

Aprendiendo a resolver problemas en los primeros años de la Educación Básica: relaciones entre el desarrollo del pensamiento computacional y el aprendizaje de las matemáticas.

Luciane Mulazani dos Santos¹

luciane.mulazani@udesc.br

<https://orcid.org/0000-0001-7617-7310>

Katiane Cugik Couto¹

kcugik@yahoo.com.br

<https://orcid.org/0000-0001-9681-8744>

¹*Universidade do Estado de Santa Catarina (UDESC, Brasil)*

Recibido: 30/05/2020 **Aceptado:** 26/07/2020

Resumen

Este artículo presenta una discusión de acciones que se llevaron a cabo como parte de una investigación de maestría profesional del Programa de Posgrado en Enseñanza de Ciencias, Matemáticas y Tecnologías (PPGECMT) de la Universidad Estadual de Santa Catarina (UDESC) orientada al desarrollo del pensamiento computacional de los estudiantes en los primeros años de la escuela primaria. Mediante procedimientos metodológicos de investigación cualitativa, el estudio tuvo como objetivo desarrollar un producto educativo, en el formato de un blog de Internet, dirigido a docentes. El resultado que se presenta aquí es, por tanto, un recorte de la investigación. Muestra que es posible insertar actividades didácticas de programación para el desarrollo del pensamiento computacional desde los primeros años de Educación Básica y que trabajar con el pensamiento computacional ayuda al aprendizaje resolviendo problemas de tal manera que este puede contribuir de manera positiva en el aprendizaje de contenidos matemáticos. La investigación nos hizo coincidir con profesores e investigadores que creen que las relaciones que se establecen en la escuela se modifican en la época contemporánea debido a los recursos tecnológicos digitales que tenemos acceso para utilizar en diferentes situaciones de nuestra vida diaria.

Palabras clave: Pensamiento computacional; Enseñanza de programación; Solución de problemas; Producto educativo.

Aprender a resolver problemas nos anos iniciais da Educação Básica: relações entre o desenvolvimento do pensamento computacional e a aprendizagem de matemática.

Resumo

Este artigo apresenta uma discussão das ações que foram realizadas no âmbito de uma pesquisa de mestrado profissional do Programa de Pós-Graduação em Ensino de Ciências, Matemática e Tecnologias (PPGECMT) da Universidade do Estado de Santa Catarina (UDESC) voltadas ao desenvolvimento do pensamento computacional dos alunos nos primeiros anos do ensino fundamental. Utilizando procedimentos metodológicos da pesquisa qualitativa, o estudo teve como objetivo o desenvolvimento de um produto educacional, no formato de um blog de internet, voltado aos professores. A etapa aqui discutida, como recorte da pesquisa, teve como foco a realização de uma oficina de

programação para crianças do segundo ano do Ensino Fundamental. Os resultados evidenciaram que é possível a inserção de atividades de ensino de programação para desenvolvimento do pensamento computacional desde os primeiros anos da Educação Básica e que o trabalho com o pensamento computacional ajuda a aprendizagem por meio de resolução de problemas de tal forma que isso pode contribuir de forma positiva na aprendizagem de conteúdos de matemática. A pesquisa nos fez concordar com professores e pesquisadores que acreditam que as relações estabelecidas na escola se veem modificadas na contemporaneidade em virtude dos recursos tecnológicos digitais aos quais temos acesso para uso em diferentes situações de nosso cotidiano.

Palavras-chave: Pensamento computacional; Ensino de programação; Resolução de problemas; Produto educacional.

Learning to solve problems in the early years of Basic Education: relationships between the development of computational thinking and the learning of mathematics.

Abstract

This article presents a discussion of actions that were carried out as part of a professional master's research from the Graduate Program in Teaching of Sciences, Mathematics and Technologies (PPGECMT) at Santa Catarina State University (UDESC) aimed at the development of computational thinking of students in the first years of elementary school. Using methodological procedures of qualitative research, the study aimed to develop an educational product, in the format of an internet blog, aimed at teachers. The result presented here is, therefore, a clipping of the research. It shows that it is possible to insert programming teaching activities for the development of computational thinking since the early years of Basic Education and that working with computational thinking helps learning by solving problems in such a way that this can contribute positive way in learning math content. The research made us agree with teachers and researchers who believe that the relations established in the school are modified in the contemporary times due to the digital technological resources that we have access to use in different situations of our daily lives.

Keywords: Computational thinking; Programming teaching; Problem solving; Educational product.

Introdução

De uma forma geral, pensamento computacional é aquele que entra em cena quando resolvemos algum problema com o uso de certas ferramentas relacionadas à área da Computação. Quanto mais a tecnologia digital avança e nos permite acessar novos modos de viver, mais o pensamento computacional nos ajuda em tarefas cotidianas, escolares ou profissionais, seja por nossa ação ou por desfrutarmos de soluções construídas por ações de outras pessoas. Nesse contexto, o desenvolvimento do pensamento computacional é tema cada vez mais estudado por pesquisadores tanto da área da Computação quanto da Educação. Isso, porque percebeu-se que ensinar a pensar dessa

forma, já desde os primeiros anos escolares, pode ajudar os estudantes tanto na aprendizagem dos conteúdos do currículo quanto na interação com diferentes situações vividas da sociedade contemporânea. É um cenário que faz sentido no âmbito educacional porque o desenvolvimento do pensamento computacional nos ajuda a resolver problemas dos mais diversos tipos em diferentes áreas, o que vale para a escola e vale para a vida.

Nosso grupo de pesquisa, o Núcleo de Estudo e Pesquisa em Tecnologia Educacional e Educação Matemática (NEPESTEEM), certificado junto ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), estuda temas relacionados ao desenvolvimento do pensamento educacional na Educação Básica há cerca de seis anos. De lá para cá, pesquisas de iniciação científica, de conclusão de curso de graduação, de mestrado e de doutorado se dedicam a diferentes perspectivas de ação e de investigação. Neste artigo, apresentamos uma discussão que foi construída sobre resultados de uma dessas pesquisas, que teve como objetivo o desenvolvimento de um produto educacional no curso de Mestrado Profissional do Programa de Pós-Graduação em Educação (PPGECMT) da Universidade do Estado de Santa Catarina (UDESC)¹. Nessa pesquisa, houve uma articulação entre o desenvolvimento do pensamento computacional e a tendência de resolução de problemas, no âmbito da Educação Matemática. Nas seções seguintes, apresentamos tal estudo e promovemos uma discussão sobre o foco escolhido para este artigo: a realização de uma oficina de programação com crianças do segundo ano do Ensino Fundamental.

Temas da pesquisa: pensamento computacional e resolução de problemas

Pensamento computacional

Abrimos este tópico com uma citação de Valente (1999, p. 32) que afirmou, à época, que a educação “se adequa às concepções paradigmáticas que vive a sociedade. Portanto, ela passa pelas mesmas transformações que outros segmentos da sociedade passam”. Fazemos isso para evidenciar que as relações entre educação escolar e tecnologia digital são estudadas desde longa data e sob as mais diversas perspectivas. Uma dessas perspectivas aborda o desenvolvimento do pensamento computacional.

¹ Dissertação resultante da pesquisa:

https://sucupira.capes.gov.br/sucupira/public/consultas/coleta/trabalhoConclusao/viewTrabalhoConclusao.xhtml?popup=true&id_trabalho=6400489

Produto educacional resultante da pesquisa: <http://educapes.capes.gov.br/handle/capes/429203>

Para resgate da origem dos estudos das relações entre pensamento computacional e processos de ensino e aprendizagem, nós nos apoiamos em Valente (1995, 1999, 2013, 2016). Há duas décadas e meia, Valente (1995) colocou em perspectiva, para a educação, aspectos positivos decorrentes de ações que proporcionam aos estudantes a utilização de computadores para buscar informações, processá-las e utilizá-las na resolução de problemas. Segundo o autor, esse movimento os ajuda a construir seus próprios conhecimentos e melhor compreenderem aquilo que fazem. No final da década de 90, Valente defendeu que a inserção da tecnologia digital na educação não poderia ser feita para “mera informatização do atual processo” e, sim, para “verdadeira revolução do processo de aprendizagem” (Valente, 1999, p. 47). Esse cenário se relaciona com as mudanças da natureza do conhecimento na sociedade que experimenta a tecnologia digital: o conhecimento não mais se associa à memorização de informações e sim a saber buscar, selecionar e usar essas informações. (Valente, 2013). As questões contemporâneas ligadas ao desenvolvimento do pensamento computacional atendem a essas demandas ao possibilitarem a criação de conhecimento.

