANN, 2016, p. 50).

Nessa perspectiva, também convém salientar a importância das intervenções do professor nas atividades propostas, já que a sistematização do professor é fundamental para criar possibilidades para os alunos construir conhecimento, uma vez que o professor é capaz de criar situações que possibilitem processos de desequilíbrio e equilíbrio. Desse modo, “[...] a criança passa a ser desafiada a agir sobre diferentes objetos, e os experimentos são criados de modo que a fala do pesquisador seja cada vez menos presente. O experimentador modifica (complica) a situação no sentido de provocar as suas hipóteses [...] (SANTOS, 2007, p. 36).”.

Com o uso das atividades desconectadas, o professor pode, a partir do concreto, criar situações problema e fazer intervenções significativas, buscando que o aluno seja capaz de construir conceitos sobre o Pensamento Computacional. Essas intervenções podem seguir uma perspectiva de pesquisa em que o professor é também pesquisador, argumenta Kebach (2007):

[...] primeiro, porque o pesquisador deve seguir o pensamento da criança procurando compreender sua lógica, exigindo assim um esforço de descentramento por parte do adulto de não sugerir a criança, não procurar que ela responda aquilo que se

¹¹ I hope to bring more of the unplugged learning activities and integrating a more cross-curricular approach so that more can be motivated to bring CS into their classroom in a more consistent way.

¹² Unplugged activities- sometimes we need to be away from the device to think through a problem and relate our findings back to and apply to device.

deseja, proporcionado a reflexão a partir das proposições da problemática (KEBACH, 2007, p. 45).

Segundo, porque é preciso formular provas referentes ao assunto que devem ser trabalhados (pesquisado) a partir de hipóteses predeterminadas e criar meios de aplicá-las, em função do pensamento espontâneo da criança em relação ao assunto pesquisado (KEBACH, 2007, p. 45).

Dentro dessa perspectiva, a intervenção do professor se dá no intuito de buscar as hipóteses do aluno, e não simplesmente para que ele responda ao que o professor está questionando.

Nesse sentido, as atividades desplugadas contribuem para, ao lado da intervenção do professor, provocar o aluno a conhecer e construir novas hipóteses acerca do mundo da Ciência da Computação.

5.3.3 Atividades *on-line*

Também foi possível observar a emergência da categoria “Atividades *On-line*”, que consiste nas atividades *on-line* disponíveis na plataforma *Code.Org* e que são mencionadas nos planos de aula. O plano de aula 3 (P3) propõe que os alunos utilizem os tutoriais da Aula 3 (A3), referente ao fornecimento de instruções com o objetivo de levar o personagem “*Angry Bird*” até o porco. A Figura 25 exemplifica a atividade.

Figura 25 - Labirinto e sequência



Fonte: A3-F1.

Nesse tipo de atividade, a criança arrasta blocos de comandos para a direita, com o intuito de programar os personagens na tela para realizar uma ação solicitada. Como as atividades nos planos anteriores propuseram o uso de programação desplugada, agora o aluno poderá testar suas habilidades utilizando a atividade *on-line*.

Enquanto seus alunos trabalham nos quebra-cabeças, observe como eles planejam o caminho para o pássaro. Identifique as diferentes estratégias utilizadas e peça aos alunos para compartilhar com toda a turma. Isso ajuda os alunos a reconhecer que há muitas maneiras de abordar esses problemas. Você pode querer passar por alguns quebra-cabeças no projetor. Ao fazer isso, você pode pedir a um aluno que trace o caminho na tela enquanto outro escreve as instruções em um quadro branco (P3).

Apesar de a estratégia proposta por esse plano de aula ser pautada no uso da atividade A3(*on-line*) no *Code.Org*, o plano de aula fornece instruções para que o professor continue utilizando atividades do mundo concreto e, além disso, incentiva o trabalho em grupo, como esta:

Em pequenos grupos, permita que os alunos projetem seus próprios labirintos e desafiem uns aos outros a escrever programas para resolvê-los. Para maior diversão, faça labirintos em tamanho natural com os alunos como o porco e o pássaro. (P3).

A partir desses conceitos, trabalhados nas aulas anteriores, o plano de aula 5 traz uma proposta de aula que faz uso da atividade *on-line* na qual é necessário usar blocos de repetição para solucionar um problema proposto. A seguir, a figura 26 mostra a primeira problemática sugerida na atividade, demonstrando uma quantidade desnecessária de comandos *avance*. E, em seguida, a figura 27 propõe que esses comandos sejam substituídos por uma estrutura de repetição.

Figura 26 - Labirinto e Sequência 2



Fonte: A6-F1

Figura 27- Laços de Repetição



Fonte: A6-F2

Essas atividades seguem uma sequência a partir dos conceitos trabalhados nas aulas anteriores para que os alunos possam, a partir das experiências concretas, compreender o conceito e utilização das estruturas de repetição. Essa sequência pode ser percebida ao se analisar a sequência das atividades propostas pelos planos de aula. No Plano de Aula 7 (P7), o professor é orientado a utilizar a Atividade 7 (A7) para fazer os alunos desenharem na tela e, com isso, entender comandos mais específicos de repetição, a saber:

Conte o número de vezes que uma ação deve ser repetida e represente-a como um *loop*; Decomponha uma forma na sua menor sequência repetitiva; Crie um programa que desenhe formas complexas repetindo sequências simples. (P7).

A partir da análise das atividades *on-line* na categoria estratégias, evidenciamos que elas são um recurso para o desenvolvimento do pensamento computacional, e é a principal forma disponível na plataforma, já que essas atividades são a base dos cursos propostos e, apesar dos planos de aula intercalarem atividades desplugadas e atividades *on-line*, essas atividades são as que são principalmente realizadas pelos usuários da plataforma. Essa afirmação se sustenta uma vez que os cursos são abertos e inúmeros usuários realizam essas atividades todos os dias, sem que necessariamente exista a intervenção de um professor.

As atividades *on-line* aparecem como um recurso importante para o aprendizado e desenvolvimento do pensamento computacional, pois permitem um *feedback* instantâneo quando o sujeito realiza a atividade. Nesse sentido, as atividades *on-line* e as atividades *off-line* aparecem como estratégias complementares para serem usadas pelos professores como recursos promotores da aprendizagem.

5.4 DIFICULTADORES DA PRÁTICA COM O PC

Dentro das categorias descritas no capítulo anterior, foram considerados como Dificultadores da Prática com o Pensamento Computacional os relatos de professores que apresentaram dificuldades e/ou percalços na atuação com o Pensamento Computacional.

Essa categoria é dividida em três subcategorias: Tempo, Condições Estruturais e Dificuldades com TI. Na subcategoria Tempo, de acordo com a análise, há indícios de que o tempo de aula é breve e configura uma restrição para a implementação de um currículo que contemple os conceitos de Ciência da Computação. Destacamos aqui os seguintes trechos:

O meu maior desafio será o tempo. Eu só vejo meus alunos 1x por semana por 40 minutos e às vezes as coisas são interrompidas com as atividades da escola, ou seja, atrasos, assembleias, viagens de campo, em serviço.¹³ (T1-P83).

As questões do tempo sempre são um obstáculo para mim, já que eu também sou obrigado a ensinar outros tópicos de laboratório no meu laboratório.¹⁴ (T1-P2).

Vejo restrições de tempo como o meu maior desafio. Eu quero fazer tudo no currículo, mas com aulas de 50 minutos que realizo em cerca de 40 minutos, eu me pergunto como vou realizar isso. [...] ¹⁵ (T1-P67).

As falas indicam que há dificuldades de tempo, uma vez que os horários das aulas de informática são geralmente restritos a quarenta ou cinquenta minutos. Os planos de aula também trazem menções que dão indícios de que o tempo pode ser um empecilho no desenvolvimento das atividades da *Code.org*. A transcrição literal a seguir exemplifica esta questão:

O tempo básico da lição inclui apenas atividade. As sugestões introdutórias e de encerramento podem ser utilizadas para aprofundar a leitura quando o tempo o permitir (P5).

Além disso, os professores precisam desenvolver outras atividades comuns do currículo de informática e relatam que não há tempo para as atividades de pensamento computacional. Também há evidências de que, em alguns casos, os alunos ainda possuem outras

¹³ My biggest challenge is going to be time. I only see my students 1x per week for 40 minutes and sometimes things are interrupted with the school activities i.e. delays, assemblies, field trips, in-service.

¹⁴ Time issues is always a hurdle for me since I am also required to teach other lab topics in my lab.

¹⁵ I see time constraints as my biggest challenge. I want to do everything in the curriculum, but with 50 minute classes that realize into about 40 minutes, I wonder how I will accomplish it [...].

atividades que podem ser realizadas durante a aula de informática, o que demanda ainda mais prejuízos para o tempo limitado. Relatou o professor:

Só os vejo uma vez por semana durante 30 minutos e eles também precisam pegar livros naquele tempo [...] ¹⁶ (T1- P82).

