

## **A CONSTRUÇÃO DO PENSAMENTO COMPUTACIONAL NOS ANOS INICIAIS DO ENSINO FUNDAMENTAL: UMA ANÁLISE CARTOGRÁFICA**

### **CONSTRUCTION OF COMPUTATIONAL THINKING IN THE EARLY YEARS OF ELEMENTARY SCHOOL: A CARTOGRAPHIC ANALYSIS**

Cristina M. Pescador<sup>1</sup>  
Sintian Schmidt<sup>2</sup>  
Aline Silva De Bona<sup>3</sup>

#### **RESUMO**

Este estudo se propôs a cartografar e identificar indícios de construção de pensamento computacional por estudantes do 2º ano do Ensino Fundamental em uma escola da rede pública municipal no interior do Rio Grande do Sul. Durante alguns períodos, na aulas regulares desenvolvidas no laboratório de informática na educação (LIE), as crianças foram desafiadas a resolver problemas ora usando um robô programável, ora desenvolvendo atividades propostas no site “Hora do Código”, ora trabalhando em atividades de programação com o software Scratch. O método de investigação utilizado se baseia nos princípios de Pesquisa Cartográfica propostos por Kastrup, com base em Deleuze e Guattari. A análise dos movimentos dos estudantes na busca de soluções para os problemas apresentados é feita à luz dos conceitos de pensamento computacional (WING), letramento em codificação (DUDENEY et al) e habilidades digitais (PAPERT). Destaca-se como resultado o desenvolvimento do processo de aprendizagem autônomo dos estudantes/crianças, e as evidências de abstração nas resoluções das atividades propostas, assim como se percebe a ação/pensamento coletivo das crianças quando desejam compartilhar com os colegas suas soluções, do seu jeito.

**Palavras-chave:** Pensamento computacional. Letramento em codificação. Habilidades digitais. Cartografia.

#### **ABSTRACT**

This study intends to map and identify signs of the construction of computational thinking by second-grade students from an elementary municipal school in Rio Grande do Sul. During some periods, during regular classes developed at the school’s computer lab, the children were challenged to solve problems sometimes using a programmable robot, or developing activities proposed by the website “Hour of code”, or working with programming activities on a software called Scratch. The research method is based on principles of Cartography as proposed by Kastrup, based on Deleuze and Guattari. The analysis of how students tried to solve the problems presented is carried out considering concepts of computational thinking (WING), coding literacy (DUDENEY et al) and digital abilities (PAPERT). The results point out children’s autonomous learning process and evidences of abstraction in solving the activities proposed. They also show children’s collective actions/thoughts while sharing solutions with their peers, their way.

**Keywords:** Computational thinking. Coding literacy. Digital abilities. Cartography.

<sup>1</sup> Professora no Programa de Pós-Graduação em Educação (Mestrado em Educação) na Universidade de Caxias do Sul, Doutora em Informática em Educação. ID Lattes: K4737450P4 Orcid: 0000-0002-9659-3685 Email: cmpescad@ucs.br

<sup>2</sup> Professora na Rede Municipal de Ensino e Assessora Pedagógica de Informática Educativa da Secretaria Municipal da Educação em Caxias do Sul, Mestre em Educação, ID Lattes: K4631980Y9 Orcid: 0000-0003-0261-6418 Email: sintians@gmail.com

<sup>3</sup> Professora no Instituto Federal do Rio Grande do Sul, Campus de Osório, Doutora em Informática em Educação. ID Lattes: K4481712Y9 Orcid: 0000-0002-0052-1987 Email: aline.bona@osorio.ifrs.edu.br



## Introdução

Há inúmeras iniciativas com foco em atividades voltadas para ensinar crianças, jovens e adultos a programar. Algumas têm um foco maior no desenvolvimento profissional e no mercado de trabalho como, por exemplo, a Code Academy. Outras, mais focadas em crianças e adolescentes e no desenvolvimento do pensamento computacional, como o projeto “A Hora do Código”, o projeto do Massachusetts Institute of Technology (MIT) “Scratch”, o “Code Org”, e o “Code Club Brasil”.

Nessa perspectiva, este trabalho de cunho exploratório investigativo se propõe a observar, registrar e analisar o trabalho desenvolvido com foco em Pensamento Computacional com dois grupos de estudantes do 2º ano do Ensino Fundamental em uma escola da rede pública municipal com o intuito de verificar e identificar que indícios de construção do pensamento computacional podem ser percebidos enquanto essas crianças são desafiadas a resolver problemas com a programação. Entendemos que, ao observá-las enquanto “brincam de programar” (grifo nosso), ou como descrito por Papert (1988, 2008), enquanto “ensinam o computador a pensar” (grifo do autor), poderemos identificar algumas possibilidades que se abrem para seu desenvolvimento cognitivo e construção do processo de abstração.

Como opção para o método de investigação, optamos pela cartografia, proposta por Deleuze e Guattari (1995) e descrita por Kastrup (2007). Essa escolha se justifica porque permite ao pesquisador observar os processos em andamento, justificando o envolvimento das autoras com a experiência em questão. Cabe esclarecer que a ideia na cartografia não é definir um conjunto de regras a serem aplicadas durante a pesquisa. Na verdade, o trabalho cartográfico requer foco nas posturas e atitudes do pesquisador, que se mantém atento às experiências vividas com uma atitude constante de atenção flutuante e aberto ao processo que emerge aos seus olhos. Ao marcar os caminhos e movimentos, o pesquisador cartográfico mapeia dados, construindo análises possíveis baseadas em seus objetivos e referenciais teóricos.

Estudos como este podem contribuir para a compreensão de uma aprendizagem autônoma, em que os estudantes agem como protagonistas, aprendendo a pensar com os problemas que a programação lhes apresenta e a buscar soluções para esses problemas. E essa compreensão poderá servir de inspiração para o planejamento e desenvolvimento de programas de capacitação e formação de professores que consigam pensar suas aulas e sua atuação dentro de um contexto de crianças que programam.

Assim, apresentamos este estudo em cinco partes, o qual inicia com essa introdução,



seguida de uma contextualização teórica apresentando os conceitos que norteiam o estudo: cultura digital, pensamento computacional, letramento em codificação e habilidades digitais. A seguir, apresentamos uma descrição da cartografia, método investigativo que orienta nossas ações, intervenções e considerações. Depois, apresentamos a produção e construção dos dados, tecidos em conjunto com uma análise fundamentada nos conceitos norteadores do estudo, bem como em conceitos cuja compreensão se tornou necessária em função da cartografia e dos movimentos que se fizeram conhecer ao longo do processo. Na sessão seguinte, apresentamos algumas considerações sobre o estudo e as referências utilizadas em sua construção.