Ainda como resgate, lembramos dos trabalhos de Seymour Papert, da década de 1960, com o desenvolvimento da linguagem LOGO, o que se tornou uma possibilidade para inserção da programação de computadores em ações pedagógicas ligadas à educação escolar básica, ou seja, para iniciativas de desenvolvimento do pensamento computacional.

Um marco do debate sobre o pensamento computacional e sua relação com o ensino e aprendizagem é derivado dos estudos de Wing (2006). Segundo a autora, pensamento computacional é a capacidade de desencadear um processo de formulação de um problema e de o solucionar, uma habilidade tão fundamental de ser desenvolvida nos estudantes quanto a leitura, a escrita e as operações aritméticas. Essa é uma abordagem fundamentada em processos da área da Ciência da Computação, pois envolvem organização e análise de dados, criação de modelos, simulação, construção de algoritmos e automatização de soluções para resolver de problemas. Desde então, diferentes organizações fomentam estudos e realizam ações voltadas ao desenvolvimento do pensamento educacional de estudantes dos níveis básicos de ensino. É o caso, por exemplo, das internacionais *Computer Science Teachers Association* (CSTA),

International Society for Technology in Education (ISTE) e *National Science Foundation (NSF)* e do Centro de Inovação para a Educação Brasileira (CIEB)².

O consórcio CSTA, ISTE e NSF (2011) propôs um conjunto de ferramentas, que chamaram *Computational Thinking Toolkit* (kit de ferramentas para pensamento computacional), que envolve nove fases:

- (i) Coleta de dados. Como reunir os dados de maneira apropriada?
- (ii) Análise de dados. Como dar sentido aos dados, encontrar padrões entre eles e estabelecer conclusões?
- (iii) Representação de dados. Como representar os dados e a análise?
- (iv) Decomposição de problemas. Como fragmentar o problema em partes menores para facilitar o processo de busca de soluções?
- (v) Abstração. Como focar no objetivo principal para reduzir a complexidade do problema?
- (vi) Algoritmos. Qual a melhor sequência de passos que pode resolver o problema?
- (vii) Automação. Que recursos computacionais utilizar para executar as tarefas?
- (viii) Paralelismo. Como organizar os recursos que podem desenvolver simultaneamente as tarefas para encontrar a solução?
- (ix) Simulação. Como representar ou modelar o processo e a sua execução?

Vemos, assim, que essas fases estão relacionadas à compreensão e à resolução de um problema por meio de coleta e análise dos dados, fragmentação e desenvolvimento de passos para resolvê-lo, em acordo com as definições de pensamento computacional conforme Wing (2006).

Valente (2016, p. 9) apresentou caracterizações do processo de criação de um programa de computador para a resolver um problema as quais “têm sido úteis para explicitar as atividades que o aprendiz realiza na interação com as tecnologias digitais e ajudam a entender como a interação com as tecnologias digitais contribuem para o desenvolvimento do pensamento computacional”. O pesquisador definiu o ciclo de ações executadas em uma atividade de programação e o chamou de DERDN: Descrição, Execução, Reflexão, Depuração e Nova descrição. Segundo ele, o desenvolvimento de um programa começa com a ideia da pessoa para resolver um problema; depois, essa pessoa

² <https://cieb.net.br/>

desenvolve uma solução que é passada para um programa específico, na forma de uma sequência de comandos, o que corresponde à ação Descrição; na sequência, a ação Execução corresponde ao momento em que o programa escolhido executa os comandos solicitados e mostra o resultado à pessoa que, então, observa esse resultado e, na ação Reflexão, verifica se ele é o que esperava; se for o esperado, considera aquele problema como resolvido, se não for, realiza a Depuração, quando analisa o código e busca identificar onde estão as eventuais falhas; esse ciclo se reinicia sempre que necessário, partindo da ação Nova descrição.

Percebem-se aproximações das características do ciclo DERDN com as fases definidas pelo consórcio CSTA, ISTE e NSF (2011) e com as definições de Wing (2006). A elas, podemos acrescentar as definições do CIEB, que indicam quatro ações fundamentais envolvidas no pensamento computacional que têm importante papel no processo de resolver um problema:

(i) Decompor: é dividir um problema em partes, ou seja, desmembrar um problema grande em problemas menores. Importa fazer isso porque, ao encontrarmos as soluções para os problemas menores – que, em tese, seriam mais fáceis de resolver – essas podem ser utilizadas na resolução do problema maior. Podemos buscar na matemática um exemplo de decomposição: a resolução da soma $1648 + 251$ pode ser decomposta: $1648 = 1000 + 600 + 40 + 8$ e $251 = 200 + 50 + 1$. A decomposição em ordens e classes ajuda a encontrar a solução para o problema, pois somam-se parcelas menores, mais fáceis de serem resolvidas: $1648 + 251 = (1000) + (600 + 200) + (40 + 50) + (8 + 1)$. Ou seja: $1648 + 251 = 1000 + 800 + 90 + 9 = 1899$.

(ii) Abstrair: é pensar em ações semelhantes que podem ser realizadas em contextos diferentes, porque tais contextos guardam aproximações entre si. Por exemplo, se eu sei como se faz um bolo de chocolate, é possível que eu abstraia a ação “fazer um bolo” para conseguir fazer, também, um bolo de laranja. Respeitando as diferenças entre os ingredientes, o processo, em si, é semelhante, ou seja, há uma lógica comum no ato de fazer um bolo de chocolate e no ato de fazer um bolo de laranja.

(iii) Reconhecer padrões: é, a partir da observação de certas situações, perceber a existência de algo, nelas, que permite prever o que acontecerá em situações futuras ou em contextos semelhantes. Por exemplo, se eu percebo que a conexão de internet via *wifi* da minha casa fica mais lenta todos os dias de uma determinada semana às seis horas da

tarde, posso entender isso como um padrão e supor que esse fato também se dará em outras semanas, ao longo do mês.

(iv) Criar algoritmos: é representar um determinado processo ou atividade por meio de uma sequência de passos lógicos finitos com o objetivo de resolver um problema. Essa representação pode ser feita, por exemplo, com a escrita de códigos utilizando determinada linguagem de programação. Para exemplificar uma situação que pode ser resolvida por meio de um algoritmo, deixamos aqui uma pergunta ao leitor: qual a melhor maneira de ir do seu quarto até à cozinha de sua casa? Essa pergunta representa a questão central da discussão que queremos debater neste artigo: as estratégias desenvolvidas para criação de algoritmos para resolução de problemas, ou seja, as ações de desenvolvimento do pensamento computacional.

A respeito de possíveis modos de realizar atividades de programação, há estudos e práticas que mostram que eles podem se dar como computação plugada ou desplugada. (Bell et al., 2011). Os conceitos de computação plugada e desplugada se referem à utilização (plugada) ou não utilização (desplugada) de tecnologia digital. Isso nos mostra que é possível trabalhar com o ensino de programação e com o pensamento computacional mesmo sem a utilização da tecnologia digital.

As definições aqui apresentadas embasaram a concepção, o planejamento, a proposta e a aplicação do produto educacional desenvolvido no mestrado profissional.

Resolução de problemas

Na pesquisa de Couto (2018), a ponte entre tecnologia educacional e Educação Matemática se construiu pela tendência da resolução de problemas. Nas abordagens da pesquisa e do produto educacional, tratamos do desenvolvimento do pensamento computacional como um passo importante para ensinar as crianças a resolverem problemas. A resolução de problemas como tendência em Educação Matemática abre um espaço de discussão sobre o lugar dos problemas na aprendizagem da matemática escolar. Isso porque, dada às características dessa área, diz-se, de uma maneira ou de outra, que os problemas conectam a matemática com a realidade e que resolver problemas, por meio da investigação, serve não somente para aprender os conteúdos escolares, mas também para inserir-se criticamente no mundo. Nesse contexto, há mais de uma definição para problemas e mais de uma compreensão sobre o que é e como se faz para encontrar a solução para um problema.

Para a pesquisa, escolhemos as abordagens desde Onuchic (1999) a Onuchic e Allevato (2014). As pesquisadoras definem problema como tudo aquilo que não se sabe fazer, mas que se está interessado em resolver. Além disso, nos dizem que a resolução de problemas é uma abordagem metodológica que contribui para o ensino e a aprendizagem escolares em todos os níveis de ensino. Essas ideias foram importantes para caracterizar como problemas as atividades propostas para desenvolvimento do pensamento computacional dos estudantes participantes da pesquisa, ou seja, como situações que eles se interessassem em resolver propondo soluções. Além disso, para propor problemas cuja processo de resolução – por meio de algoritmos – lhes permitissem aprender diferentes conceitos, não somente da área de Computação.

Cabe ressaltar que a discussão sobre a resolução de problemas como tendência em Educação Matemática é mais ampla do que a forma aqui apresentada em razão das limitações de um texto deste formato. Porém, tal síntese cabe adequadamente na discussão proposta neste artigo.