Nesse sentido, quando os professores falam do tempo da disciplina, é possível inferir que a informática ou os conceitos de PC aparecem como outra disciplina trabalhada isoladamente de outros componentes curriculares, e não articulado com o currículo, conforme propunha Papert (1980).

Com relação ao tempo, podemos retomar aqui uma discussão epistemológica. Em primeiro lugar, a escola ou o sistema educacional, em geral, se organiza em uma estrutura curricular compartimentada, que separa o conhecimento em disciplinas e inclui mais subdivisões ou disciplinas de pouco tempo e que não contempla o mínimo necessário para o desenvolvimento das atividades. Em segundo, se a proposta é o desenvolvimento da aprendizagem e do pensamento computacional, é possível pensar um conteúdo a ser vencido e com limite de tempo para essa construção? Os extratos mostram que os professores precisam usar o tempo de uma disciplina já existente, o que é ainda pior. Essas falas vão à contramão da proposta de Papert (1980) que, justamente, criticava a inclusão da programação em uma perspectiva instrucionista.

Em sequência à análise proposta, na subcategoria “Condições Estruturais”, encontramos depoimentos que relatam desde dificuldades com o número de alunos, *hardware* ou *software* disponível para todos até largura da banda. Assim, destacamos a seguinte fala:

Já tenho e continuarei a ver desafios com a largura de banda e não tenho certeza do que posso fazer para superar esse problema ¹⁷ (T1-P87).

Tendo em vista que as atividades do *Code.Org* são, em sua maioria, *on-line*, a implementação das atividades de pensamento computacional com a plataforma em questão pode ser inviabilizada. Convém salientar, nesse sentido, um estudo de Leite et. al (2017) em que os autores argumentam que escolas brasileiras também carecem de equipamentos e de acesso à *internet* que possibilitem uma prática pedagógica com o pensamento computacional,

¹⁶I only see them once a week for 30 minutes and they also have to get books in that time [...].

¹⁷ I have already and will continue to see challenges with bandwidth and I am unsure what I can do to overcome this issue.

mas esclarecem que a formação de professores é ainda mais crítica e urgente no contexto nacional.

Na sequência, ainda há relatos que indicam número excessivo de alunos, como a apresentada por este professor:

Eu organizei apenas uma turma para um clube, então 27 poderá ser um desafio¹⁸ (T2-P126).

Essas evidências aparecem com maior frequência na categoria “Aspectos promotores da aprendizagem”, uma vez que os professores apresentam relatos de sucesso para vencer esta dificuldade como o ‘*Pergunte a três antes de mim*’.

A última subcategoria deste grupo apresenta as “Dificuldades com TI”, que englobam uma série de problemas relativos às dificuldades que professores apresentam com o aprendizado e ensino de programação e conceitos de Ciência da Computação, como os apresentados a seguir:

Estou um pouco preocupado com o fato de que muitas crianças foram expostas à programação nos últimos dois anos em suas aulas de arte/tecnologia para que eles possam saber mais do que eu [...] ¹⁹ (T1-P35).

Esta é a minha primeira vez e estou lutando para entender. [...]. É difícil ficar animado com algo que estou lutando comigo mesmo ²⁰ (T1-P49).

Este será o meu primeiro ano usando *Code.org* na minha sala de aula. [...]. Eu vou fazer a ciência da computação em álgebra com a minha classe. Eu estou um pouco nervoso. Esta primeira corrida será interessante, mas acho que meus alunos me ajudarão com isso ²¹ (T1-P68).

O meu maior desafio é superar o fato de que vou aprender com eles e não necessariamente ser o especialista ²² (T1-P58).

As falas dos professores evidenciam, muitas vezes, despreparo e insegurança ou, ainda, resistência para o trabalho com as atividades de programação e pensamento computacional em sala de aula. Observamos que alguns professores se encontram descontentes

¹⁸ I have only set up a class for a club, so 27 might be challenging.

¹⁹ I am a tad concerned about the fact that many kids have been exposed to coding the last two years in their art/tech classes so they may know more than me [...].

²⁰ This is my first time and I am struggling to understand. [...]. It's hard to be excited over something that I am struggling with myself.

²¹ This will be my first year using Code.org in my classroom. [...]. I am going to do the Computer Science in Algebra with my class. I am a little nervous. This first run will be interesting, but I think my students will help me through it.

²² My biggest challenge is overcoming the fact that I will be learning it right along with them, and not necessarily be the expert.

com as novas demandas do ensino (T1-P49) e outros que não se sentem confiantes para a atuação docente, uma vez que possuem pouco conhecimento das ferramentas utilizadas (T1-P35).

Entendemos que os processos de formação pelos quais esses professores passaram nas oficinas promovidas pela comunidade *Code.org* não são suficientes para o preparo dos professores para atuar com o ensino do currículo proposto pela CSTA (2017). De todo modo, as discussões levantadas até aqui nos permitem esboçar um cenário de introdução do pensamento computacional a partir da plataforma *Code.org* e, assim, traçar alguns indicadores para que se possa pensar a formação de professores em um contexto de introdução do pensamento computacional.

Neste texto, trouxemos as dificuldades relatadas pelos professores a partir da aplicação da proposta presente na *Code.org*. Contudo, esse levantamento também nos permite buscar relações com os aspectos dificultadores da inclusão e da prática pedagógica com o pensamento computacional em qualquer outro contexto da vida real. É o caso das questões de tempo que, pelas entrelinhas, podemos notar que o problema vai além do próprio período e caracteriza um problema do sistema educacional como um todo, disciplinarizado e sem espaço para questões transversais.

5.5 INDICADORES PARA A DOCÊNCIA COM O PC

O referencial teórico e a análise apresentada até aqui nos possibilitam pensar aspectos que consideramos essenciais para que o professor possa atuar como mediador do desenvolvimento do pensamento computacional. No início desta dissertação, traçamos o percurso do conceito de pensamento computacional, buscando uma relação entre o conceito proposto por Jeannet Wing (2006) e o conceito proposto por Seymour Papert (1980), nos anos 1980. Essa trajetória, inclusive, permite-nos entender que o pensamento computacional não é um conceito recente, mas um constructo teórico importante e já experimentado na educação nos últimos anos.

A maneira como Papert (1980, 1985, 2006) pensou a inclusão das linguagens de programação na educação e a sua teoria, o Construcionismo, nos é útil para pensar que características, ou que indicadores, são importantes para traçar pressupostos para a formação de professores. Papert, certamente influenciado por Piaget, construiu uma maneira de se pensar a educação e a informática na educação que se afastou da simples instrução e da transmissão de

saberes. Dessa forma, compreendemos que o professor, nesse contexto, precisa entender, em primeiro lugar, **que aprender é mais importante que ensinar**.

Não foi por acaso que iniciamos esta dissertação com a frase “Tudo que a gente ensina a uma criança, a criança não pode mais, ela mesma, descobrir ou inventar”. Concordamos com Piaget, no sentido de que, antes de apenas ensinar, é preciso, sobretudo, criar possibilidades, desafios, inquietações para que o sujeito seja capaz de construir o seu próprio conhecimento. Em nossa visão, ambientes como o *Scratch*, por exemplo, estão diretamente embasados em uma linha teórica sustentada pela filosofia LOGO e, nesse sentido, tanto o *Code.Org*, como o *Scratch*, *Tynker*, os Brinquedos Lego e outros tantos são a materialização das ideias de Papert (1980) quando ele diz: “Vejo-a como instrumento educacional válido, mas sua principal função é servir como modelo para outros objetos ainda a serem inventados.” (PAPERT, 1988, p. 26). Conhecemos um grande teórico quando suas ideias se tornam atemporais.

O LOGO, se pensado apenas como uma linguagem de programação é, sobretudo, obsoleto. Por outro lado, ao olharmos para a Filosofia LOGO como propôs Papert (1980), percebemos que essa maneira de pensar está viva em inúmeros ambientes e recursos que hoje são utilizados para o desenvolvimento do pensamento computacional. Ora, se o LOGO se mantém nesses recursos e a filosofia LOGO foi baseada em princípios construtivistas, não é possível que distanciemos o professor também dessa mesma base teórica.

Assim, o que significa dizer que aprender é mais importante que ensinar? Significa que o professor precisa entender que o centro do processo de aprendizagem é o aluno e que ele não terá a função de apenas apresentar conteúdos que serão executados e replicados pelos alunos. É preciso, nesse sentido, vencer a ideia de depósitos de conhecimentos que serão, em algum momento da vida, extraídos e postos em prática (FREIRE, 1996).