### **Conceitos norteadores**

Inicialmente, retomamos a importância de se ter em mente que, na pesquisa cartográfica, o pesquisador não carrega consigo procedimentos pré-estabelecidos ou planejados nem adota a nomenclatura de “pressupostos teóricos” para o aporte teórico que lhe dá sustentação para o estudo. Pescador (2016) compara a conduta investigativa do pesquisador-cartográfico à caminhada de um explorador que mantém sua atenção aberta ao novo e aos relevos que se fazem perceber em sua caminhada, carregando consigo alguns conceitos teóricos que lhe servem de norteadores. Esses conceitos irão sustentar sua exploração e análise do que percebeu, mas, por estar aberto ao novo, outros conceitos poderão ser agregados ao longo da caminhada.

Diante disso, escolhemos como nossos conceitos norteadores a cultura digital segundo Lévy (2000) e Lemos (2009) refletindo como a escola pode se inserir nessa cultura; o pensamento computacional, considerado por Wing (2011) como o “novo letramento do século 21” (p. 3), associando-o ao que Dudeney et al (2016) chamam de letramento em codificação e relacionando ao que Papert (1988, 2008) apresenta como habilidades digitais, isto é, fluência digital e tecnológica associadas ao “pensar como um computador” e ao “ensinar o computador a pensar”.

### **A escola e a cultura digital**

Estudar e pesquisar a inserção de equipamentos e recursos tecnológicos em sala de aula diverge do que a mídia jornalística insiste em apresentar como algo que “torne as aulas mais atraentes”. Valente (2011) afirma que o impacto que a tecnologia pode exercer sobre a aprendizagem dos estudantes não advém da simples inserção de computadores ou outros dispositivos móveis na sala de aula, mas sim de um ambiente enriquecido que ofereça condições favoráveis ao aprendizado.

De modo semelhante, Blikstein (2008) argumenta que não é ensinando as crianças a



navegar na internet ou a combinar informações que se cria conhecimento novo, pois deveríamos focar em uma das habilidades mais importantes para este século, “a habilidade de transformar teorias e hipótese em modelos e programas de computador, executá-los, depurá-los, e utilizá-los para redesenhar processos produtivos [...]” (p. 01).

Esse contexto, caracterizado por inúmeras relações humanas e manifestações sociais mediadas pelas TDIC via redes interconectadas, é descrito por Lévy (2000) e Lemos (2009) como “cibercultura” ou “cultura digital”. Segundo Lévy (2000), a cibercultura é um conjunto de atitudes, práticas, modos de pensar, valores e técnicas “que se desenvolvem com o crescimento do ciberespaço” (p. 17).

Entretanto, a escola e o sistema educacional ainda não possuem a estrutura logística e o planejamento necessários para incorporar essas tecnologias. Tampouco conseguiram quebrar o que Lévy (2000) chama de um hábito antropológico em que um professor fala enquanto o aluno escreve, como se essa fosse a única interação possível para garantir a aprendizagem.

Assim, inserir a escola na cultura digital representa ampliar o espaço de aprendizagem favorecendo uma reaproximação da proposta original de Papert (1988) em que o computador é uma máquina para a criança pensar sobre o pensar. Nessa perspectiva, entendemos que o acesso a softwares de programação e produção poderá promover o desenvolvimento da fluência tecnológica e a construção do pensamento computacional e do letramento em codificação, ou seja, o uso das tecnologias com autonomia e autoria.

#### Pensamento computacional, letramento em codificação e habilidades digitais

O uso das TDIC na escola percorre diversos caminhos, fundamentado em diferentes propostas pedagógicas e pressupostos teóricos. Um desses pressupostos baseia-se no construcionismo de Papert (1988; 2008), cujo foco está na construção de conhecimento e na programação, pois ao aprender a programar, o estudante explora como pensa e, nessa exploração, desenvolve “novas possibilidades de aprender, pensar e crescer, tanto emocional como cognitivamente (1988, p. 34).

Passadas mais de três décadas dos primeiros movimentos de inclusão digital no país, ainda estamos longe da universalização do acesso às TDIC imaginada por Papert (1988) e descrita por Resnick (2012) como um lugar em que “as crianças não apenas aprendem a usar novas tecnologias, mas se tornam fluentes com elas” (2012, p. 42. Tradução livre). Nessa perspectiva, o computador poderia ser uma “máquina de aprender” (Papert, 2008). Em vez disso, as aulas de Informática Educativa propostas para estudantes do Ensino Fundamental, baseadas em softwares



educativos e jogos focados na repetição e memorização, ainda reproduzem e privilegiam o instrucionismo, conforme descreve Valente (1999).

Para superar essas concepções, Blikstein (2008) propõe o desenvolvimento de habilidades e conhecimentos necessários para o pleno exercício da cidadania no século XXI, enfatizando o Pensamento Computacional (PC) enquanto “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.” (p. 01).

Wing (2006), por sua vez, considera que o PC traz uma metodologia para resolver problemas, combinando o pensamento crítico com os fundamentos da computação, favorecendo o aprender a aprender. Entre as capacidades mencionadas pela autora, destacamos neste estudo a capacidade para pensar em diferentes níveis de abstração, a capacidade humana para resolver problemas e a habilidade de pensar além do mundo físico e construir mundos virtuais.

A abstração pode ser o processo mais elevado do Pensamento Computacional de modo que o sujeito desenvolva habilidades intelectuais que lhe permitem generalizar, criar parâmetros e definir padrões em qualquer domínio, sem necessariamente se restringir à ciência computacional (WING, 2011). Nessa mesma perspectiva, observa-se que a CSTA - Computer Science Teachers Association (2011) vem difundindo o caráter transversal da introdução dos conceitos de computação em todas as áreas do conhecimento.

Nessa perspectiva, habilidade de programar tem sido considerada tão importante quanto ler, escrever, falar e fazer operações aritméticas por Wing (2006), o que tem servido para estimular novas práticas no uso das TDIC na educação.