Evidenciamos, assim, as referências que fundamentaram as conclusões da pesquisa a respeito das possibilidades de ensinar crianças a resolverem problemas por meio de algoritmos em atividades de computação plugada e desplugada, mediadas pelo professor. No tópico seguinte, apresentamos o desenvolvimento do produto educacional.

Decisões e ações: a proposta e a implementação de um produto educacional voltado a professores da Educação Básica.

O produto educacional

Um curso de mestrado profissional na área de ensino possibilita a formação acadêmica com o compromisso de desenvolvimento de um produto educacional (que pode ter diferentes formatos, tais como sequência didática, jogo, software, vídeo, website etc.), conforme determinações³ da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior Conselho (CAPES), entidade que regula a pós-graduação *stricto sensu* no Brasil. Nesse contexto, o PPGECMT da UDESC espera que seus/as mestrandos/as e orientadores/as se envolvam em estudos que resultem na produção de um produto educacional e de um trabalho final no formato de dissertação, como este que aqui apresentamos: o desenvolvimento de um produto educacional no formato de um blog

³ Portaria Normativa nº 17, de 28 de dezembro de 2009 - https://www.capes.gov.br/images/stories/download/legislacao/PortariaNormativa_17MP.pdf

publicado na internet, voltado a professores, sobre conteúdos relacionados ao desenvolvimento do pensamento computacional dos estudantes. Couto (2018) justifica essa escolha, em sua dissertação, ao dizer que:

Uma das possibilidades de construção de novas práticas docentes chega às salas de aula e também aos espaços não formais de educação com as experiências que tratam da inserção do ensino de programação no currículo escolar da Educação Básica. **Programar tem a potencialidade de dar ao aluno um papel ativo em seu processo de aprendizagem:** para além de ser usuário de um software, ele passa a usar a tecnologia para produzir seus próprios conteúdos digitais, tais como jogos e animações. (Couto, 2018, p. 15, grifos nossos).

Como parte dos procedimentos metodológicos do desenvolvimento do produto educacional, foi realizada uma revisão de literatura com o objetivo de identificar um panorama das pesquisas sobre o ensino de programação na Educação Básica realizadas no Brasil no período entre 2012 e 2015, uma vez que o estudo foi realizado no segundo semestre de 2016 e quis investigar os quatro anos anteriores. Utilizou-se o método proposto por Kitchenham (2004). A seguir, apresentamos uma síntese⁴ do método e dos resultados encontrados.

A revisão de literatura foi realizada em três fases: planejamento; execução; análise e divulgação dos resultados. (Kitchenham, 2004). Foi norteadada pela pergunta “como ensinar programação na Educação Básica?” e pelos termos de busca: “programação”, “Educação Básica” e “ensino”. As bases de dados on-line consultadas pela internet foram: anais do Simpósio Brasileiro de Informática na Educação (SBIE) e do Workshop de Informática na Escola (WIE); periódicos Revista Brasileira de Informática na Educação (RBIE) e Revista Nova Tecnologias na Educação (RENOTE).

A coleta de dados aconteceu em três etapas: pré-seleção; seleção; tabulação e análise dos resultados. Na pré-seleção, foram encontradas 101 publicações. Na seleção, após aplicação dos critérios de inclusão e de exclusão, restaram 26 publicações, sendo que a maioria das descartadas ou se referiam ao Ensino Superior ou não se relacionavam ao uso de programação. Depois de analisar as 26 publicações, verificou-se que uma mesma pesquisa apareceu em três diferentes conjuntos de dados, o que levou à decisão de excluir os dois textos mais antigos. Assim, o corpus de análise foi formado por 24 textos.

Após análise dessas publicações, concluímos que 54% das pesquisas eram voltadas ao Ensino Médio; 29% ao Ensino Fundamental II (6º ao 9º ano); 13% ao Ensino

⁴ Descrição completa da revisão de literatura pode ser encontrada no texto da dissertação.

Fundamental I (4º e 5º anos) e 4% aos anos iniciais do Ensino Fundamental I (1º ao 3º ano). Notadamente, encontramos uma evidência da existência de poucos estudos voltados ao Ensino Fundamental I, particularmente aos anos iniciais. O Scratch foi a plataforma de programação mais citada. Em mais de 50% dos casos, o ensino de programação não estava vinculado a disciplinas ou a conteúdos do currículo escolar e sim ao ensino de estruturas de linguagens de programação e ao desenvolvimento do raciocínio lógico, por meio da criação de jogos e animações. Ainda, a maioria dos trabalhos apontou que as atividades, quando práticas, aconteceram extraclasse, no contraturno escolar. Quanto à vinculação dos relatos a universidades públicas, 96% dos trabalhos encontraram-se nessa situação. Quanto à localização em que se deram os estudos, do total analisado, 10 trabalhos foram oriundos da região Nordeste do Brasil, 7 da região Sudeste, 6 da região Sul e 1 da região Norte.

A partir dos dados coletados e analisados na revisão de literatura, conhecemos um cenário das pesquisas sobre o ensino de programação na Educação Básica realizadas no Brasil entre 2012 e 2016, relatadas em eventos científicos e em periódicos eletrônicos. Concluímos que, nesse contexto, há carência de pesquisas sobre o uso da programação como recurso de ensino e aprendizagem nos anos iniciais do Ensino Fundamental (1º ao 3º ano). Vimos, assim, relevância na proposta do produto educacional que estava em desenvolvimento no mestrado profissional para abertura de um caminho de investigação e de desenvolvimento de práticas na área. A análise desse cenário contribuiu para a realização dos demais procedimentos de desenvolvimento do produto educacional. Dentre eles, o planejamento de uma oficina, cuja realização foi fundamental para organização do conteúdo do blog.

O Blog Programador do Futuro foi desenvolvido na plataforma Blogger⁵, gratuita e disponível na internet. Foi publicado no endereço <http://programadordofuturo.blogspot.com>. Seu conteúdo foi organizado em três seções: Despluga Aí!, Code.org e Scratch, sendo que a primeira reúne propostas de atividades para computação desplugada e as duas outras para computação plugada.

“Despluga Aí!” apresenta uma sequência didática para desenvolvimento do pensamento computacional por meio da computação desplugada, baseada no livro *Computer Science Unplugged* (Bell et al., 2011). As atividades são voltadas a crianças da faixa etária de 7 a 10 anos. Ao navegar nessa área, os professores têm acesso aos roteiros

⁵ <https://www.blogger.com>

completos de todas as atividades propostas, com objetivos, conjunto de materiais e de mídias, sugestões de aplicação.

A seção Code.org apresenta propostas de utilização de algumas atividades dessa plataforma, que é mantida por uma organização sem fins lucrativos⁶, presente em mais de 180 países, criada com o objetivo de expandir o ensino da ciência da computação para crianças e jovens. As atividades de programação realizadas na Code.org não requerem a escrita de códigos de programação porque a interação é visual, por meio de blocos manipulados em modo arrastar-soltar. Na plataforma, o professor pode criar uma turma, cadastrar seus alunos para realizarem as atividades e acompanhar suas interações e desempenhos no curso proposto.

Na seção Scratch, são apresentadas propostas de programação que podem ser realizadas com essa plataforma, com tutoriais para professores e para estudantes, bem como sugestões de atividades. Assim como acontece na Code.org, a programação na plataforma Scratch⁷ é feita por blocos, sem necessidade de escrever os códigos de programação e de forma lúdica, o que a torna uma interessante opção para trabalhar com as crianças.

O próximo tópico é dedicado à descrição, apresentação e análise da oficina que ajudou no processo de desenvolvimento do produto educacional

Uma oficina de programação para estudantes do segundo ano do Ensino Fundamental: resolução de problemas com desenvolvimento do pensamento computacional.

Como parte das ações realizadas para desenvolvimento do Blog Programador do Futuro, a oficina de programação com estudantes contribuiu para escolha e preparação das atividades que foram publicadas, ou seja, para pensar nos limites e possibilidades do produto educacional.

A oficina foi realizada com uma turma de estudantes do segundo ano do Ensino Fundamental de uma escola do Município de Itajaí/SC e ofereceu atividades de aprendizagem de conteúdos de programação. Foi organizada e ministrada por Katiane, planejada como uma das etapas de sua pesquisa no mestrado profissional. Envolveu atividades de resolução de problemas com o uso de algoritmos de programação, as quais

⁶ <https://code.org>

⁷ <https://scratch.mit.edu>

nos auxiliaram a sintetizar a discussão que é tema deste artigo, a qual nos dedicamos a partir de agora.