Na análise, discutimos que as atividades da *Code.org*, por exemplo, privilegiam pouco a criação ou, ainda, permitem que o aluno crie apenas no final do curso. E se o professor pudesse dar o espaço de criação logo no início da aula? E se o professor criasse questionamentos e atividades que propiciasse a experimentação e a criação antes de apresentar conceitos? As atividades da *Code.Org* apresentam uma limitação nesse sentido, mas outras plataformas permitem a livre criação e a intervenção do professor no sentido de estimular a criatividade e a exploração. É o caso do *Scratch*, por exemplo, que permite ao sujeito explorar, brincar, desenhar, criar e aprender se divertindo, sem que antes tenha que necessariamente passar por uma série de fases pré-determinadas. Os recursos citados no Capítulo 3 desta dissertação podem dar apoio para o professor criar estratégias a fim de que os alunos desenvolvam o pensamento computacional.

Um professor que privilegia o ensino não permite que um aluno acesse um ambiente como o *Scratch* e possa explorar a ferramenta e descobrir coisas novas. Ele certamente irá apresentar a interface da ferramenta e as inúmeras possibilidades de criação, comandos e o máximo de recursos possíveis. Depois de apresentar o máximo de recursos, o professor irá realizar uma atividade guiada, pois ele entende que os alunos só poderão desenvolver alguma coisa nova depois que já estiverem devidamente apresentados à plataforma. Esse professor certamente se sente muito confortável nas atividades do *Code.org*, pois elas são exatamente como ele precisa. Apresentam cada atividade e conduzem o sujeito por um percurso já conhecido. Se, pelo contrário, o sujeito se entedia e tentar criar algo novo, certamente esse professor solicitará que ele retorne para a atividade proposta, sem espaços para coisas novas ou atividades que não foram previstas para a aula. Como ele entende que o conhecimento dele é transmitido para o aluno, ele precisa também checar se esse conhecimento foi devidamente “absorvido” e, para isso, há necessidade de uma avaliação formal e individual. É o que analisamos no capítulo anterior, quando os planos de aula sugerem que o professor distribua a planilha de avaliação que deverá ser resolvida individualmente. O que acabamos de descrever se assemelha muito com o modelo de professor empirista trazido por Becker (1993, 1999).

Ao contrário de tudo isso, um professor que privilegia a aprendizagem vai se distanciar de qualquer recurso que limite a criação de seus alunos. Quando se privilegia a aprendizagem, o professor, em uma plataforma como o *Code.Org*, por exemplo, iniciará as atividades por aquelas que permitem livre criação. Solicitará que os alunos explorem a interface proposta e contem para a turma o que encontraram de novo, a partir daí criará algumas problemáticas e deixará que os alunos busquem resolvê-las e conhecer o máximo de cada recurso. Em ambientes como *Scratch* e *Tynker*, esse professor fará uma primeira atividade de livre exploração, em que os alunos terão que tentar descobrir novas coisas a partir da própria curiosidade. Ele não vai explicar ferramenta por ferramenta, item por item, porque ele entende que os alunos já trazem uma bagagem e que a interface do *software* lhes permitirá fazer relações. Assim, esse professor não trabalhará com atividades totalmente guiadas, porque ele entende que as atividades guiadas limitam a criatividade e a capacidade de construção do conhecimento. Ele não fará avaliações no final, mas entenderá que todo o processo de criação e socialização do conhecimento com o professor e com outros alunos faz parte de um processo de avaliação constante, que tem como fim possibilitar o construir e o reconstruir dos conhecimentos. Nesse sentido, ele pode eleger outros inúmeros instrumentos de avaliação, como os trabalhos realizados, um diário de bordo ou, ainda, as interações e trocas realizadas pelos alunos na sala de aula.

Ainda nesse contexto, outro indicador para a docência com o pensamento computacional é o **professor ser mediador do processo de aprendizagem**. Ou seja, se o professor compreende que o seu papel não é o de ser transmissor, que postura ele precisa tomar em sua prática pedagógica? Fagundes (1999) e Valente (1999) argumentam que o professor no contexto das tecnologias digitais precisa atuar como mediador dos processos de aprendizagem, pois, caso contrário, ele seria facilmente substituído pelas tecnologias da informação e comunicação. Silva (2004) vai além ao dizer que o professor, nesse contexto, constitui-se como um sistematizador de experiências. Nesse sentido, o professor que atua com o ensino e a aprendizagem do pensamento computacional também pode se valer dessa postura, uma vez que ele opera como alguém que provoca, orienta ou sistematiza experiências para que os alunos possam desenvolver as competências do pensamento computacional.

Por que essa postura é importante? Tomemos por base o professor que, ao ensinar, apresente conceitos no quadro, em seguida ele solicita que os alunos realizem exercícios e segue essa sistemática aula a aula. Ao ser questionado, ele dá as respostas ou pede para o aluno pensar, sem fazer nenhum tipo de intervenção. Nesse exemplo, os alunos que irão aprender são aqueles que a explicação ou os exercícios do professor possibilitaram algum processo de desequilíbrio e construção de novas equilibrações. Aqueles que, do contrário, não conseguiram atingir esse estágio ficarão com maiores dificuldades. Se, por outro lado, o professor trabalhar com um problema e a partir dele permitir que os alunos pensem sobre as situações e ainda atuar individualmente buscando compreender e questionar as ações dos alunos, certamente esse professor estará atuando como um mediador, pois em vez de simplesmente “passar um conteúdo” e exercícios, ele irá buscar compreender a lógica do aluno e questioná-lo para levar cada um a um processo meta-cognitivo. Essa ação se aproxima, então, do Método Clínico piagetiano, pois a partir dos questionamentos o professor poderá buscar compreender o ponto de vista do aluno, suas hipóteses e as estratégias que ele propõe para a resolução de cada problema apresentado em sala de aula.

[...] Piaget reconhece que, para avançar a sua investigação sobre a lógica da criança, deve centrar o interrogatório em diferentes objetos que esta possa manipular. Assim a criança passa a ser desafiada a agir sobre diferentes objetos, e os experimentos são criados de modo que a fala do pesquisador seja cada vez menos presente (SANTOS, 2007, p. 36).

A partir disso, entendemos que a intervenção do professor, não apenas para o ensino de crianças, mas em qualquer contexto, aproxima-se de uma perspectiva de pesquisa clínica.

Ou seja, significa que o professor investiga e questiona, mobilizando cognitivamente o sujeito para que construa o conhecimento a partir dos desequilíbrios criados pelo professor mediador.

Esse movimento vai muito além da prática tradicional de sala de aula e permite que os alunos realmente construam o conhecimento ao invés de simplesmente captar informações e soluções algorítmicas para problemas vistos em sala de aula. A nossa experiência acerca do ensino de computação mostra que, muitas vezes, os alunos, apesar de terem resolvido diversas questões e algoritmos em sala de aula, não são capazes, em uma avaliação formal, de desenvolver o raciocínio computacional para a solução de um problema diferente daqueles vistos em aula. O ponto de vista de Kebach (2007) vem ao encontro dessa problemática:

[...] os docentes se perdem em situações delicadas e voltam a reproduzir quadros empiristas em suas aulas. Se não se verifica na prática (observação de dados empíricos) como o ser humano constrói conhecimento, não é possível proporcionar situações de progressão das aprendizagens dos alunos. Portanto, conhecer meios de desafiar os alunos por meio de uma postura dialético-didática, propor tarefas interessantes e significativas e observar suas ações, deduções e induções, experimentações, livres ações, trocas de ponto de vista, conflitos cognitivos, pesquisas coletivas, é necessário para que se possa partir de uma postura construtivista (KEBACH, 2007, p. 47).

Ou seja, o professor mediador ou sistematizador se aproxima de uma epistemologia construtivista e, nesse caso, entende que o conhecimento não está nele e, portanto, sabe que sua função no processo de aprendizagem é a de criar condições para que o conhecimento seja construído, e não simplesmente a de transmitir informações.

Um professor mediador também compreende que a aprendizagem é um processo que valoriza a interação social, por esse motivo, elencamos outro importante indicador para a docência como pensamento computacional, sendo ele **entender que aprender com o outro é fundamental**. O professor que compreende que o processo de interação com os colegas é fundamental cria estratégias que permitem interações e trocas entre os diversos alunos da turma. Nesta dissertação, por exemplo, analisamos as falas de professores que utilizaram o “Pergunte a três” para que os alunos pudessem interagir, questionar o colega e, conseqüentemente, aprender. Wadsworth (1996) acrescenta que a interação social é fundamental nos processos de aprendizagem:

A interação entre colegas torna-se particularmente importante para o desenvolvimento cognitivo a partir do momento em que a criança passa a ser capaz de assimilar os pontos de vista dos outros e quanto eles se mostram diferentes dos seus próprios pontos de vista. (WADSWORTH, 1996, p. 153).