Neste estudo, apresentamos algumas reflexões a partir da experiência vivida pelas pesquisadoras junto a estudantes do 2º ano dos anos iniciais em uma escola de ensino fundamental da rede pública municipal, nas aulas de Informática Educativa: um robô programável (Bee-Bot®), atividades de introdução ao letramento em codificação propostas online (“Hora do Código”) e atividades de programação com o software “Scratch”. O Bee-Bot® é um robô programável, considerado pelas crianças um brinquedo, que executa funções relacionadas à sua movimentação e rotação, inspiradas nas pesquisas de Papert (2008) e na linguagem de programação Logo (PAPERT, 1988). “A Hora do Código” é uma plataforma online de jogos digitais baseados na programação visual, contando com tutoriais e cursos que visam ao ensino de programação. E o Scratch é um ambiente de programação visual desenvolvido a partir da linguagem Logo, baseado em blocos, com o objetivo de possibilitar criar sem aprender ou dominar a sintaxe de uma linguagem de programação. A intersecção entre os três recursos está no



desenvolvimento de habilidades cognitivas, de forma lúdica e acessível, com crianças com idade entre 7 e 8 anos.

Mais recentemente, estudos realizados por Blikstein (2013) mostram que recursos tecnológicos que desafiam a criança a desenvolver o pensamento computacional abriram caminho para o movimento *maker*. Ao compreender os usos complexos da tecnologia e aprender a programar, as crianças deixam de lado a passividade de ser meros consumidores de produtos tecnológicos e passam a ser construtores ativos com o uso das tecnologias.

### **Cartografia como método de investigação**

A cartografia, proposta por Deleuze e Guattari (1995) e descrita por Kastrup (2007), possibilita a produção dos dados da investigação acompanhando o processo desde a introdução de um robô programável (Bee-Bot®) aos estudantes até o envolvimento dos estudantes com “A Hora do Código” e a programação em “Scratch”. Trata-se, pois, de pesquisa que se faz “em movimento, no acompanhamento de processos, que nos tocam, nos transformam e produzem mundos.” (KASTRUP, 2007, p. 73).

Enquanto pesquisa qualitativa de natureza exploratória, a cartografia tem sido eleita por pesquisadores que se propõem a observar processos em andamento, o que justifica o envolvimento das autoras com a experiência em questão. O trabalho cartográfico está centrado nas próprias posturas e atitudes do pesquisador, que deve se manter em uma atitude constante de atenção flutuante e aberto ao processo que emerge aos seus olhos, pois são das experiências vividas que surgem as pistas cartográficas, de acordo com Passos, Kastrup e Escóssia (2010).

Na cartografia, um aspecto importante é a atenção e suas variedades de funcionamento, consistindo em pista fundamental para orientar a ação do cartógrafo. Kastrup (2007) define quatro variedades do funcionamento da atenção do cartógrafo. São elas: (1) o rastreio, espécie de varredura do território, momento em que o pesquisador está buscando pistas ao acompanhar os movimentos contínuos, que podem se destacar e se fazer conhecer. Esse é o momento do (2) toque, quando algo desperta a atenção do cartógrafo, como um “fenômeno de irrupção de algo no campo perceptivo que instala uma situação de decalagem em relação ao estado cognitivo anterior” (KASTRUP, 2007, p.19). Aqui acontece o (3) pouso, uma reconfiguração da atenção, “o momento em que o cartógrafo faz uma parada e seu campo de atenção se fecha numa espécie de zoom, criando janelas atencionais onde a atenção muda de escala e se focaliza naquilo que foi tocado” (SCHMIDT, 2015). E, finalmente, tem-se o (4) reconhecimento atento, momento de identificar o que está acontecendo dentro do processo, focando a atenção naquilo que afetou o



pesquisador (KASTRUP, 2010, p. 45).

Ao mapear os dados que produz em suas observações, o cartógrafo está registrando o que Pescador (2016, p. 101) descreve como uma “representação de algo que estava em movimento em determinado instante da observação. Tal é a situação, por exemplo, de um rio em um mapa. O fato de ele estar representado cartograficamente não significa que suas águas estejam paradas.” Assim, na pesquisa cartográfica, é necessário manter uma atitude aberta para acolher o inesperado (KASTRUP, 2007), como também se manter conectado ao objeto de pesquisa, não permitindo, portanto, que o inesperado o afaste desse objeto.

Neste trabalho apresentamos três momentos da variedade atencional das pesquisadoras, mais especificamente três pousos, apresentando alguns territórios investigados, as relações estabelecidas com nosso objeto de pesquisa e os movimentos que observamos.

### **Produção e construção de dados**

O estudo se propõe a cartografar o processo de construção do pensamento computacional em uma experiência com estudantes do 2º ano das séries iniciais em uma escola de ensino fundamental da rede pública municipal.

Os relatos das observações descritas e analisadas neste trabalho se referem a algumas aulas desenvolvidas durante o ano letivo de 2017 no Laboratório de Informática na Educação (LIE), em uma escola de Ensino Fundamental da rede pública municipal. Durante as aulas de Informática Educativa, que aconteciam semanalmente em períodos de cinquenta minutos, foram observadas duas turmas de estudantes do 2º ano do Ensino Fundamental em uma escola da rede pública municipal, totalizando 50 crianças. Os grupos eram mistos, com número equilibrado entre meninos e meninas, com idade entre 7 e 8 anos. A maioria dos estudantes já estava alfabetizada, dominando os processos básicos de leitura e escrita, sendo que apenas cerca de 10% se encontrava em processo de alfabetização. Entre os dois grupos havia três estudantes com deficiência, sendo um acompanhado por cuidador.

Essas aulas foram orientadas por uma das autoras que era a professora responsável pelas atividades no LIE no turno da tarde e este estudo foca em um total de 12 encontros. Foram dois encontros usando o robô programável Bee-Bot®, sete encontros nas trilhas da “Hora do Código” e três de programação em “Scratch”. Desses encontros, três contaram com a presença adicional de uma pesquisadora (e também autora deste trabalho) atuando como observadora participante.

O planejamento dos encontros visava à exploração de programação com um brinquedo de programar (BeeBot®), com atividades disponíveis propostas pelo projeto “Hora do Código” e



com o software livre Scratch. Tendo em vista o objetivo de observar as manifestações do pensamento computacional, a opção por essas atividades baseou-se no fato de estarem fundamentadas em princípios de programação semelhantes aos propostos por Papert (1988), segundo os quais, a criança ao usar os comandos de programação, desenvolve a compreensão de como os objetos se movem no espaço.