Participaram da oficina onze estudantes, na faixa etária dos 7 e 8 anos de idade, no período de 01 de setembro a 30 de novembro de 2017, nas dependências da escola, no horário regular das aulas, com a presença da professora regente da turma. A participação foi autorizada pelos/as responsáveis pelos estudantes, declarada pela assinatura de Termos de Consentimento Livre e Esclarecido (TCLE) e de autorizações para registro de fotografias, vídeos e gravações; os originais desses documentos ficaram sob guarda da pesquisadora. As atividades foram realizadas em 13 horas/aula, divididas em duas etapas: a primeira, para atividades de programação desplugada e a segunda para atividades de programação utilizando o computador (plugadas). A abordagem escolhida para ambas as etapas foi a resolução de problemas. Levou-se em conta a faixa etária das crianças e, por isso, houve a preocupação com a utilização de abordagem lúdica e que levasse em consideração que elas estavam em fase de alfabetização.

As atividades desplugadas, ou seja, sem a utilização de recursos digitais, foram realizadas na sala de aula em 8 horas/aula. As atividades plugadas, ou seja, de programação pelo computador foram realizadas na sala de informática da escola em 5 horas/aula, em dois momentos: 2 horas/aula para estudo do ambiente Code.org e 3 horas/aula para estudo do ambiente Scratch.

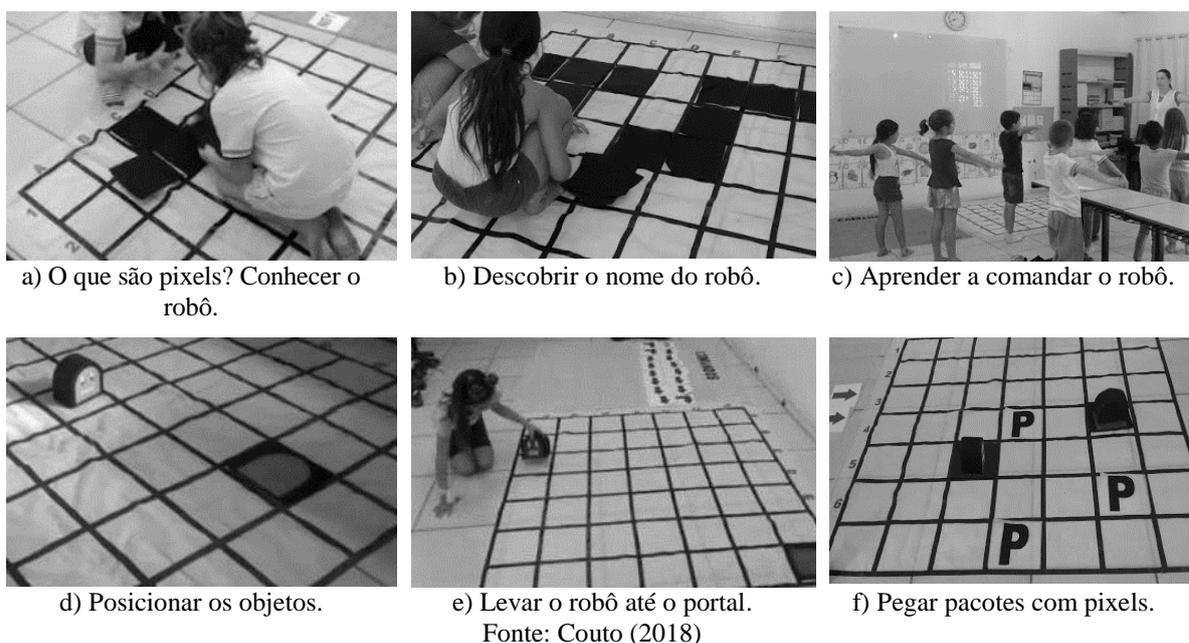
A participação das crianças foi registrada em vídeo e fotos e suas produções foram salvas nas plataformas Code.org e Scratch. Ao final de cada encontro, a pesquisadora anotou suas observações em um diário de bordo e, no final da oficina, por meio de um questionário, coletou a opinião das crianças relativa à sua participação. Desse modo, o corpus de análise da pesquisa foi formado pelos registros das manifestações orais e escritas das crianças, pelas respostas dadas por elas ao questionário, pelas anotações da pesquisadora a respeito da participação, envolvimento e produção das crianças na oficina. Além de observar esses dados para desenvolvimento do produto educacional, a partir deles foi possível identificar elementos do pensamento computacional – decomposição, abstração, reconhecimento de padrões e criação de algoritmos – e formas de utilização do Ciclo DERDN nas estratégias implementadas pelos estudantes para resolução dos problemas propostos.

Sobre as atividades desplugadas

Foram preparadas e aplicadas a partir da adaptação do livro *Computer Science Unplugged* de Bell et al.. Sua realização e análise gerou o conteúdo que foi publicado na seção Despluga Aí! do blog Programador do Futuro. Esta parte da oficina foi dividida em dois momentos:

Atividades em grupo – primeiro momento

Figura 1: Atividades desplugadas da oficina de programação



Todos os materiais utilizados nesta etapa foram produzidos pela pesquisadora, com recursos não digitais. Eles podem ser consultados e reproduzidos a partir do blog Programador do Futuro ou do relatório do produto educacional⁸. As crianças foram organizadas em três equipes. Cada equipe foi representada por uma cor. Para dividir o grupo, a pesquisadora entregou, aleatoriamente, uma pulseira colorida a cada criança para representar a cor da equipe da qual faria parte. A Figura 1 mostra as etapas das atividades, que são descritas na sequência.

a) O que são pixels? Conhecer o robô.

Um pixel é a menor unidade que compõe uma imagem digital, ou seja, um conjunto de pixels forma uma imagem inteira. Seu nome pode ser traduzido para “elemento de imagem”, pois é a combinação das palavras *picture* e *element*. Fotos digitais,

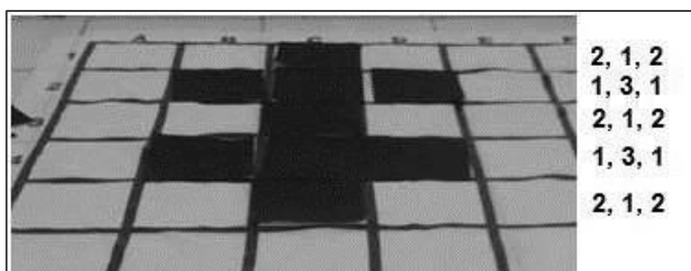
⁸ <http://educapes.capes.gov.br/handle/capes/429203>

telas de smartphone, monitores e TVs são formadas por pixels. Em imagens digitais, conseguimos visualizá-los, na forma de quadradinhos, quando utilizamos os recursos de aproximação (zoom). É um termo que se tornou comum por se ligar a diferentes recursos tecnológicos que utilizamos no dia a dia. Portanto, conhecer sobre eles ajuda as crianças a compreenderem o uso que fazem da tecnologia.

Na oficina, as crianças aprenderam que as telas digitais são formadas por uma grade dividida em pixels. Para experimentarem esse conceito, realizaram a atividade de formação de uma imagem com peças de quadrados brancos e pretos em um grande tapete quadriculado, como se fosse uma tela digital. Cada casa do tapete quadriculado foi representada pela combinação de uma letra com um número, sendo que a letra indicava a posição da coluna e o número indicava a posição da linha, de modo semelhante a um jogo de batalha naval, em alusão a um plano de coordenadas cartesianas.

Para resolverem o problema, as crianças foram desafiadas por um robô que lhes enviou um enigma em áudio (gravado pela pesquisadora como se fosse a voz do robô) recitando sequências numéricas que deveriam ser representadas, uma em cada linha, no tapete quadriculado, utilizando os quadrados brancos e pretos. O segredo, explicado pela pesquisadora, estava na ordem de utilização dos quadrados: na sequência de três números, o primeiro e o terceiro número representavam a quantidade de quadrados brancos enquanto o segundo número representava a quantidade de quadrados pretos. A Figura 1a mostra como as crianças trabalharam nessa etapa. Cada equipe recebeu um tapete, um conjunto de quadrados e a sequência numérica e manipulou esses materiais no chão da sala. A solução do problema foi a representação da imagem de um carrinho, o que é mostrado na Figura 2, que apresenta a produção feita por uma das equipes.