Nesse sentido, propor atividades em grupo e mediar esse trabalho permite que os sujeitos cooperem para desenvolver tarefas. Essa ação permite à criança, segundo Piaget (1998), descentrar-se, ou seja, colocar-se no ponto de vista do outro. É importante que o professor crie um ambiente propício para a aprendizagem com o outro e permita que a turma entenda também a importância de cada aluno ser também mediador. Ou seja, incentivar que os alunos, ao ajudarem o colega a desenvolver uma atividade, não apenas deem respostas, mas façam o colega pensar. Isso é importante, pois as aulas vão se reconfigurando e a visão de aprendizagem dos alunos também se modifica, pois eles começam a entender que ensinar não é apenas dar as respostas aos colegas. Essa estratégia aparece na análise quando o professor utiliza o “Pergunte a três”, por exemplo. Esse é um caminho difícil, mas que cria efetivas mudanças na configuração da sala de aula e nas relações dos alunos com o professor e fortalece a autonomia dos alunos.

É importante observar que, ao falarmos em cooperação no contexto da autonomia, situamos o indivíduo sempre em um grupo social no qual realizará as trocas necessárias à construção de suas hipóteses de conhecimento e, pelos conflitos surgidos nas trocas no grupo, às modificações posteriores dessas hipóteses (RANGEL, 1992, p. 69).

Ou seja, a troca com os indivíduos do grupo permite que o sujeito modifique suas hipóteses e ajude a modificar as hipóteses do outro, criando um ambiente propício de verdadeira construção do conhecimento. O professor que compreende esse ganho cria possibilidades e favorece a ação cooperativa dos alunos.

Além de utilizar estratégias que contemplem a cooperação, o professor poderá lançar mão de outras estratégias, tais como a escolha de recursos digitais e *unplugged* para o ensino e a aprendizagem. Por esse motivo, entendemos que outro indicador importante para a atuação docente consiste em **conhecer e utilizar diversos recursos (on-line e off-line) para auxiliar no processo de ensino e aprendizagem**. No decorrer desta dissertação, apontamos uma série de recursos, digitais e não digitais, que podem ser utilizados para o ensino e a aprendizagem do pensamento computacional. Dentre os recursos digitais, listamos o *Scratch*, o *AppInventor*, o *Code.org*, entre outros. Essas plataformas têm diversos pontos em comum. Em primeiro lugar, utilizam a programação em blocos; em segundo, são grandes recursos para o início da aprendizagem de programação. Do mesmo modo, as atividades *off-line* ou *unplugged* se constituem como um importante recurso para iniciar a aprendizagem de conceitos de programação, conforme Bell, Witten e Fellows (2011) e Backmann. (2016).

Cabe ao professor conhecer o máximo de recursos disponíveis e pensar estratégias que possibilitem aos alunos explorar cada uma dessas ferramentas e suas especificidades. Enquanto as atividades *upplugged* fornecem uma experiência a partir de recursos concretos, as atividades *on-line* possibilitam a imediata criação e testagem da programação. Por exemplo, o professor pode criar atividades simples que permitam ao aluno conhecer os primeiros comandos sequenciais de programação sem que, necessariamente, seja necessário usar o computador. Programar o colega, por exemplo, pode ser uma das atividades. O professor pode organizar a turma para que um grupo escreva comandos no quadro ou em uma folha de papel, enquanto outro grupo executa as instruções como se fosse um robô. Uma atividade como esta, além de ser divertida, permite que o aluno entenda que o computador (colega) irá executar apenas os comandos programados.

Em uma sequência dessa atividade, o professor poderá utilizar algum recurso *on-line*, as atividades do *Code.org*, por exemplo, para que o aluno possa experimentar a programação com blocos a partir dos conceitos que ele já desenvolveu com a atividade desplugada. Aliar esses dois mecanismos fornece ao professor uma riqueza de práticas que se embasa diretamente no Construcionismo (Papert, 2006).

Para o professor dispor desses conhecimentos, ele precisa investigar ao máximo os recursos e estratégias disponíveis e, por isso, assim como qualquer professor no contexto contemporâneo, precisa adquirir uma postura de pesquisador.

Os indicadores mencionados até aqui são necessários, não apenas para o professor que atue com o ensino e a aprendizagem do pensamento computacional, mas para o professor que atua no contexto das tecnologias digitais. Esses indicadores retratam também posturas e concepções epistemológicas que são úteis para o professor, de maneira geral, na prática docente. Contudo, essa última categoria traz uma especificidade, a necessidade de um professor **que conheça o conceito de pensamento computacional** para que seja capaz de propor estratégias que levem à aprendizagem do Pensamento Computacional. Ou seja, isso significa que não é possível, simplesmente, que qualquer professor possa assumir a responsabilidade de atuar como professor nesse contexto. Inclusive, tratamos nesta dissertação, no capítulo 3, a partir da revisão sistemática de literatura apresentada, que profissionais da área de TI não conhecem o conceito de PC. Nesse caso, apontamos como fundamental a necessidade de uma clareza conceitual para a prática pedagógica.

A necessidade de o professor ser profundo conhecedor de sua área de atuação vem ao encontro do que Kebach (2007), baseada no método clínico e na epistemologia genética piagetiana, postula: que o professor mediador precisa, previamente, tentar entender a

construção das hipóteses do sujeito que constrói conhecimento para que, dentro de uma perspectiva clínica, seja capaz de questionar e criar desequilíbrios que levem à aprendizagem. Ora, se o professor não conhecer de fato a área em que atua, poderia ele questionar e possibilitar os desequilíbrios necessários para a construção do conhecimento? Assim, se ele não dispuser desses conhecimentos, muitas vezes atuará apenas como um professor expositor de conteúdos, e não saberá privilegiar os conhecimentos e as hipóteses apresentadas pelos seus alunos, intervindo apenas em níveis muito superficiais de construção de conhecimento.

Conforme já discutido no capítulo 3, não há consenso na literatura sobre tudo o que, de fato, contempla o pensamento computacional. Tomemos por base a proposta de Brennan e Resnick (2012), em que o pensamento computacional se constitui por três grandes áreas, a saber: 1 - Conceitos Computacionais, 2 - Práticas Computacionais e 3 - Perspectivas computacionais. É claro que o professor precisa entender, de fato, como utilizar cada uma dessas grandes áreas em sua atuação docente. Sobretudo, compreender o pensamento computacional como uma linguagem que permite ao sujeito se expressar e se conectar com o outro como uma nova linguagem que emerge em um contexto global. Ou seja, o pensamento computacional materializado por meio das linguagens (seja ela computacional, matemática ou algébrica) toma uma importância que é transversal a todas as áreas do conhecimento como um novo idioma. Sob essa perspectiva, apontamos que o professor precisa, então, compreender essa dimensão e a sua importância na sociedade do século XXI.

Wing (2006) acrescenta que uma das habilidades mais importantes do pensamento computacional consiste na abstração. Segundo a autora, abstrair significa ter a capacidade de extrair apenas o que é necessário para a resolução de um problema e descartar os dados que não são necessários. Para a autora, a abstração está presente na escrita de algoritmos, na compreensão e organização de um sistema e em diversas situações que envolvem a resolução de um problema.

Outra habilidade, segundo Wing (2010), é a capacidade de decomposição. Ou seja, a redução de um problema grande e complexo em problemas menores e mais fáceis de serem resolvidos. A capacidade de decomposição pode ser desenvolvida a partir de uma situação problema em que o aluno precise reduzir um grande problema em problemas menores para que possa resolvê-los utilizando uma linguagem algorítmica. A partir da criação de diversas soluções, os alunos poderão reconhecer padrões e reutilizar soluções já criadas.

Então, para que o professor consiga atuar nesse sentido, ele precisará ser capaz, em primeiro lugar, de compreender, ao menos, os principais conceitos para criar estratégias para usar na prática pedagógica. Conhecer esse conceito se torna fundamental, não só para o

professor que seja responsável pela tecnologia na escola, mas é também imprescindível para o professor de outras áreas do conhecimento. Dentre as diversas menções acerca do pensamento computacional na Base Nacional Comum Curricular (BNCC), convém destacar a seguinte:

Outro aspecto a ser considerado é que a aprendizagem de Álgebra, como também aquelas relacionadas a outros campos da Matemática (Números, Geometria e Probabilidade e Estatística), podem 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 (BRASIL, 201, p. 269).