Com vistas à análise dos movimentos identificados ao longo do processo, destacamos a seguir os momentos em que registramos uma mudança de escala em nossa atenção a qual se focou em novas janelas atencionais conhecidas em cartografia como “pousos”. E como os pousos são geralmente seguidos de outra variedade atencional que é o reconhecimento atento, tecemos nosso texto de modo a apresentar simultaneamente os dados produzidos por nossa atenção e a análise que fazemos deles à luz dos conceitos que norteiam nosso estudo.

### Três pousos

Os pousos apresentados a seguir se referem a situações que despertaram a atenção das pesquisadoras por serem momentos relevantes e reveladores de indícios do Pensamento Computacional. Ao varrer o território pesquisado, mantivemos nossa atenção em pistas que pudessem apontar as capacidades mencionadas por Wing (2006), como pensar em diferentes níveis de abstração, resolver problemas e a habilidade de pensar além do mundo físico e construir mundos virtuais.

Entre as atividades realizadas com os grupos, destacamos aqui aquelas que evidenciam o desenvolvimento do Pensamento Computacional por meio do uso do robô programável (Bee-Bot®), das trilhas da “Hora do Código” e da programação em “Scratch”.

### Pouso 1: brincando com um robô

Foram realizadas duas aulas com o robô programável (Bee-Bot®) com cada grupo. No primeiro contato, as crianças demonstraram um grande entusiasmo e afeto pelo robô, batizando-o de “bee”, por se tratar de uma abelha e por estabelecerem relação com o filme “Bee Movie”.

Vale lembrar que essa reação de afetividade das crianças para com o objeto de estudo e com o desafio proposto tem um papel importante no funcionamento da inteligência, conforme Piaget (1962). O autor reconhece ainda que fatores como interesse, a necessidade de encontrar respostas ou soluções para suas perguntas ou problemas só existem porque há afeto. Ele argumenta, porém, que isso não é suficiente para a constituição da inteligência, a qual está relacionada ao desenvolvimento da estrutura cognitiva, que, por sua vez, se desenvolve independente do afeto.





Foi difícil controlar a emoção e esperar a vez de ‘brincar’, pois só havia um robô para cada grupo, com 25 crianças cada. As pesquisadoras apresentaram o robô e conversaram sobre as hipóteses iniciais, ouvindo o que os estudantes já sabiam e como imaginavam que a “bee” iria se movimentar. Foram organizadas regras com as crianças para que todos tivessem a oportunidade de manusear o robô e pudessem contribuir com suas ideias. A atividade foi realizada com as crianças sentadas em um círculo, com o tabuleiro e o Bee-Bot® no centro, sendo que cada estudante tinha o direito de propor e executar um movimento com o robô. O colega a seguir na roda deveria dar sequência ao percurso, prosseguindo ou corrigindo a rota.

O primeiro desafio consistiu em levar o robô de um ponto a outro sobre um tabuleiro, uma grade quadriculada desenhada em papel pardo. Foram distribuídos alguns obstáculos pelo percurso para criar diferentes níveis de dificuldade no caminho. O primeiro movimento foi descobrir os principais comandos representados por setas (para a frente, para trás, virar à esquerda, virar à direita). O segundo foi corrigir os erros, refazendo os caminhos ou recolocando o robô na rota certa, sempre utilizando apenas os comandos. Houve algumas tentativas de recolocar a “bee” no caminho certo utilizando-se a mão para isso, mas o grupo não aceitou, já que essa era uma regra que havia sido estabelecida pelas pesquisadoras no início do desafio.

O reconhecimento atento dessa atitude das crianças em relação aos seus erros nos permite entendê-la como parte do que Papert (1988) chama de processo de *debugging*. Ou seja, em vez de perceber o que não funcionou para solucionar o problema apresentado pelo desafio como erro, isso é visto como “uma parte integrante do processo de compreensão do programa. O programador é encorajado a estudar o *bug* ao invés de esquecê-lo.” (p. 85). Entender o que aconteceu e porque determinado comando não funcionou conforme o necessário irá contribuir para que as crianças explorem as diversas maneiras de corrigir o que aconteceu. Dessa forma, a criança percebe que é possível haver progressos e que nem o certo nem o errado são coisas absolutas, mas sim indicadores do que está melhor e do que ainda pode ser melhorado.

O segundo desafio, que era levar o robô de volta refazendo o caminho inicial, foi muito difícil. As crianças não conseguiram conservar o percurso nem os comandos utilizados. Essa atitude é compreensível tendo em vista a faixa etária das crianças, que ainda não dominam a reversibilidade de pensamento. Essa capacidade, segundo Becker (1998), permite que o sujeito consiga executar a mesma ação nos dois sentidos, ou seja fazer e desfazer, ou no caso do presente estudo, refazer o percurso do robô ao contrário. O autor afirma que essa estrutura operatória de pensamento pode ser observada a partir do estágio operatório concreto no qual a criança consegue coordenar duas variáveis, consciente de que a ação nos dois sentidos é a mesma ação. No entanto,



ela ainda não consegue criar uma sistematização combinatória, ou seja, consegue variar apenas um dos fatores e conserva os demais.

Isso pôde ser observado nas tentativas que as crianças fizeram para movimentar a “bee”, usando apenas a variável do trajeto que o robô teria que percorrer para programar o retorno ao ponto de origem, sem, porém, considerar a posição em que o corpo do robô se encontrava, com as costas voltadas para o ponto de partida e o nariz da abelha apontando para o ponto de chegada.

O segundo encontro com a “bee” aconteceu mais de um mês depois, pois o robô Bee-Bot® era emprestado. As crianças estavam com saudades dela, perguntando insistentemente quando ela retornaria. O desafio proposto consistiu em deslocar o robô sobre um tabuleiro quadriculado em papel pardo com diversos desenhos, representando obstáculos a ser evitados ou objetos a ser conquistados: flores, maçãs, mel, inseticida, chuva, fogo e teia de aranha. As crianças foram provocadas a classificar os desenhos, criando categorias quanto àquilo que era bom ou prejudicial para a abelha. A partir da categorização já foram sendo criados critérios para definir o percurso, escolhendo rotas para evitar o que prejudicaria a abelha, como o inseticida e a teia de aranha, bem como para alcançar o mel e as flores.