Figura 2: Solução de uma equipe e sequência numérica da atividade de representação de pixels



Fonte: Couto (2018)

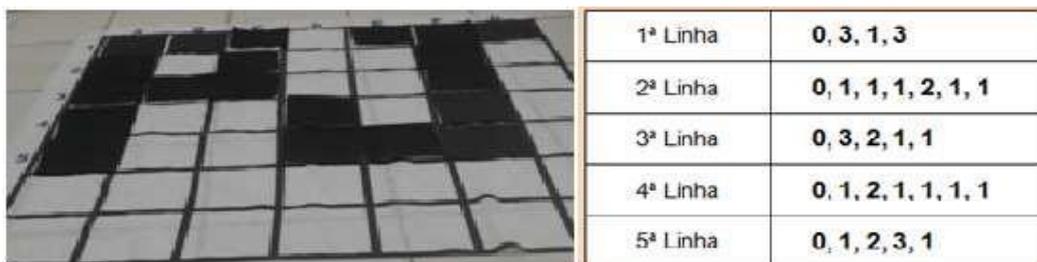
A respeito dos objetivos da oficina, nessa atividade foi possível trabalhar com as etapas de coleta, análise e representação de dados do processo de desenvolvimento do pensamento computacional. Além disso, a aplicação das sequências numéricas preparou as

crianças para a compreensão do trabalho com algoritmos que foi feito em momentos posteriores. Com relação à matemática, a discussão sobre o significado de uma sequência numérica e seu papel na solução do problema permitiu o trabalho com as habilidades de contagem, localização e compreensão de padrões. A respeito da dinâmica, percebeu-se o envolvimento das crianças, de modo colaborativo, para resolução dos problemas com discussão sobre hipóteses e estratégias.

b) Descobrir o nome do robô.

Depois que as crianças experimentaram a forma de representação de dados no tapete quadriculado pela aplicação de sequência numérica (aplicação de um algoritmo), o desafio proposto na segunda atividade foi utilizar sequências numéricas maiores para descobrir o nome do robô. A solução de uma das equipes, mostrando que o nome do robô era PJ, é apresentada na Figura 3.

Figura 3: Solução de uma equipe e sequência numérica da atividade de descobrir o nome do robô



1ª Linha	0, 3, 1, 3
2ª Linha	0, 1, 1, 1, 2, 1, 1
3ª Linha	0, 3, 2, 1, 1
4ª Linha	0, 1, 2, 1, 1, 1, 1
5ª Linha	0, 1, 2, 3, 1

Fonte: Couto (2018)

Durante a atividade, a pesquisadora percebeu que o uso do recurso de áudio para informar as sequências maiores às crianças dificultou o seu entendimento sobre o problema. Para contornar essa situação, escreveu as sequências no quadro e, posteriormente, no blog, alterou a proposta da atividade, sugerindo que as sequências sejam entregues a elas no formato impresso. Concluiu que essa mudança também seria importante no processo de leitura dos números pelas crianças, já que elas estavam em processo de alfabetização.

Os objetivos atingidos na atividade anterior também foram alcançados na segunda atividade. Além disso, ao analisar a presença das etapas do ciclo DERDN, a pesquisadora percebeu quando os estudantes ouviram as instruções (Descrição), representaram a codificação no tapete de malha quadriculada (Execução) e analisaram a imagem formada e constaram que tinham solucionado o desafio de construir a imagem do carrinho e do nome do robô (Reflexão).

c) Criar algoritmos

Quando apresentamos os fundamentos do estudo, no início deste artigo, dissemos que um algoritmo é uma sequência finita de passos destinados a resolver determinado problema e que sua criação exige a escolha de estratégias voltadas à solução desejada. Vimos, também, que quando falamos do processo de desenvolvimento do pensamento computacional, a criação do algoritmo é uma etapa fundamental. Na terceira atividade desse momento da oficina, o objetivo foi envolver as crianças em desafios que envolveram a criação de algoritmos para resolução de problemas.

Os desafios? Seguindo as orientações recebidas, conduzir o robô PJ pelo tapete quadriculado, sob três diferentes condições, até que ele chegasse a um portal. Antes de iniciarem o trabalho nas equipes para solução dos problemas, as crianças participaram de uma dinâmica, ilustrada na Figura 1c, para compreenderem formas de movimentação no espaço, girando o corpo para moverem-se para a direita, para a esquerda ou para seguirem em frente. De forma intuitiva, compreenderam os sentidos de girar 90° para a direita e girar 90° para a esquerda, necessários na atividade, mesmo sem conhecimentos matemáticos formais prévios a respeito de ângulos.

A primeira orientação que as crianças receberam, também por um arquivo de áudio que simulava a voz do robô, informava a posição (linha/coluna) que robô, obstáculos e portal deveriam ser colocados no tapete quadriculado. Receberam peças que representaram cada um desses itens e os posicionaram, como pode ser observado nas figuras 1d, 1e e 1f. Depois disso, os três desafios foram propostos:

- Primeiro desafio – sem obstáculos. As equipes aplicaram as orientações recebidas para levarem o robô PJ de sua posição inicial até a posição em que estava o portal. Ou seja, criaram um algoritmo para resolver o problema. A apresentação desses algoritmos foi feita em tapetes coloridos – um de cada equipe – com a combinação de setas que representavam os movimentos de girar 90° para a direita, girar 90° para a esquerda ou seguir em frente.

Figura 4: Problema de guiar o robô sem obstáculos



Fonte: Couto (2018)

A Figura 4 mostra as soluções S1, S2 e S3, propostas pelas equipes. Analisadas, vimos que os algoritmos de S1 e S3 foram iguais (seguir seis casas em frente, virar 90° para a esquerda e seguir seis casas em frente) e que em S2, a equipe optou por utilizar 3 comandos de virar 90° à direita (equivalente à rotação de 270° para a direita) em vez de um único comando de rotação de 90° para a esquerda. Em matemática, o resultado é o mesmo, mas em se tratando de algoritmos, significou a inclusão de dois passos a mais no processo. Essa é uma situação que pode disparar uma discussão a respeito da equivalência entre ângulos e sobre como isso evidencia a possibilidade de soluções diferentes para um mesmo problema. Foram dois diferentes algoritmos, sendo que ambos resolveram o problema porque cumpriram o objetivo proposto.

Nesta atividade, ao lidarem com comandos e algoritmos, as crianças passaram por todas as etapas do desenvolvimento do pensamento computacional e percorreram o Ciclo DERDN, bem como decompueram os problemas, abstraíram, reconheceram padrões e representaram algoritmos. O trabalho colaborativo no processo de resolução do problema, com discussão entre os membros dentro de suas equipes e entre as equipes, foi algo que nos chamou a atenção por sublinhar como esse tipo de atividade mobiliza diferentes conhecimentos e habilidades. Com relação à matemática, notamos como a atividade potencializou a compreensão de conceitos que envolvem localização no espaço, ângulos, equivalência, além da contagem. Além disso, fato importante foi discutir com os estudantes como um problema pode ter mais de uma solução correta e que, em certos casos, a diferença entre elas proporciona que se atinjam os resultados de maneira mais eficiente ou rápida. Essas considerações se estendem às análises que fizemos sobre os dois desafios seguintes.

- Segundo desafio – com obstáculos. Aqui, também o robô PJ deveria ser levado até o portal. Porém, com uma condição: ele não poderia passar por cima dos obstáculos

(pixels pretos colocados no tapete). A figura 5 mostra as soluções apresentadas pelas equipes.

Figura 5: Problema de guiar o robô com obstáculos

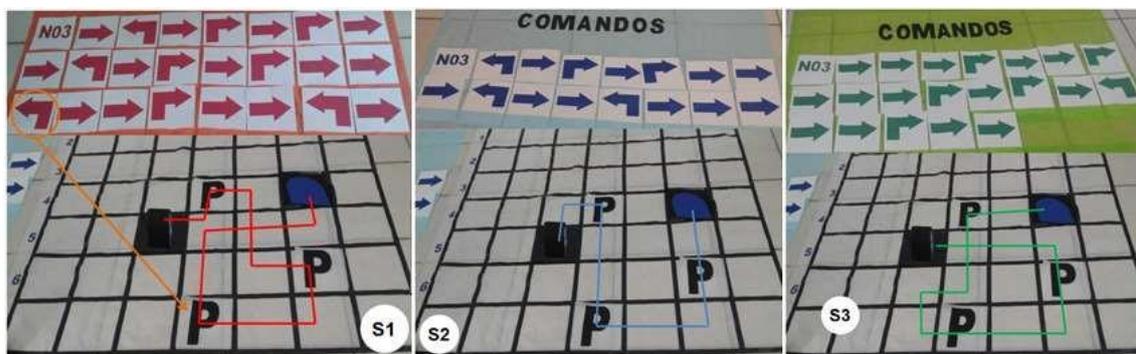


Fonte: Couto (2018)

Cada equipe apresentou um algoritmo diferente como solução, mas todas alcançaram seu objetivo. O algoritmo da solução S3 teve o menor número de comandos. Já S2, apresentou o algoritmo mais longo ao usar quatro comandos a mais do que a solução S3. A solução S1 construiu um algoritmo com 14 comandos, um a mais do que a equipe S3.