Ou seja, o pensamento computacional tem se configurado como uma competência que deve estar presente em todas as escolas e níveis de ensino no cenário global e, recentemente, reconhecido como uma competência fundamental também no cenário nacional. Portanto, evidenciamos a necessidade de o professor estar preparado para atuar e contribuir para o desenvolvimento do pensamento computacional na sala de aula.

Apontar alguns indicadores a partir do referencial teórico e dos resultados obtidos com esta dissertação nos permite pensar que competências são necessárias para o professor que atue com o pensamento computacional em sala de aula. Essas competências podem servir como um caminho inicial para que possamos refletir a respeito do que é necessário para a formação de professores nesse contexto.

6 CONSIDERAÇÕES FINAIS

O pensamento computacional tem sido discutido nos últimos anos em diversos setores da sociedade, na academia, no mundo do trabalho e na educação. Não se sabe ao certo todas as competências que giram em torno do tema, contudo, na literatura há um consenso acerca da importância do conceito para o cidadão do século XXI. Pensar computacionalmente permite que o sujeito seja capaz de resolver problemas aliando procedimentos da computação ao seu aparato cognitivo. Papert (1980, p. 187) argumenta que esse tipo de pensamento não diz respeito a pensar exclusivamente como máquina, mas sim adicionar à capacidade cognitiva humana mais um recurso para a resolução de problemas.

Ao trazermos o percurso do conceito de pensamento computacional, no Capítulo 3, fundamentando-se na proposta inicial de Papert (1980, 1985), percebemos que o pensamento computacional é algo que tem sido investigado há anos - é o que percebemos ao analisar trabalhos como Alves (2013), Dullius (2008) que tratam implicitamente do tema. A questão

agora é que esta retomada nos dá folego para pensar mais claramente que competências são essas que a introdução de uma linguagem de programação e de conceitos de Ciência da Computação poderão trazer para o cidadão do século XXI.

A emergência desse contexto nos faz pensar as perspectivas educacionais e a relação do pensamento computacional com o professor. O PC vem ao encontro de um tempo em que não se pode apenas ensinar a combinar informações, mas a utilizar todos os recursos tecnológicos para transformar o aluno em um sujeito ativo e criador de conteúdo.

Para buscar essa conexão entre o PC e o professor, optamos por uma linha teórica baseada no Construtivismo de Piaget e no Construcionismo de Papert e seus seguidores. A nosso ver, as plataformas que seguem um paradigma de blocos possuem uma herança da filosofia Logo e, conseqüentemente, dos princípios construtivistas de Piaget (1978). Seguindo nessa linha, buscamos fundamentar teoricamente a postura do professor frente a esta nova competência, o pensamento computacional.

Conforme mostramos, existem inúmeros recursos disponíveis para o desenvolvimento do pensamento computacional. Como cenário desta pesquisa, elencamos a plataforma *Code.org*, por ser um recurso mundial e, principalmente, por apresentar um grande aparato de recursos para o professor.

Buscamos descobrir os princípios epistemológicos presentes na plataforma, as estratégias para o ensino e a aprendizagem do pensamento computacional, as dificuldades apresentadas pelos professores e, por fim, traçar indicadores que possam servir de base para a formação de professores.

A partir da análise, emergiram três categorias: Concepções de Aprendizagem, Aspectos promotores da aprendizagem e Aspectos dificultadores da aprendizagem. Na primeira categoria, emergiram duas subcategorias. São elas o empirismo e o construtivismo.

Como resultado da primeira categoria, percebemos que as fontes apresentam traços de concepções de aprendizagem empiristas. Apesar de a *Code.Org* ser uma plataforma nova e com diversos recursos e apoiadores, a proposta apresentada de ensino aprendizagem é limitada.

Os cursos apresentam uma sequência de atividades desplugadas e de atividades *on-line* que guiam o aluno passo a passo, sempre pelo mesmo caminho. É importante destacar que, do nosso ponto de vista, essas atividades possibilitam pouca liberdade, tanto para o professor quanto para o aluno que realiza as atividades. Outro ponto importante que já destacamos na análise, mas vale discutir mais uma vez, é o fato de as mensagens de erro e acerto não problematizarem e não estarem devidamente programadas para orientar o aluno e mobilizá-lo

a construir novas hipóteses. Essas mensagens simplesmente informam ao sujeito se ele conseguiu ou não atingir o objetivo.

Se a plataforma trouxesse uma concepção de aprendizagem voltada a uma perspectiva construcionista, possibilitaria ao sujeito experiência mais ampla, sem uma linearidade. Se nós compararmos as atividades da *Code.org* com o ambiente *Scratch*, por exemplo, percebemos uma grande diferença. Enquanto o primeiro é extremamente linear, o segundo permite a livre criação, desde o primeiro contato. Nesse contexto, surge a pergunta: Como podemos pensar em uma cultura de criação com atividades e recursos que privam o sujeito da livre criação?

Durante a investigação, evidenciamos que os planos de aula trazem a apresentação de conceitos sem incentivar os professores a deixarem os alunos pensarem sobre, explorarem e construir hipóteses. Essa abordagem é preocupante, porque se trata de um material feito para ser usado como recurso de formação de professores e é o que, segundo a *Code.org*, é usado nas atividades de formação propostas pela plataforma. Por outro lado, os planos de aula apresentam pontos positivos, como as atividades concretas e as relações externas, ou seja, a seção em que cada plano de aula sugere relações do tema da aula com outras áreas do conhecimento.

A partir da análise, descobrimos que as fontes analisadas apresentam como aspectos promotores da aprendizagem, basicamente, três estratégias distintas que formaram subcategorias. São elas a interação entre pares, o uso de atividades *on-line* e o uso de atividades *off-line*. Tanto os planos de aula como os trechos extraídos do fórum apresentam uma série de relatos de estratégias que utilizam atividades em grupo e a interação dos alunos para promoção da aprendizagem.

Além disso, descobrimos que o uso das atividades *unplugged* são fundamentais como recurso para o ensino e a aprendizagem dos conceitos iniciais da Ciência da Computação e estão presentes em diversas atividades, indo ao encontro do estudo proposto por Brackmann (2016). Este tipo de atividade, em especial se considerarmos o contexto das escolas públicas brasileiras, cria possibilidades para o ensino do PC que vão além do uso exclusivo de *softwares* e plataformas *on-line*, permitindo oportunidades para escolas e locais com baixa tecnologia.

Do mesmo modo, ficou evidente que uma das principais formas de aprendizagem com a plataforma é o uso das atividades *on-line* em que o aluno poderá desenvolver os tutoriais e os desafios propostos. Assim, o professor poderá lançar mão dessas estratégias para promover o desenvolvimento das habilidades e competências vinculadas ao pensamento computacional. A nossa ressalva é que o professor utilize esses recursos sempre com um olhar menos instrucionista do que o proposto na plataforma.

No caminho dos achados dessa pesquisa, convém salientar também os aspectos dificultadores da aprendizagem com pensamento computacional. Nessa categoria, encontramos três subcategorias, são elas: Tempo, Condições Estruturais e Dificuldades com TI. Embora esta dissertação tenha apresentado esta última categoria de uma maneira mais sutil, ela não é de menor importância para entendermos, na prática, o que acontece na sala de aula.

As subcategorias emergentes nessa seção nos permitem verificar problemas que não são exclusivos da realidade norte-americana e que se parecem muito com os vivenciados e relatados por professores no cenário nacional brasileiro.

As condições estruturais são de demasiada importância para que se possa trabalhar com as atividades do *Code.org*, que são em sua maioria *on-line*, ou seja, demandam rede, computadores, *internet* e uma série de recursos que nem sempre estão disponíveis em uma escola. Esse indicador é importante para se relacionar com a realidade brasileira, em que muitas escolas não dispõem de *internet* e, nem mesmo, de computadores de boa qualidade para que se possa desenvolver um trabalho nesse sentido.

Do mesmo modo, nessa categoria, evidenciamos problemas que estão relacionados também com a formação e o conhecimento do professor, que nem sempre se aproxima do mínimo necessário para as atividades propostas pelos planos de aula.

Esses indicativos nos permitem visualizar uma série de aspectos que são dificultadores da implantação de atividades de Ciência da Computação e pensamento computacional na educação básica, em especial no contexto brasileiro. Além disso, é importante discutir essas limitações e a importância de se pensar alternativas nesse sentido, pois, como já mencionamos, o pensamento computacional é uma competência presente na Base Nacional Comum Curricular (BRASIL, 2017).

Além disso, convém salientar, sobretudo, os problemas relacionados à categoria tempo, que refletem a estrutura disciplinarizada e engessada da educação norte-americana (e por que não brasileira?), ou seja, as questões de tempo giram em torno da estrutura da escola que deseja incluir um novo “conteúdo”, mas que não dispõe de um ajuste curricular que contemple de maneira eficaz esses novos conhecimentos. Pelo relato de alguns professores, podemos perceber que as atividades de Ciência da Computação muitas vezes ocupam a disciplina de informática, ou outras aulas e tarefas, como a hora de buscar livros na biblioteca, por exemplo.