Antes de iniciar a realização do desafio, foram retomadas as regras construídas no primeiro encontro com o Bee-Bot® e os comandos para movimentá-lo. As crianças demonstraram que haviam conservado as informações, aplicando-as novamente com facilidade, apesar do período de tempo transcorrido. Na execução do desafio, os estudantes se mostraram mais confiantes e ousados em suas tentativas/experimentações. Apesar da regra de que cada um poderia usar apenas um comando por vez, diversas crianças tentaram combinar mais de um comando para fazer o robô chegar ao destino. Na maioria das vezes, as tentativas foram frustradas, cabendo ao colega a seguir consertar a sequência e retomar o percurso da “bee”. Em diversos momentos a turma se manifestava, tentando alertar o colega que a sequência estava errada, porém poucos consideravam as intervenções, mantendo suas hipóteses iniciais.

## Pouso 2: explorando as trilhas do conhecimento

A Hora do Código (<https://hourofcode.com/br>) é uma plataforma online de jogos digitais baseados na programação visual, com tutoriais e cursos que visam ao ensino de programação. De acordo com o site oficial, a Hora do Código “começou como uma introdução de uma hora à ciência da computação, criada para desmistificar a programação e mostrar que qualquer pessoa pode aprender os fundamentos básicos e ampliar a participação na área desta ciência.” (Hora do Código, 2017). As atividades e tutoriais foram apresentados aos estudantes como jogos, por seu



caráter lúdico. O principal objetivo é movimentar um personagem pela tela, até resolver o desafio proposto, arrastando e soltando blocos de programação. Os jogos são autoinstrucionais, com diversos tutoriais disponíveis para aqueles que tiverem alguma dúvida ou dificuldade. Na medida em que o jogador vai avançando, os desafios vão se complexificando, aumentando a necessidade de domínio das habilidades trabalhadas para poder avançar.

Os estudantes tiveram acesso ao site da Hora do Código em sete aulas, em momentos alternados com outras atividades. Uma característica das duas turmas é a facilidade com que perdem o interesse pelos projetos realizados. Tendo em vista essa situação, a professora/pesquisadora decidiu não concentrar todas as atividades em um mesmo período, alternando com outros projetos envolvendo outros temas e softwares.

Nas aulas 1, 2 e 3, os estudantes jogaram a trilha “Aventureiro do Minecraft”, pois tinham uma fascinação pelo jogo Minecraft. A maioria das crianças não tinha acesso ao game em suas residências, mas todos já haviam jogado o Minetest, um software livre e similar disponível no Laboratório de Informática.

Na aula 1, os estudantes escolheram suas duplas. A única restrição foi de que aqueles que ainda não dominavam a leitura tinham que escolher um colega que já soubesse ler, pois todos os desafios são descritos de forma escrita. O primeiro desafio foi encontrar o jogo usando a ferramenta de pesquisa do Google®. Foi interessante observar quais duplas leram a descrição antes de abrir o link e quais foram clicando até encontrar o site. Outro aspecto foi a colaboração dos que acessaram primeiro para com os demais, pois prontamente saíram ajudando os colegas a encontrar o jogo. Como o jogo é autoinstrucional, a professora não fez nenhuma intervenção inicial, e deixou que as crianças descobrissem como jogar. A principal vantagem é de que cada dupla pode avançar no seu ritmo.

O jogo “Aventureiro do Minecraft” tem 14 fases, sendo que começa explorando os blocos de movimento (para frente, para trás, virar à direita, virar à esquerda), e vai avançando progressivamente até chegar aos blocos de ação (tosar, plantar, etc.), de repetição e condicionamento (função SE). Cada fase apresenta um novo desafio: chegar até a ovelha, tosar a ovelha, etc. Na medida em que vão avançando, os estudantes consolidam algumas habilidades e são desafiados a fazer novas descobertas.

Ao começar o jogo, as duplas foram experimentando suas hipóteses. Alguns leram o desafio e o resolveram rapidamente, outros observaram a dupla vizinha e copiaram suas ações, mas a maioria foi pela tentativa e erro, tentando combinar os blocos de forma aleatória e executando as ações sem sucesso. Foi possível observar que as crianças que costumam jogar games em casa



tentavam, inicialmente, usar a mesma lógica a que estão acostumados (passar de fase), o que não funcionou. Observou-se que as crianças que se destacaram estavam mais concentradas, lendo os desafios e usando as dicas e tutoriais disponíveis na Hora do Código, criando estratégias para conduzir o personagem e atingir o objetivo. O que chama atenção aqui é que aqueles que jogam games regularmente têm domínio dos periféricos, manuseando o mouse e o teclado com muita facilidade, trocando de tela e buscando soluções, mas têm uma grande ansiedade em vencer e passar de fase.

A postura que as crianças assumiram durante o jogo reflete a conduta que costumam ter nas aulas de Informática Educativa, pois são autônomos e colaborativos. As duplas conversavam o tempo todo, buscando estratégias e opinando, sendo extremamente cooperativos, auxiliando prontamente os colegas. É mais comum pedir ajuda para a dupla lado, perguntando “como vocês fizeram?”, do que copiar o que o vizinho fez, demonstrando a vontade de aprender. As crianças transitam livremente pela sala, levantando para ver como uma dupla fez para superar o desafio ou indo até o computador da outra dupla para “consertar” a programação.

O grande destaque foram duas duplas que, entre 50 crianças, conseguiram concluir as 14 fases, em um período de 50 minutos sem nunca terem acessado o jogo antes. Vale ressaltar que o desafio é planejado para ser resolvido em uma hora e nossa atenção é definitivamente afetada ao lembrar que os estudantes do 5º ano, na mesma escola, não conseguiram esse feito em um período de aula. A primeira dupla a concluir era composta por duas meninas, que passaram a aula muito concentradas e perguntando sempre que tiveram dificuldades, não perdendo tempo e com grande desejo de aprender. A segunda dupla era mista, sendo que o menino é um grande jogador de games em casa. Ele fez uso das habilidades que já tinha para descobrir os comandos usando, principalmente, a tentativa e erro, tentando mais do que pedindo ajuda, e conduzindo sua colega na vez dela jogar, dizendo muito “faz assim”.