- Terceiro desafio – com missão. Desta vez, o robô PJ deveria recolher três pacotes e levá-los consigo ao portal. Esses pacotes foram representados no tapete quadriculado por um pixel identificado com a letra P. Para recolher esses pacotes, bastava que o robô passasse pelas casas que os continham, ou seja, ao contrário do desafio anterior, havia a condição de passar por uma determinada posição. A figura 6 mostra as soluções apresentadas pelas equipes, todas diferentes entre si.

Figura 6: problema de guiar o robô com missão



Fonte: Couto (2018)

O algoritmo de S2 teve o menor número de passos em relação à solução S3, mas ambas conseguiram atingir os objetivos. Já a solução S1 teve o algoritmo mais longo e

apresentou um erro na utilização de um dos comandos – que aparece em destaque na Figura 6 – ao indicar a utilização da seta virar 90° à esquerda em vez de virar 90° à direita.

Ao analisar as soluções apresentadas nestes três desafios e a dinâmica da oficina, a pesquisadora percebeu que foi possível trabalhar com as seis etapas do desenvolvimento do pensamento computacional (coleta, análise de dados, representações de dados, decomposições de problemas, abstração e algoritmos) e com as seis etapas do ciclo DERDN (Descrição, Execução, Reflexão, Depuração e Nova descrição). Além disso, houve oportunidades para trabalhar conceitos de matemática, leitura, interpretação e resolução de problemas. Concluiu, também, que desde o início das atividades, houve possibilidade de se trabalhar de forma interdisciplinar o ensino de programação com a leitura, tanto em Língua Portuguesa, quanto da linguagem matemática, bem como concluiu que a organização da oficina em equipes pôs em evidência o trabalho colaborativo e a discussão de ideias para construção de soluções para os problemas e testes de hipóteses, ações essenciais para compreensão dos conceitos envolvidos e para a formação dos estudantes para entenderem o mundo a sua volta e com ele interagirem.

Atividades individuais – segundo momento

As atividades individuais iniciaram desplugadas e depois passaram a ser plugadas, com uso do computador. A pesquisadora as elaborou de modo a contemplar as etapas do processo de desenvolvimento do pensamento computacional e o ciclo DERDN (Valente, 2016).

Cada criança recebeu um roteiro composto por uma sequência de atividades relacionadas a um tema, baseado nas aventuras de um pirata que precisava resgatar um tesouro. Essas atividades compreenderam cinco missões e foram aplicadas em 2 horas/aula. A tarefa da primeira missão foi descobrir o nome do pirata decifrando um código de pixels. Depois, as missões pediram para as crianças programarem o pirata para que ele encontrasse o tesouro; a cada missão, foram incorporados novos elementos que aumentaram o nível de dificuldade da programação envolvida. O objetivo dessa proposta foi propiciar que as crianças abstraíssem e generalizassem as ações que realizaram nas atividades em equipe para apresentarem soluções para novos problemas por meio da criação de algoritmos.

As crianças resolveram os problemas das missões e escreveram suas respostas, por escrito, nas folhas do roteiro. Essas soluções foram analisadas pela pesquisadora, guiada pelos critérios: D (resolução desenvolvida corretamente), PD (resolução apresenta a ideia parcialmente desenvolvida), ED (resolução que apresenta indícios dos conceitos abordados, mas não alcançou a resposta correta), ND (situação em que o estudante tentou resolver, mas a solução não estava correta) e NF (a questão não foi resolvida). O Quadro 1 mostra a lista de atividades propostas e o quantitativo dos critérios alcançados pelos estudantes em cada uma delas.

Quadro 1: Conceitos atribuídos às soluções apresentadas pelas crianças

Atividades propostas	D	PD	ED	ND	NF
Codificação	8	1	0	1	0
Nome do pirata	8	0	0	2	0
Missão 01	8	2	0	0	0
Problema conceito multiplicação	7	1	0	2	0
Missão 2	1	1	5	3	0
Problema conceito divisão	5	1	0	4	0
Missão 3	0	2	3	5	0

Fonte: Couto (2018)

Do Quadro 1, percebemos que todas as crianças apresentaram suas soluções para todos os problemas. Além disso, percebemos que as missões 2 e 3 foram as atividades nas quais os estudantes apresentaram mais dificuldades para apresentação de resoluções corretas. Com relação aos problemas relacionados a conteúdos de matemática, a maioria resolveu corretamente os problemas de multiplicação. A respeito dos problemas de divisão, a pesquisadora constatou que apesar das crianças não terem trabalhado com esse conceito em suas aulas de matemática, metade delas conseguiu encontrar a solução correta ao resolverem os problemas usando os passos da programação, o que evidenciou como aquele trabalho, voltado ao desenvolvimento do pensamento computacional por meio da resolução de problemas, contribuiu para a construção do conhecimento matemático. Ao olhar para as respostas dadas às missões 1, 2 e 3 e lembrar que elas foram propostas com ampliação gradativa do nível de dificuldade, a pesquisadora concluiu que as crianças compreenderam o objetivo de cada missão, porém precisariam de mais discussão e de mais tempo para ampliarem suas habilidades na criação de algoritmos para apresentação de soluções corretas.

Sobre as atividades plugadas

Foram realizadas com uso do computador em duas plataformas: Code.org e Scratch. Nenhuma das crianças participantes da oficina conhecia esses ambientes. A seguir, apresentamos um extrato com algumas discussões sobre as atividades realizadas. Ressaltamos que o estudo ampliado consta na dissertação da qual deriva este artigo.

a) Code.org – no blog

A pesquisadora propôs um roteiro do curso Labirintos Clássicos, oferecido pela plataforma, dividido em 20 níveis. As onze crianças participaram usando os computadores do laboratório da escola, em 2 aulas de 56 minutos, em dois dias distintos. Devido a faltas, três crianças não concluíram as atividades propostas.

No Code.org, as atividades de programação são propostas pelo professor na plataforma digital, onde os estudantes participam e apresentam suas soluções. Também na plataforma é possível, ao professor, acompanhar o trabalho dos estudantes e utilizar os índices ali apresentados em seus critérios de avaliação da aprendizagem.

A figura 7 mostra o registro do progresso das crianças participantes da oficina, feito automaticamente pela plataforma, nas atividades propostas pela pesquisadora. Foi a partir da análise desses registros que ela ponderou a respeito das atividades propostas e realizadas.

Figura 7: Registro do progresso dos estudantes no curso Labirinto Clássico do Code.org



Fonte: Couto (2018)

Vemos, pela figura 7, que os círculos numerados representam as questões do curso Labirinto clássico. Os círculos marcados em verde escuro representam as etapas das atividades que foram concluídas completa e corretamente pelas crianças. Já os círculos marcados em verde claro representam aquelas que as crianças concluíram corretamente, porém utilizaram muitos blocos (comandos). Os amarelos, por sua vez, indicam atividades que estavam em andamento quando esses dados foram extraídos da plataforma; como os círculos brancos indicam etapas não iniciadas, vemos que duas crianças interromperam suas participações nas atividades. Pelos círculos brancos, percebemos a não participação de três crianças.

De todas as análises possíveis a partir dos dados dos progressos das crianças, apresentamos aqui uma discussão da situação da questão 8. Nela, percebemos que nove dentre onze participantes criaram algoritmos com muitos blocos, ou seja, com passos além da quantidade mínima necessária. Essa proporção nos leva a pensar sobre a necessidade de ofertar às crianças oportunidades de testar e refinar seus códigos para criarem estratégias de solução de problemas. Lembrando que são crianças do segundo ano do Ensino Fundamental e que, portanto, é necessária a relativização da avaliação de progresso, ainda assim é uma reflexão válida se compararmos as avaliações da questão 8 com as demais. O problema proposto na questão 8 é mostrado na Figura 8. Vemos que, segundo a avaliação automática feita pela plataforma, o critério que a fez considerar que a questão não recebeu o conceito “concluído, perfeito” foi o fato de que a criança participante não utilizou a estrutura de repetição “repita 4 vezes” e, no lugar, utilizou por quatro vezes um mesmo comando (“avance”). Importa observar que não podemos dizer que as crianças que fizeram isso estavam erradas, uma vez que ambos os algoritmos levam ao mesmo resultado. No âmbito do desenvolvimento do pensamento computacional, temos a ponderar aquilo que significa quando as crianças compreendem as vantagens de se utilizar uma estrutura de repetição em vez de repetir n vezes um mesmo comando; essa questão, no âmbito do ensino de matemática e da resolução de problemas propicia discussões sobre a relação que há entre a operação de adição com a operação de multiplicação. Voltando ao ambiente Code.org e sua interação com os participantes, interessa observar como a plataforma orienta sobre como otimizar um código, situação ilustrada na Figura 8 na fala do personagem.