A realidade dos professores analisada a partir das postagens não diverge da realidade dos professores no contexto nacional. Conteúdos a vencer, carga horária estreita e inúmeras atividades que sobrecarregam o fazer docente.

No caminho de solucionar alguns desses problemas relativos ao tempo e incluir o Pensamento Computacional como competência curricular, muitos países adicionaram uma nova disciplina, com intuito de contemplar as competências da Ciência da Computação. É o caso do Reino Unido (Valente, 2016). Contudo, essa postura é, segundo Papert (2016), avessa ao construcionismo, porque apenas reforça uma estrutura disciplinarizada e instrucionista, típica da educação tradicional.

Percebemos que, de um lado há a perspectiva de Papert (1980, 1985, 2006) e um pensamento computacional transversal como conhecimento; de outro, a inclusão de uma disciplina que fomente e instrumentalize os alunos. Para a eficácia da primeira, seria necessário um modelo pedagógico que se afastasse da disciplinarização e, possivelmente, uma aprendizagem baseada em projetos. A segunda, contudo, parece ser a solução mais simples, em especial do ponto de vista do mundo do trabalho, já que se introduz mais um conhecimento nas gavetas que, mais tarde, poderá ser utilizado para resolver alguns problemas.

Sabemos que a educação no cenário norte-americano é demasiada instrucionista (Papert, 1980) e, possivelmente, essas concepções são refletidas na estrutura da plataforma *Code.org*. O problema é que utilizar exclusivamente esses cursos para ensinar Ciência da Computação e, sobretudo, para formar professores, é reproduzir uma pedagogia da transmissão.

Observamos uma deficiência da plataforma em discutir e apresentar nos planos de aula a importância de o aluno construir o seu conhecimento e não de apenas realizar atividades desplugadas, atividades *on-line* e avaliações. Ou seja, a epistemologia que encontramos na plataforma dialoga com uma postura muito mais empirista do que construtivista, indo na contramão da filosofia LOGO.

Este estudo traz uma questão relevante e de necessária discussão: Analisar as concepções de aprendizagem para levantar princípios acerca da necessidade de formação de professores que, conforme já citado, é uma demanda no cenário nacional.

A literatura discute técnicas, mas muitas vezes não discute estratégias e concepções de aprendizagem, o que implica diretamente na deficiência de processos de ensino e aprendizagem e, conseqüentemente, na ausência de processos de formação docente.

Há inúmeros trabalhos de profissionais da computação, mas pouco se fala em pensamento computacional de um ponto de vista que se aproxime do construtivismo ou do construcionismo, como já apontamos anteriormente. Basta analisar as publicações dos últimos anos do Congresso Brasileiro de Informática na Educação. Essas publicações muitas vezes convergem para uma série de técnicas e recursos, mas não discutem a postura do professor e essas perspectivas do ponto de vista da aprendizagem.

Adotar uma postura construtivista permite refutar a ideia de transmissão de saberes e, dessa forma, possibilita que o professor utilize todo o aparato e os recursos que envolvem o PC para promover uma aprendizagem em que o aluno é o centro do processo. Assim, transformando os sujeitos em criadores, e não apenas em receptores de conceitos prontos de Ciência da Computação. A nosso ver, e em diálogo com os teóricos apresentados nessa dissertação, percebemos que a educação, de maneira geral, se beneficia dessa postura, pois permite ir além do que é possível com uma postura tradicional.

Sabemos que as discussões acerca do pensamento computacional aparecem nos últimos anos, em especial pela extrema necessidade de profissionais de TI. Isso nos revela um intuito excessivamente técnico e as escolas, mais uma vez, adicionam outra disciplina ou uma atividade extracurricular para suprir uma necessidade urgente.

Ao que nos parece, esse movimento se sustenta a partir da necessidade que a sociedade possui de profissionais na área da computação. E vem ao encontro, mais uma vez, do que Papert (1980) criticava ao se introduzir a computação nas escolas exclusivamente como uma nova disciplina, a fim de formar futuros profissionais para as áreas de Ciência da Computação.

Diante dessas reflexões, trouxemos, nesta dissertação, alguns indicadores que podem servir de base para se pensar a formação de professores. São eles: Entender que aprender é mais importante que ensinar; Ser mediador do processo de aprendizagem; Entender que aprender com o outro é fundamental; e, finalmente, Conhecer o conceito de pensamento computacional. Esses indicadores foram construídos à luz do referencial teórico proposto, buscando uma articulação com as evidências desta pesquisa.

Esses indicadores podem ser um caminho inicial para que se possa pensar que competências são necessárias para o professor do século XXI que pretende atuar com o pensamento computacional.

Por fim, cabe salientar que este estudo buscou investigar, principalmente, as concepções de aprendizagem presentes em uma plataforma *on-line*, lançando mão desses conhecimentos para articular indicadores para a formação docente.

Uma limitação desta investigação é não dispor de fontes suficientes para discutir como ocorreram as interações dos alunos a partir dos relatos dos professores como, por exemplo, verificar se existiu colaboração ou cooperação.

Outra limitação, nesse sentido, é haver pouca interação entre as postagens, o que não permite evidenciar os aspectos cooperativos ou colaborativos no próprio fórum. Este estudo poderia ser enriquecido se fosse possível analisar como acontecem os cursos de formação propostos pela plataforma *Code.org*. De todo modo, essas limitações não reduzem a relevância

do estudo, pois evidenciaram estratégias de colaboração que podem ser utilizadas em qualquer contexto de sala de aula, além de fornecer um panorama geral das propostas relatadas pelos professores no fórum. Além disso, fornece uma contribuição em dois campos ainda pouco explorados no cenário nacional: formação de professores e a plataforma *Code.org*. Nesse caminho, é importante salientar que este estudo pode servir de base para outros trabalhos futuros, como, por exemplo:

- Evidenciar as concepções de aprendizagem presentes em outras plataformas para o desenvolvimento do pensamento computacional (*Scratch, Tynker, AppInventor*, etc);
- Propor, por meio de uma pesquisa de campo com professores, outros indicadores necessários para a atuação docente com o Pensamento Computacional;
- Caracterizar o desenvolvimento do pensamento computacional na plataforma *Code.org*;
- Investigar como os professores podem utilizar as plataformas *on-line* para ensinar e aprender conceitos de Ciência da Computação;
- Explorar outras estratégias que podem ser utilizadas para contribuir para o ensino e a aprendizagem do pensamento computacional e
- Investigar como avaliar o desenvolvimento do pensamento computacional em cursos de formação pedagógica.

Além desses problemas aqui listados, este estudo fornece inquietações para o desenvolvimento de um estudo empírico com professores e alunos, a fim de investigar possíveis concepções de aprendizagem, aspectos promotores e aspectos dificultadores da aprendizagem, podendo mapear interações e trocas entre os participantes do grupo.

Por fim, salientamos que o pensamento computacional tem se constituído não apenas como uma competência para os profissionais das áreas técnicas, mas, sobretudo, como uma nova competência para o professor do século XXI, que tem agora uma nova perspectiva e responsabilidade, a de formar cidadãos aptos para se comunicarem em uma linguagem global baseada nas competências do pensamento computacional.

7 REFERÊNCIAS

ABOUT CSTA. Disponível em: <<https://www.csteachers.org/page/About>>. Acesso em: 16 abr. 2018.

ALMEIDA, Maria Elizabeth Bianconcini de. Educação e tecnologias no Brasil e em Portugal em três momentos de sua história. **Educação, formação & Tecnologias**, v. 1, n. 1, p.23-36, maio 2008. Disponível em: <Disponível em <http://eft.educom.pt>>. Acesso em: 12 out. 2016.

ALVES, Rafael Machado. **Duinoblocks: Desenho e implementação de um ambiente de Programação Visual para Robótica Educacional**. 2013. 112 f. Dissertação (Mestrado) - Curso de Programa de Pós-graduação em Informática, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2013.

AMARAL, Érico Marcelo Hoff do. **Processos de Ensino e Aprendizagem de Algoritmos Integrando Ambientes Imersivos e o Paradigma de Blocos de Programação Visual**. 2015. 255 f. Tese (Doutorado) - Curso de Programa de Pós-graduação em Informática na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2015.

BARCELOS, Thiago et al. Relações entre o Pensamento Computacional e a Matemática: uma Revisão Sistemática da Literatura. In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 4., 2015, Maceió. **Anais dos Workshops do IV Congresso Brasileiro de Informática na Educação**. Maceió: CBIE, 2015. p. 1369 - 1378. Disponível em: <<http://www.br-ie.org/pub/index.php/wcbie/article/view/6311/4420>>. Acesso em: 05 dez. 2016.