Nas aulas 2 e 3 os estudantes apresentaram condutas muito semelhantes. O destaque foi que a professora orientou as duplas para partirem da fase em que estavam na aula anterior, porém muitas duplas quiseram burlar a regra e avançaram. Neste momento ficou evidente a necessidade de dominar as habilidades anteriores, pois as fases foram ficando muito difíceis e as crianças não conseguiam avançar. Como o “Aventureiro do Minecraft” é um jogo que tem a clara intenção de apresentar os principais comandos e ensinar os primeiros passos na área da programação, se os comandos ainda não foram aprendidos não é possível avançar. As crianças se deram conta e acabaram retornando, voluntariamente, para as fases anteriores.

As duas duplas que já haviam finalizado avançaram para o segundo jogo “Designer de



Minecraft”, com a mesma proposta de desafios, só que com grau de dificuldade maior. Na segunda aula nenhuma dupla conseguiu finalizar este jogo. As demais duplas foram, gradativamente, finalizando o primeiro jogo e acessando o segundo, porém, muitos estudantes acharam o jogo muito difícil e pediram para voltar ao primeiro. A grande diferença entre os jogos é que no segundo havia a combinação de diferentes blocos, reunindo mais de um atributo por vez (movimento, repetição e ação, por exemplo).

Na segunda rodada de atividades com a “Hora do Código”, nas aulas 4 e 5, os estudantes jogaram Kodable, cujo objetivo é levar bolinhas coloridas por percursos determinados em cada rodada. O jogo é dividido em 4 fases, com 5 níveis cada uma. Cada fase trabalha com um conjunto de conhecimentos de programação: sequência, condição (função SE), looping e combinação de funções. Não é possível avançar sem ter concluído o desafio proposto, diferente do “Aventureiro do Minecraft” em que as crianças podiam avançar ou retornar.

Na primeira fase, que explora as sequências de movimento, os estudantes tiveram muita facilidade, demonstrando que já reconheciam os comandos. Apesar de não dominarem a nomenclatura (esquerda, direita), as crianças demonstraram uma boa noção de lateralidade e de movimento no espaço, levando os personagens pelo caminho correto. Quando as crianças não conseguiam, a professora intervinha, provocando com questionamentos que faziam a criança pensar no percurso a ser percorrido: “me mostra com o dedo o caminho que você quer fazer?”. E após a demonstração da criança: “qual é a seta que é igual a esse caminho?”. Depois de algumas rodadas usando o dedo, as crianças conseguiam abstrair os movimentos e usar as setas de orientação corretamente.

O reconhecimento atento para entender o tipo de conduta adotado pela professora encontra respaldo em Papert (1988) tendo em vista a importância de propor que a criança utilize também o corpo e se movimente para aprender. De acordo com ele, isso “mobiliza a experiência e o prazer com o movimento. Toda essa experiência faz uso de um campo de conhecimento bem familiar à criança, a ‘geometria do corpo’ (grifo do autor), um ponto de partida para o desenvolvimento de conexões com a geometria formal” (PAPERT, 1988, p. 81). Ao descrever essa “geometria do corpo”, Papert está se referindo ao conhecimento que a criança adquire ao aprender a andar e a se deslocar no espaço e esse conhecimento estaria agora sendo aplicado ao objeto que está programando com comandos de deslocamento.

Similarmente, Bannel et al (2011) contestam o pressuposto de que, para aprender, a criança precisa permanecer em silêncio, sentada e concentrada durante um longo período. Os autores ressaltam que a construção de conhecimento não pode estar relacionada ao bloqueio da percepção



e que a aprendizagem ocorre em um movimento de alternância entre atenção e dispersão. Assim, enquanto se movimentam e exploram suas representações corporais, as crianças estão descobrindo que há mais de um jeito de aprender.

Na segunda fase, entre os comandos, havia a necessidade de usar a condição SE, isto é, se o personagem batesse em um quadrado colorido, deveria mudar a rota. O que mais chamou a atenção nesse momento foi que as primeiras crianças a descobrirem o que deveriam fazer saíram pela sala ensinando os outros. E depois de entendido o processo inicial, nenhuma dupla teve dificuldade, mesmo nas rodadas mais difíceis deste nível. Apesar de ser um conceito complexo, o condicionamento foi assimilado de uma forma tranquila, possivelmente devido às experiências anteriores ou pela complexificação gradativa proposta no jogo.

Podemos pensar que a forma como as crianças resolveram a dificuldade que estavam enfrentando com o comando está relacionada com o fato de haver uma intenção cuidadosamente projetada de adicionar uma nova expressividade a materiais e práticas do cotidiano da criança aos recursos propostos para ensiná-la a programar. Assim, ao utilizar os comandos de movimento de corpo - “para frente”, “para trás” - ou comandos de comportamento utilizados em robótica - com o uso da condição “se”, a criança desenvolve habilidades de abstração, de uso de linguagem, noções de movimento corporal no espaço e estratégias para solucionar problemas.

Na terceira e na quarta fases os desafios começaram a ficar mais complexos, pois exigiam o domínio das funções de repetição e agrupamento de comandos. Assim como aconteceu com o jogo anterior, a maioria das crianças não conseguiu avançar e acabou desistindo. No entanto, como aqui não era possível retornar e mudar de fase, foram dedicadas apenas duas aulas para o jogo.

O destaque no jogo “Kodable” foi um dos estudantes com Deficiência Intelectual. Apesar de ainda não ter consolidado o processo de alfabetização, ele se encantou e quis jogar sozinho, pois apresenta baixa tolerância na convivência com os demais. Ele avançou mais do que todos os colegas, chegando nas fases 3 e 4. O estudante conseguiu usar as funções de repetição e agrupamento, sempre usando tentativa e erro. As poucas vezes em que pediu ajuda foi para a professora e, apesar de descrever todas as suas ações, ele não foi capaz de ajudar os colegas.

Percebemos que, quando a pesquisadora que estava presente como observadora-participante solicitava que as crianças descrevessem para ela o que estavam fazendo, elas descreviam as atividades de programação como algo lúdico. Isso está presente, por exemplo, na resposta do menino citado no parágrafo anterior. Quando lhe foi perguntado o que estava fazendo, sua resposta foi que estava “brincando com o jogo das bolinhas coloridas”, ou “jogando Star Wars”.



Na verdade, o que ele descreve como jogo são algumas trilhas de programação propostas no site da “Hora do Código”.