Figura 8: Exemplo de solução da questão 8 do curso Labirinto Clássico



Fonte: Couto (2018)

A questão 11 teve análise semelhante, desta vez relacionada ao comando “Repita até”, como mostra a Figura 9.

Figura 9: Exemplo de solução da questão 11 do curso Labirinto Clássico



Fonte: Couto (2018)

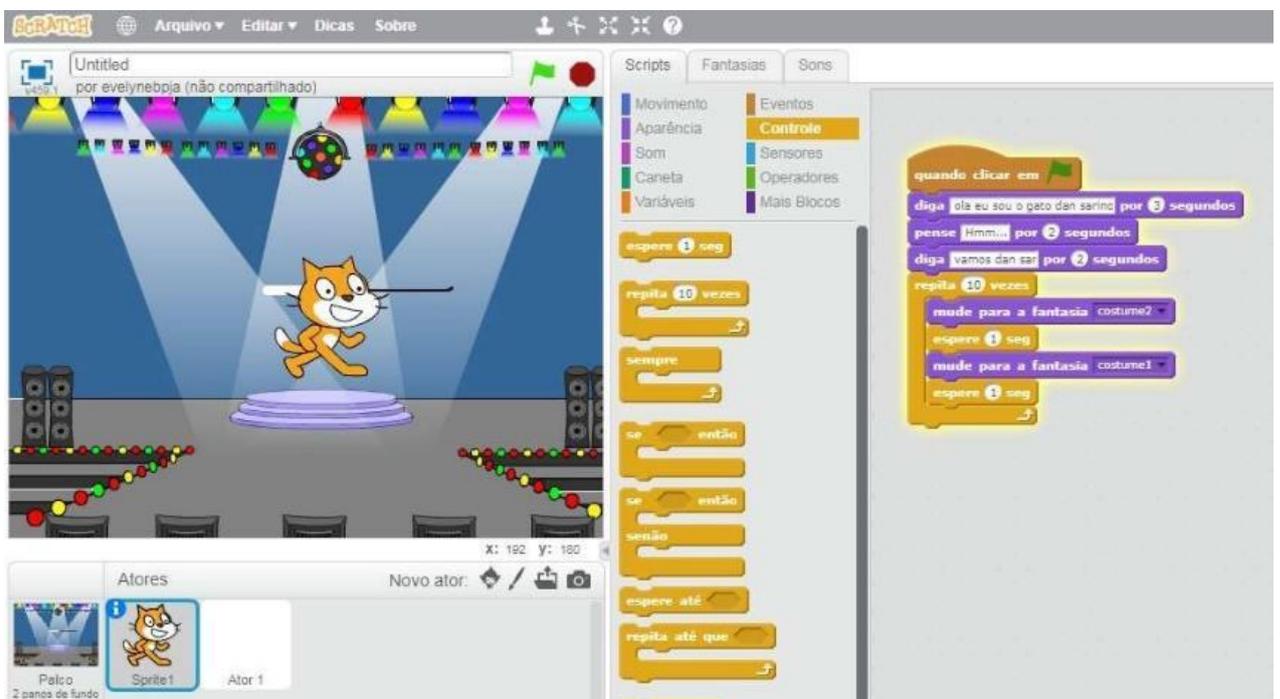
b) Plugadas: Scratch – no blog

Aqui, o objetivo foi propor às crianças atividades de utilização da plataforma Scratch para criação de algoritmos para solucionarem problemas. Foram três missões, sendo que novos elementos e blocos de programação foram incluídos em cada uma delas. As crianças trabalharam em duplas e receberam um roteiro que as guiou, com explicações sobre o ambiente de programação do Scratch, a descrição dos objetivos das missões e explicações sobre a utilização de seus blocos de comandos. A primeira missão foi “Olha o morcego passando”. Nela, o objetivo foi criar uma programação para deslocar um morcego pela tela, fazê-lo bater as asas e falar a mensagem solicitada no roteiro. A segunda foi “Pega-pega com o gato”. Desta vez, o objetivo foi animar um gato, fazê-lo se mover pela tela e, quando clicado, mostrar uma mensagem. Na terceira missão, “Labirinto

ninja”, o objetivo foi levar o ninja, por meio das setas do teclado, até o portal, desviando de obstáculos que encontrasse pelo caminho; ao chegar no portal, devia aparecer a mensagem “você venceu”.

As três missões possibilitaram que elas conhecessem o ambiente de programação do Scratch. Depois de encerradas as atividades das missões, as crianças tiveram tempo livre para criarem seus próprios projetos. As figuras de 10 a 13 mostram algumas dessas produções livres e evidenciam como aplicaram em novos projetos pessoais aquilo que aprenderam na plataforma.

Figura 10: Animação O gato dançarino



Fonte: Couto (2018).

Figura 11: Animação O Gato Andante



Fonte: Couto (2018)

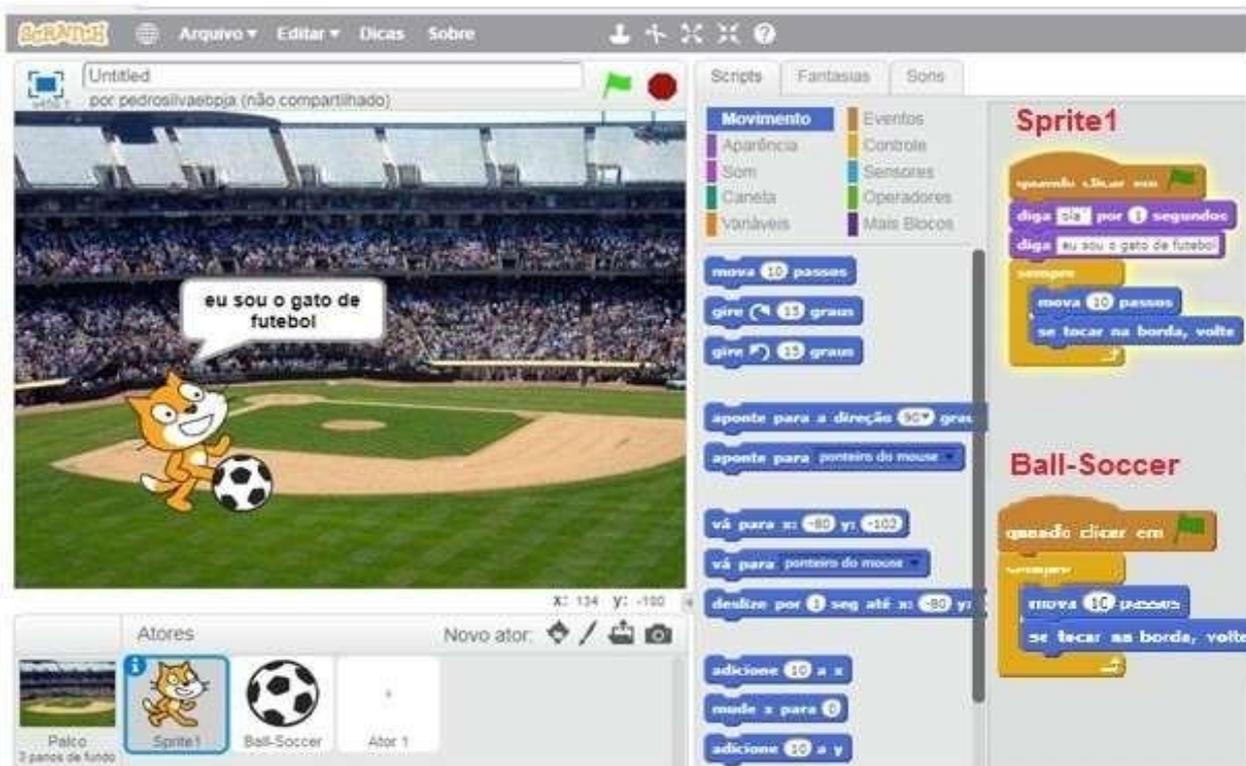
Figura 12: Animação Lá vem o fantasma



Fonte: Couto (2018)

Além disso, mostram a utilização de novos recursos, como aconteceu com o “Gato do futebol”, em que as crianças inseriram dois personagens em cena e utilizaram um código mais complexo para promover a interação.

Figura 13: Animação O Gato do Futebol



Fonte: Couto (2018)

Após análise dos códigos e das ações das crianças no seu processo de criação de algoritmos, a pesquisadora percebeu indícios do desenvolvimento do pensamento computacional na construção das animações: coleta, análise de dados, representações de dados, decomposições de problemas, abstração, criação de algoritmos; automação e simulação também foram percebidas, uma vez que os estudantes planejaram suas animações, criaram e testaram comandos. Além disso, concluiu que as crianças conseguiram agrupar os códigos e perceber a sua funcionalidade, com presença de etapas do ciclo DERDN no processo de desenvolvimento do código, execução da simulação, reflexão dos resultados obtidos, depuração do processo e melhoramento do código iniciando um novo ciclo. Com relação à interação com o computador, percebeu que elas demonstraram facilidade para transpor para o meio digital aquilo que antes desenvolveram nas atividades desplugadas. Os ambientes utilizados, por serem lúdicos, despertaram a atenção das crianças, estimularam seu interesse em explorar as plataformas digitais. Com relação à matemática, as atividades possibilitaram o trabalho com conceitos

de contagem, localização, lateralidade, posição no espaço e operação entre números por meio das estruturas de repetição.