BARCELOS, Thiago; BORTOLETTO, Rodrigo; ANDRIOLI, Mary Grace. Formação online para o desenvolvimento do Pensamento Computacional em professores de Matemática. In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 5., 2016, Uberlândia. **Anais dos Workshops do V Congresso Brasileiro de Informática na Educação**. Uberlândia: CBIE, 2016. p. 1228 - 1237. Disponível em: <<http://br-ie.org/pub/index.php/wcbie/article/view/7048/4922>>. Acesso em: 5 dez. 2016.

BARNES, David J.; KÖLLING, Michael. **Programação Orientada a Objetos com Java**. São Paulo: Pearson Prentice Hall, 2004.

BECKER, Fernando. **Modelos pedagógicos e modelos epistemológicos**. Educação e Realidade, Porto Alegre, RS, v. 19, n. 1, p. 89-96, 1999.

BECKER, Fernando. **Epistemologia do professor: cotidiano da escola**. 15. ed. Petrópolis: Vozes, 1993.

BELL, Tim; WITTEN, Ian H.; FELLOWS, Mike. **Computação Unplugged: Ensinando Ciência da Computação sem o uso do computador**. [salvador]: Desconhecido., 2011. Disponível em: <<https://classic.csunplugged.org/wp-content/uploads/2014/12/CSUnpluggedTeachers-portuguese-brazil-feb-2011.pdf>>. Acesso em: 05 abr. 2018.

BLISKTAİN, Paulo. **O pensamento computacional e a reinvenção do computador na educação**. 2008. Disponível em:

http://www.blikstein.com/paulo/documents/online/ol_pensamento_computacional.html. Acesso em: 02 ago. 2016.

BOMBASAR, James et al. Ferramentas para o Ensino-Aprendizagem do Pensamento Computacional: onde está Alan Turing? In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 26., 2015, Maceió. **Anais do XXVI Simpósio Brasileiro de Informática na Educação**. Maceió: SBIE, 2015. p. 81 - 90. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/5120/3525>>. Acesso em: 05 dez. 2016.

BOUCINHA, Rafael Marimon. **Aprendizagem Do Pensamento Computacional e Desenvolvimento do Raciocínio**. 2017. 132 f. Tese (Doutorado) - Curso de Programa de Pós-graduação em Informática na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017.

BOYLE, Martin. **The History of Mr. Papert**. 2004. Disponível em: <<http://www.stager.org/omaet2004/papertbio.html>>. Acesso em: 30 set. 2017.

BOZOLAN, Sandra Muniz. **O pensamento computacional: ensino e aprendizagem através do software processing**. 2016. 145 f. Dissertação (Mestrado) - Curso de Programa de Estudos Pós-graduados em Tecnologia da Inteligência e Design Digital, Pontifícia Universidade Católica de São Paulo, São Paulo, 2016.

BRACKMANN, Christian Puhmann. **Desenvolvimento Do Pensamento Computacional Através De Atividades Desplugadas Na Educação Básica**. 2017. 226 f. Tese (Doutorado) - Curso de Programa de Pós-graduação em Informática na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017.

BRASIL. Ministério da Educação. **Base Nacional Comum Curricular – BNCC**. Brasília, DF, 2017.

BRENNAN, Karen; RESNICK, Michel. **New frameworks for studying and assessing the development of computational thinking**. 2012. Disponível em: <http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf>. Acesso em: 12 out. 2016.

BRESSAN, Manuelle Lopes Quintas. **Scratch! Um Estudo De Caso**. 2016. 2012 f. Dissertação (Mestrado) - Curso de Programa de Pós-graduação em Tecnologia e Sociedade, Universidade Tecnológica do Paraná, Curitiba, 2016.

CARBAJAL, Marleny Luque. **Design e desenvolvimento de um ambiente de programação tangível de baixo custo para crianças**. 2016. 87 f. Dissertação (Mestrado) - Curso de Programa de Pós-graduação em Ciência da Computação, Universidade Estadual de Campinas, Campinas, 2016.

CASTELSS, Manuel. **A era da Informação: economia, sociedade e cultura**. 6. ed. São Paulo: Paz e Terra, 2002.

CAVALCANTE, Ahememenson Fernandes; COSTA, Leonardo dos Santos; ARAÚJO, Ana Liz Souto Oliveira de. Um Estudo de Caso Sobre Competências do Pensamento Computacional Estimuladas na Programação em Blocos no Code. Org. In: CONGRESSO

BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 5., 2016, Uberlândia. **Anais dos Workshops do V Congresso Brasileiro de Informática na Educação**. Uberlândia: CBIE, 2016. p. 1117 - 1126. Disponível em: <<http://br-ie.org/pub/index.php/wcbie/article/viewFile/7037/4911>>. Acesso em: 05 dez. 2016.

CAVALCANTE, Ahemenson Fernandes et al. Um Estudo Exploratório da Aplicação de Pensamento Computacional Baseado nas Perspectivas de Professores do Ensino Médio. In: VI CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 6., 2017, Recife. **Anais dos Workshops do VI Congresso Brasileiro de Informática na Educação**. Recife: CBIE, 2017. p. 992 - 1001. Disponível em: <<http://br-ie.org/pub/index.php/wcbie/article/view/7488>>. Acesso em: 01 mar. 2018.

CODE. CODE, 2016. Disponível em: <https://code.org/>, Acesso em: 10 de outubro 2016.

COUTO, Gabriel Miletto. **Pensamento Computacional Educacional: Ensaio sobre uma perspectiva libertadora**. 2017. 181 f. Dissertação (Mestrado) - Curso de Programa de Pós-graduação em Educação e Currículo, Educação, Pontifícia Universidade Católica de São Paulo, São Paulo, 2017.

DARGAINS, André Rachman. **Estudo Exploratório Sobre o Uso Da Robótica Educacional No Ensino de Programação Introdutória**. 2015. 239 f. Dissertação (Mestrado) - Curso de Programa de Pós-graduação em Informática, Matemática, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2015.

DEMO, Pedro. Olhar do Educador e Novas Tecnologias. **Boletim Técnico do Senac**, Rio de Janeiro, v. 37, n. 2, p.15-25, maio 2011.

DILLENBURG, P., Baker, M. Blaye, A. & O'Malley C. (1996) The evolution of research on collaborative learning, In: E. Spada & P. Reiman (Eds.) **Learning in Humans and Machine: Towards an interdisciplinary learning science**. (p. 189 - 211). Exford: Elsevier.

DULLIUS, Simone Rosanelli. **O Ambiente De Autoria Scratch E Suas Possibilidades De Apoio Ao Processo De Aprendizagem**. 2008. 56 f. Monografia (Especialização) - Curso de Programa de Pós-graduação em Informática na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2008.

FAGUNDES, Léa da Cruz. **Aprendizes do Futuro: as inovações começaram**. Brasília: Proinfo/seed/mec, 1999.

FARIAS, Adelito B.; ANDRADE, Wilkerson L.; ALENCAR, Rayana A. Pensamento Computacional em Sala de Aula: Desafios, Possibilidades e a Formação Docente. In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 4., 2015, Maceió. **Anais dos Workshops do IV CBIE**. Maceió: CBIE, 2015. v. 1235, p. 1226 - 1235. Disponível em: <<http://www.br-ie.org/pub/index.php/wcbie/article/view/6262/4384>>. Acesso em: 05 dez. 2016.

FERNANDES, Hugo Batista; SILVEIRA, Ismar Frango. Pensamento Computacional: iniciativas para o seu desenvolvimento por meio da modalidade de Ensino a Distância. In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 5., 2016, Uberlândia. **Anais dos Workshops do V Congresso Brasileiro de Informática na**

Educação. Uberlândia: CBIE, 2016. p. 1070 - 1077. Disponível em: < <http://br-ie.org/pub/index.php/wcbie/article/view/7032/4906> >. Acesso em: 5 dez. 2016.

FONSECA FILHO, Cléusio. **História da Computação:** O Caminho do pensamento e da tecnologia. Porto Alegre: EdpucRS, 2007.

FREIRE, Paulo. **Pedagogia da Autonomia:** Saberes Necessários à Prática Educativa. 39. ed. São Paulo: Paz e Terra, 1996.

GOMES, et al. Avaliação de um Jogo Educativo para o Desenvolvimento do Pensamento Computacional na Educação. In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 4., 2015, Maceió. **Anais dos Workshops do IV Congresso Brasileiro de Informática na Educação.** Maceió: CBIE, 2015. p. 1349 - 1358. Disponível em: < <http://www.br-ie.org/pub/index.php/wcbie/article/view/6307/4416> >. Acesso em: 5 dez. 2016.