Essa situação ilustra uma ideia altamente difundida por Papert (1988), na qual o poder da tecnologia reside nas possibilidades de expressão pessoal, de empoderamento das crianças e de interação humana que se abrem. O uso criativo, exploratório e lúdico da tecnologia por parte das crianças que aprendem enquanto brincam pode ser, segundo o autor, uma ferramenta emancipatória com a qual é possível pensar em uma educação construcionista, baseada em projetos e centrada no aluno.

Na terceira rodada de atividades com a “Hora do Código”, nas aulas 6, 7 e 8, os estudantes puderam escolher livremente um jogo. Na aula 6, eles exploraram todas as opções, abrindo os jogos aleatoriamente, descobrindo como funcionavam e trocando logo em seguida, pois tinham uma grande ânsia em conhecer todos que fossem possíveis. Nas aulas 7 e 8 as crianças escolheram seus preferidos, e se concentraram para realizar os desafios. A maioria das duplas permanecia nos jogos até chegar a uma fase considerada difícil, demonstrando pouca persistência, assim como muitos escolheram os jogos já conhecidos. Os jogos mais populares nessas aulas foram os que tinham o mesmo nível de dificuldade dos que já conheciam como “Star Wars”, “Moama” e “Angry Birds”.

O principal destaque foi para a dupla que terminou por primeiro o jogo do “Aventureiro do Minecraft”, pois foram as únicas estudantes que não desistiram no jogo “Frozen”, avançando até o final, explorando e descobrindo novos conhecimentos, envolvendo conteúdos que não são comuns a sua faixa etária, como ângulos. Ao ter que construir uma estrela de dez pontas, as meninas tiveram que identificar as medidas dos ângulos. No primeiro momento, trabalharam com base no princípio de tentativa e erro, até descobrir que havia um transferidor digital entre as ferramentas disponíveis que poderia ajudá-las. Neste momento, chamaram a professora que as fez perceber que a abertura do transferidor deveria ter a mesma medida que a do losango. Elas usaram os dedos para ver se a abertura era “maior” ou “menor” do que estavam imaginando.

### Pouso 3: experimentando possibilidades

No final do ano, os estudantes tiveram três aulas para conhecer e criar utilizando o Scratch, um ambiente de programação visual desenvolvido a partir da linguagem Logo, baseado em blocos. As crianças já estavam acostumadas aos principais comandos, pois eles se assemelham aos utilizados na “Hora do Código”, especialmente ao arrastar e soltar.

A primeira aula foi exploratória e os estudantes tiveram 50 minutos para descobrir as



principais ferramentas. Sempre em duplas, eles puderam demonstrar todos os conhecimentos de programação construídos até então: conseguiram movimentar o personagem; inserir objetos; trocar fundo; criar objetos e fundos; inserir sons e caixas de texto. A postura de compartilhar as aprendizagens se manteve, mas o destaque foi para a curiosidade dos colegas que queriam saber “como você fez?”, e depois de observar e ouvir as explicações, voltavam para seu computador e tentavam fazer também.

Nas segunda e terceira aulas, o desafio proposto pela professora foi criar uma animação. Pela primeira vez a professora deu uma explicação sobre os principais comandos, pois havia uma quantidade muito maior de opções do que as que estavam acostumados: blocos de controle, movimento, aparência, som, sensores, canetas, operações e variáveis. Enquanto a professora falava, as crianças acompanhavam na tela do seu computador e podiam experimentar. A ênfase dada foi para os comandos de controle, movimento e aparência, mas as crianças ficaram curiosas com todos.

As animações criadas foram todas muito simples, mas extremamente significativas e importantes para as crianças, que fizeram questão de apresentar os resultados finais para a turma. A maioria das duplas fez o gato (personagem padrão) se mexer. Algumas descobriram como mudar a aparência do gato, e trocaram o personagem. Poucas duplas conseguiram fazer mais de um personagem mexer, combinando movimentos. E apenas uma dupla conseguiu inserir diálogo escrito na animação (um fala e o outro responde).

É visível que, na primeira aula, quando as crianças exploraram os comandos aleatoriamente, foram mais curiosas. Já no segundo e terceiro encontros, usaram menos recursos. Acreditamos que eles possam ter sido mais seletivos no uso dos comandos com a intenção de realizar o desafio (criar uma animação). Outra possibilidade, que tem relação com as observações realizadas anteriormente, é sobre a dificuldade de trabalhar com mais de um atributo, diretamente relacionada com o nível de abstração característico do estágio de desenvolvimento cognitivo em que se encontram.

Essa situação ilustra o que Bannell et al (2016) afirmam sobre a importância de uma aprendizagem prazerosa em que se aprende fazendo e descobrindo novas coisas. Para os autores, “o fator que mais influencia a aprendizagem é o conhecimento que o aprendiz já possui.” (p.109). Assim, percebe-se que suas experiências anteriores com o robô programável e com as trilhas de programação da “Hora do Código” contribuíram para aprenderem a “pensar computacionalmente” (grifo nosso). Com o pensamento computacional, conforme Blikstein (2008), é possível identificar quais as tarefas cognitivas que poderão ser realizadas eficientemente





pelo computador.

Por mais simples que a produção dos estudantes possa parecer ao nosso olhar, as habilidades digitais desenvolvidas permitiram que pudessem pensar sobre os problemas que precisavam ser resolvidos e se sentiram confiantes para arriscar a criar coisas novas. E a aprendizagem, segundo Papert (2008), não está em aplicar regras memorizadas para resolver problemas, mas em “brincar com os problemas” (grifo do autor) e isso ajudará o sujeito a desenvolver as habilidades e competências necessárias para encontrar as soluções. E a sala de aula que abre espaço para o estudante dialogar e compartilhar diferentes saberes e experiências, na opinião de Bannell et al (2016), é uma sala que contribui para a construção do pensamento crítico e que permite ao estudante perceber que pode mudar o que quer realizar, como tentar realizar e, mais importante, acreditar que seja possível realizar.

### **Considerações**

Ao analisar a construção do pensamento computacional nos anos iniciais do Ensino Fundamental, com o intuito de verificar e identificar que indícios de construção do pensamento computacional podem ser percebidos enquanto as crianças foram desafiadas a resolver problemas com a programação, identificamos algumas possibilidades que se abrem para seu desenvolvimento cognitivo e construção do processo de abstração. Por meio da cartografia pudemos acompanhar os processos e marcar alguns caminhos e movimentos, mapeando dados, para construir análises possíveis baseadas em nossos objetivos e referenciais teóricos.

Acreditamos que a aprendizagem autônoma, tendo os estudantes como protagonistas, é uma grande contribuição da programação para a educação. Ao buscar soluções para os seus problemas, as crianças criam estratégias que se aplicam em outras situações da vida. A postura proativa frente aos erros, quando percebem que uma hipótese não deu certo e exploram outras maneiras de corrigir o que aconteceu, reforça as ideias de Papert (1988), que apresenta o ‘erro’ como parte do processo de compreensão.

A possibilidade de pensar em diferentes níveis de abstração, proposta por Wing (2006), também é evidenciada na pesquisa. Os avanços e descobertas relatados no texto demonstram quanto conhecimento uma criança de 7 anos pode construir interagindo no ambiente virtual. No início, as hipóteses são testadas em ações de tentativa e erro, mas à medida em que os desafios avançam, a capacidade de abstração vai sendo evidenciada, pois é necessário conservar as aprendizagens anteriores para poder avançar. Em alguns casos a criança desiste e volta para uma fase anterior, demonstrando que ainda não desenvolveu as habilidades necessárias para aquele



desafio. Em outros, ela acolhe o desafio e busca novas alternativas, demonstrando que já consolidou os conhecimentos necessários e tem as estruturas necessárias para avançar.

Ao se deparar com conteúdos que ainda não conhece, como no caso dos ângulos do losango, os estudantes tiveram a possibilidade de experimentar uma outra forma de aprender: pela tentativa e erro, usando novas ferramentas e testando hipóteses até compreender como fazer. Essa experiência nos faz pensar em Lévy (2000) e na necessidade de superar o que ele classifica como “adestramento educacional”. No lugar de aulas expositivas sobre ângulos, por exemplo, os estudantes podem construir esse conceito interagindo com softwares e aplicativos, de forma lúdica e construtiva. É urgente o incentivo ao desenvolvimento de práticas pedagógicas que deixem para trás a mesmice da aula centrada no professor e que estimulem o pensamento crítico e possibilitem a construção do conhecimento.

Seguindo essa linha, chegamos ao que Wing (2006) e Blikstein (2008) propõem como construção de mundos virtuais e construção de novos conhecimentos. Ao criar uma animação com Scratch, os estudantes fizeram uso das aprendizagens que construíram ao longo do processo. O fato de terem desenvolvido diversas habilidades digitais interagindo com o robô programável Bee-Bot® e as trilhas do conhecimento da Hora do Código, abriu a possibilidade de criar virtualmente. Aplicar o conhecimento construído demonstrou quais habilidades digitais relacionadas à fluência digital e tecnológica cada criança desenvolveu, corroborando com Papert (1988, 2008) e suas ideias de “pensar como um computador” e “ensinar o computador a pensar”.

O objetivo de introduzir tecnologias digitais nas escolas não pode ser visto apenas como uma alternativa para melhorar a educação tradicional ou, como se vê na mídia, jornalística, “para deixar as aulas mais atrativas” (grifo nosso). É preciso resgatar Papert (1988) e vê-las como um recurso emancipatório que coloca ao alcance das crianças materiais de construção altamente valiosos. Em nossa análise percebemos que o computador pode ser uma máquina para a criança pensar sobre o pensar, compreender os usos complexos da tecnologia e aprender a programar, deixando de lado a passividade de ser mera consumidora de produtos tecnológicos e passa a ser construtora ativa com o uso das tecnologias.

## Referências

ALMEIDA, Maria I; PIMENTA, Selma G. **Estágios supervisionados na formação docente**. São Paulo: Cortez, 2014.

BARREIRO, Iraide Marques de Freitas; GEBRAN, Raimunda Abou. **Prática de Ensino e Estágio Supervisionado na formação de Professores**. São Paulo: Avercamp, 2006.



BRASIL. Conselho Nacional de Educação. **Parecer CNE/CP 28, de 02 de outubro de 2001. Dá nova redação ao Parecer CNE/CP 21/2001, que estabelece a duração e a carga horária dos cursos de Formação de Professores da Educação Básica, em nível superior, curso de licenciatura, de graduação plena.** Pareceres CNE/CP, Brasília, 2001b. Disponível em: [http://portal.mec.gov.br/dmdocuments/cnecp\\_212001.pdf](http://portal.mec.gov.br/dmdocuments/cnecp_212001.pdf). Acesso em: 13 jun. 2019.

BRASIL. Conselho Nacional de Educação. **Resolução CNE/CP 1, de 18 de fevereiro de 2002. Institui Diretrizes Curriculares Nacionais para a Formação de Professores da Educação Básica, em nível superior, curso de licenciatura, de graduação plena.** Resoluções CNE/CP, Brasília, 2002 a. Disponível em: [http://portal.mec.gov.br/seesp/arquivos/pdf/res1\\_2.pdf](http://portal.mec.gov.br/seesp/arquivos/pdf/res1_2.pdf). Acesso em: 14 jun. 2019.

BRASIL. Conselho Nacional de Educação. **Resolução CNE/CP n. 02/2015, de 1º de julho de 2015. Define as Diretrizes Curriculares Nacionais para a formação inicial em nível superior (cursos de licenciatura, cursos de formação pedagógica para graduados e cursos de segunda licenciatura) e para a formação continuada.** Diário Oficial da União, Brasília, DF, 2 jul. 2015, p. 8-12. Seção 1

LIMA, Maria Socorro Lucena. **Estágio e Aprendizagem da Profissão Docente.** Brasília: Liber Livro, 2012.

PIMENTA, Selma G. **A Didática como mediação na construção da identidade do professor: uma experiência de ensino e pesquisa.** In: ANDRÉ, M. e OLIVEIRA, M. R.(orgs.). Alternativas do Ensino de Didática. Campinas: Papirus, 1997.

PIMENTA, S. G; Lima, M. S. L. **Estágio e docência.** São Paulo: Cortez Editora, 2004.

SCALABRIN, I. C.; MOLINARI, A. M. C. **A importância da prática do estágio supervisionado nas licenciaturas.** UNAR, v. 17, n. 1, 2013.

**Artigo recebido em: 23/08/2020.**

**Artigo aceito em: 31/08/2020.**