GOMES, Tancicleide C. S.; ALENCAR, Andreza L. B. de. Análise Empírica de Jogos Educativos para Dispositivos Móveis voltados a Disseminação do Pensamento Computacional na Educação Básica. In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 4., 2015, Maceió. **Anais dos Workshops do IV Congresso Brasileiro de Informática na Educação.** Maceió: CBIE, 2015. p. 731 - 740. Disponível em: <<http://www.br-ie.org/pub/index.php/wcbie/article/view/6101/4277>>. Acesso em: 5 dez. 2016.

HOBBSAWM, Eric John. **Era dos extremos:** o breve século XX 1914-1991. 2. ed. São Paulo: Cia das Letras, 1995.

KEBACH, Patrícia Fernanda Carmem. O professor Construtivista: Um pesquisador em ação. In: BECKER, Fernando; MARQUES, Tania B. I. (Org.). **Ser professor é Ser pesquisador.** Porto Alegre: Editora Mediação, 2007. p. 43-53.

LEITE, Maici Duarte et al. Pensamento Computacional nas Escolas: Limitado pela Tecnologia, Infraestrutura ou Prática Docente? In: VI CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 6., 2017, Recife. **Anais dos Workshops do IV Congresso Brasileiro de Informática na Educação.** Recife: Cbie, 2017. p. 1002 - 1010. Disponível em: <<http://www.br-ie.org/pub/index.php/wcbie/article/view/7489>>. Acesso em: 05 abr. 2018.

LOCKE, JOHN. **Ensaio Acerca do Entendimento Humano.** São Paulo: Nova Cultural, 1999.

LOPES, Daniel de Queiroz. **Brincando com Robôs:** desenhando problemas e inventando porquês. Santa Cruz do Sul: EDUNISC, 2010.

MACEDO, Rosa Maria Stefanini. Piaget: Vida e Obra. In: PIAGET, Jean. **A Epistemologia Genética; Sabedoria e Ilusões da Filosofia; Problemas de Psicologia Genética.** São Paulo: Cultural, 1978. p. 6-18

MIT MEDIA LAB (Massachussets). **In memory:** Seymour Papert. 2017. Disponível em: <<https://www.media.mit.edu/posts/in-memory-seymour-papert/>>. Acesso em: 30 set. 2017.

MITCH Resnick: Let's teach kids to code. [s.l.], 2012. Son., color. Legendado. Disponível em: <https://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code?language=pt-br>. Acesso em: 05 jun. 2016.

MORAES, R.; GALIAZZI, M. C. Análise textual discursiva: processo reconstrutivo de múltiplas faces. **Ciência & Educação**, v. 12, n. 1, p. 117-128, 2003.

MORAES, R; GALIAZZI, M. C. **Análise Textual Discursiva**. Ijuí: Unijuí, 2011.

MORIN, Edgar. **Ciência com consciência**. Rio de Janeiro: 8. ed. Bertrand Brasil. 2005
MUNARI, A. **Jean Piaget**. Recife: Massangana, 2010.

OLIVEIRA JÚNIOR, Evandro Jose de. **Plataforma de Suporte às Ferramentas de Aprendizagem Para o Ensino de Lógica de Programação na Educação Básica**. 2017. 59 f. Dissertação (Mestrado) - Curso de Programa de Pós-graduação em Educação, Universidade Federal dos Vales do Jequitinhonha e Mucuri, Diamantina, 2017.

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

PAPERT, Seymour. **Logo: computadores e educação**. São Paulo: Brasiliense, 1988.

PAPERT, Seymour. **A máquina das crianças: repensando a escola na era da informática**. Porto Alegre: Artmed, 2008.

PIAGET, Jean. **O Nascimento da Inteligência na Criança**. Rio de Janeiro: Zahar, 1970.

PIAGET, Jean. **A equilibração das estruturas cognitivas**. Rio de Janeiro: Zahar, 1975.

PIAGET, Jean. **Epistemologia genética; Sabedoria e ilusões da filosofia**; Problemas de psicologia genética, 1978.

PIAGET, J. **A representação do mundo na criança**. Rio de Janeiro: Difel, 1982.

PIAGET, Jean. **Pedagogia**. Lisboa: Horizontes Pedagógicos, 1998.

PRENSKY, Marc. Digital Natives, Digital Immigrants. **MCB University Press**, [s.l.], p.1-6, Oct. 2001. Disponível em: <<http://www.marcprensky.com/writing/Prensky - Digital Natives, Digital Immigrants - Part1.pdf>>. Acesso em: 10 out. 2016.

RANGEL, Ana Cristina S.. **Educação Matemática e a Construção do Número pela Criança**. Porto Alegre: Artes Médicas, 1992.

SANTOS, Vera Lúcia Bertoni dos. O Método Clínico-Crítico de Jean Piaget: Uma Aula com Silvia Parrat-Dayan. In: BECKER, Fernando; MARQUES, Tania B. I.. **Ser professor é Ser pesquisador**. Porto Alegre: Mediação, 2007. Cap. 3. p. 29-41.

SCHOFIELD, Jack (Ed.). Seymour Papert obituary. **The Guardian**. Londres, 3 ago. 2016. Disponível em: <<https://www.theguardian.com/education/2016/aug/03/seymour-papert-obituary>>. Acesso em: 28 set. 2017.

SILVA, Marco. Indicadores De Interatividade Para o Professor Presencial e On-Line. **Revista Diálogo Educacional**, Curitiba, V. 4, N. 12, P.93-109, 2004.

SILVA, Mariana Cardoso da. **Robótica Educacional Livre: Um relato de prática no Ensino Fundamental**. 2017. 108 f. Dissertação (Mestrado) - Curso de Programa de Pós-graduação em Educação e Currículo, Educação, Pontifícia Universidade Católica de São Paulo, São Paulo, 2017.

SOUZA, Isabelle M. L.; RODRIGUES, Rivanilson S.; ANDRADE, Wilkerson L.. Introdução do Pensamento Computacional na Formação Docente para Ensino de Robótica Educacional. In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 5., 2016, Uberlândia. **Anais dos Workshops do V Congresso Brasileiro de Informática na Educação**. Uberlândia: CBIE, 2016. p. 1265 - 1274. Disponível em: <<http://www.br-ie.org/pub/index.php/wcbie/article/view/7052/4926>>. Acesso em: 5 dez. 2016.

SPERB, Bruno Fagundes. **Oficinas De Construção De Jogos Digitais Em Etoys: Aprender em Movimento**. 2014. 90 f. Dissertação (Mestrado) - Curso de Programa de Pós-graduação em Psicologia Social e Institucional, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2014.

STELLA, Ana Lucia. **Utilizando O Pensamento Computacional e a Computação Criativa No Ensino Da Linguagem De Programação Scratch Para Alunos Do Ensino Fundamental**. 2016. 92 f. Dissertação (Mestrado) - Curso de Programa de Pós-graduação em Tecnologia, Universidade Estadual de Campinas, Campinas, 2016.

STRATHERN, Paul. **Santo Agostinho: Em 90 minutos**. Rio de Janeiro: Zahar, 1997.

TARDIF, Maurice. **Saberes Docentes e Formação Profissional**. 12. ed. Petrópolis: Vozes, 2011.

VALENTE, José Armando (Org.). **O Computador na Sociedade do Conhecimento**. Campinas: Unicamp/nied, 1999.

VALENTE, José. Armando. (2016), **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, v.14, n.03, p. 864 – 897.

VALENTINI, Carla Beatris. **Tecendo e aprendendo: redes sociocognitivas e autopoieticas em ambientes virtuais de aprendizagem**. 2003. 223 f. Tese (Doutorado) - Curso de Pós-graduação em Informática na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2003.

VASCONCELLOS, Celso dos Santos. **Avaliação: Concepção Dialética-Libertadora do Processo de Avaliação Escolar**. São Paulo: Libertad, 2008.

WADSWORTH, Barry J. **Inteligência Afetividade da Criança na Teoria de Piaget**. São Paulo: Pioneira, 1996.

WING, Jeannette M.. Computational Thinking. **Communications of The Association For Computing Machinery**, [s.l.], v. 49, n. 3, p.33-35, mar. 2006.

WING, Jeannette M. **Computational Thinking: What and Why?** 2010. Disponível em: <<http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>>. Acesso em: 5 out. 2016.

WING, Jeannette M.. **Computational thinking and thinking about computing.** 2008. Disponível em: <<http://rsta.royalsocietypublishing.org/content/366/1881/3717>>. Acesso em: 10 out. 2016.

WOHLIN, C. et al. Systematic Literature Reviews. **Experimentation in Software Engineering.** Springer Berlin Heidelberg, 2012. p. 45–54.