Encerradas as atividades, as crianças responderam a um questionário para registrarem suas opiniões a respeito da oficina. Como resultado, classificaram as atividades desplugadas envolvendo brincadeiras com o robô PJ como excelentes (3) ou muito boas (8); desafios do robô PJ como excelentes (7) ou muito boas (4). O roteiro Despluga Aí! e suas brincadeiras com o pirata foram classificadas pelos estudantes como excelente (5) e muito boas (6); quanto ao nível dessas atividades, os estudantes classificaram como muito fácil (6), fácil (3), difícil (1) e muito difícil (1). O curso Labirintos Clássicos executado no Code.org foi classificado pelos estudantes como muito legal (7) e legal (4). A criação de jogos e animações no Scratch foi classificada por estudantes como muito legal (5), legal (4) e normal (2). Baseada na avaliação que fez dessas respostas, a pesquisadora concluiu que o Curso Programador do Futuro foi bem avaliado pelas crianças.

Preparar e propor as atividades, ensinar, acompanhar e ouvir as crianças foi importante para a finalização do produto educacional. Assim, podemos concluir que foi relevante a aplicação para validação do produto educacional. Após a oficina, recursos didáticos foram ajustados para publicação no blog Programador do Futuro. Como as crianças usaram o blog para acesso às atividades do Code.org e do Scratch, também foi possível analisar sua interação com o ambiente para ajustar elementos de ergonomia, usabilidade e navegabilidade.

Conclusões: aquelas provisórias, aquelas construídas como reflexão e aquelas em perspectiva

Katiane atua como instrutora de informática na rede municipal de ensino de Itajaí/SC desde 2007. Sua prática profissional e seu conhecimento a respeito de situações de uso da tecnologia no ambiente escolar por parte de professores e estudantes apoiaram o desenvolvimento de seu produto educacional. Esta fala resume sua motivação para estudo e desenvolvimento realizados no mestrado profissional:

Baseada na minha experiência profissional, acredito que, em se tratando da tecnologia educacional, é preciso ir além do uso objetos de aprendizagens e jogos educativos, como por exemplo, propor atividades que envolvam a criatividade e o desenvolvimento do pensamento computacional no processo de aprendizagem dos estudantes. (Couto, 2018, p. 12).

Meu olhar de orientadora, de docente e de pesquisadora, para tais produções, despertaram minhas análises não somente sobre o trabalho desenvolvido por Katiane, mas também para as questões ligadas ao desenvolvimento do pensamento computacional das crianças e à sua formação escolar em matemática nos primeiros anos do Ensino Fundamental.

Portanto, as discussões que apresentamos neste artigo se referem aos contextos dos anos iniciais do Ensino Fundamental e da Educação Matemática – porque neles se deram as práticas aqui tratadas – mas podem ser estendidas para outros níveis de ensino onde couber o debate a respeito do desenvolvimento do pensamento computacional de estudantes. O estudo realizado nos fez concordar com professores e pesquisadores que acreditam que as relações estabelecidas na escola, tanto para ensino quanto para aprendizagem se veem modificadas na contemporaneidade em virtude dos recursos tecnológicos digitais aos quais temos acesso para uso em diferentes situações de nosso cotidiano. Essa constatação reforça o debate – que já vem de longa data – de que o professor não pode se colocar (e ser visto) como um transmissor de conhecimento e sim como um mediador do processo de construção de conhecimento vivenciado pelos estudantes.

Especificamente, o trabalho de Katiane resultou na elaboração de uma proposta de ensino de programação na perspectiva do desenvolvimento do pensamento computacional dos estudantes, na divulgação dessa proposta para compartilhá-la com professores e pesquisadores interessados na temática e na contribuição com as discussões a respeito da tecnologia educacional como tendência em Educação Matemática. Dessa forma, contribuiu no que se refere à responder às inquietações a respeito do uso da tecnologia, na contemporaneidade, nos processos de ensino e aprendizagem na escola, entendendo a necessidade de colocar o estudante em um papel ativo de sua própria aprendizagem, como produtores de conteúdos e não apenas como consumidores daquilo que recebem de forma pronta. É disso que se tratam as propostas que se dispõem a ensinar programação - ou desenvolver o pensamento computacional - a estudantes desde a Educação Básica. Como o trabalho de Katiane mostrou, isso é possível desde os primeiros anos do Ensino Fundamental, para que as crianças, além de aprenderem a lidar com códigos de programação, aprendam a criar, propor, testar e implementar estratégias que solucionem problemas.

Com relação ao ensino e à aprendizagem de conteúdos de matemática, os contextos de resolução de problemas em ambientes plugados ou desplugados, lúdicos como os utilizados na pesquisa aqui apresentada, criam cenários criativos e motivadores para contextualizar os problemas e significar suas soluções. A respeito da relação entre atividades desplugadas e plugadas, viu-se como importante a combinação de ambas, para permitir que os estudantes trabalhem na transição das soluções analógicas para sua proposição e implementação nos ambientes digitais.

É importante ressaltar que a oficina proposta por Katiane teve duração total de 13 horas/aula porque fez parte de seu processo de pesquisa. Recomendamos que o trabalho com atividades para desenvolvimento do pensamento computacional seja planejado de modo ampliado e articulado com o currículo escolar, para que possa ter seu lugar de forma mais efetiva no processo de desenvolvimento das habilidades esperadas.

As ações do nosso grupo de pesquisa, na continuidade do estudo aqui apresentado, ampliaram-se e seguiram investigando o desenvolvimento do pensamento computacional na Educação Básica e sua relação com a Educação Matemática. Seguimos nossos estudos acadêmicos e profissionais para contribuímos com a construção de conhecimentos sobre o tema.

Referências

- Allevato, N. S. G., & Onuchic, L. D. L. R. (2014). Ensino-aprendizagem-avaliação de Matemática: por que através da resolução de problemas. *Resolução de problemas: teoria e prática. Jundiaí: Paco Editorial*, (35) 35–52.
- Bell, T., Witten, I. H., & Fellows, M. (2011). Computer Science Unplugged–Ensinando Ciência da Computação sem o uso do computador. *Tradução coordenada por Luciano Porto Barreto*, 3-45. <https://classic.csunplugged.org/wp-content/uploads/2014/12/CSUnpluggedTeachers-portuguese-brazil-feb-2011.pdf>
- Couto, K. C. (2018) *O ensino de programação nos anos iniciais do ensino fundamental: do estudo do pensamento computacional à proposta de mídias educacionais*. Dissertação de mestrado, Programa de Pós-Graduação em Ensino de Ciências, Matemática e Tecnologia, Universidade do Estado de Santa Catarina. Brasil. https://sucupira.capes.gov.br/sucupira/public/consultas/coleta/trabalhoConclusao/viewTrabalhoConclusao.xhtml?popup=true&id_trabalho=6400489
- Couto, K. C. (2018a). *Blog Programador do Futuro*. <http://programadordofuturo.blogspot.com>.
- Iste, C. (2011). *Computational Thinking: leadership toolkit*. <https://c.ymcdn.com/sites/www.csteachers.org/resource/resmgr/471.11CTLeadershipToolkit-S.pdf>.

- Onuchic (1999) Ensino-aprendizagem de Matemática através da Resolução de Problemas. Em Bicudo, M. A. V. (Org) *Pesquisa em educação matemática: concepções e perspectivas* (pp. 199-218). Ed. UNESP.
- Valente, J. A. (1995). Informática na educação: confrontar ou transformar a escola. *Perspectiva*, 13(24), 41-49. <https://periodicos.ufsc.br/index.php/perspectiva/article/view/10703/10207>
- Valente, J. A. (1999). *O computador na sociedade do conhecimento*. Unicamp/NIED, 6.
- Valente, J. A. (2008). *Diferentes usos do computador na educação*. Em aberto, 12(57). <http://rbepold.inep.gov.br/index.php/emaberto/article/download/1876/1847>
- Valente, J. A. (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*, 14(3), 864-897. <https://www.redalyc.org/pdf/766/76647706006.pdf>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. <https://dl.acm.org/doi/10.1145/1118178.1118215>

Autoras

Luciane Mulazani dos Santos

Professora Associada do Departamento de Pedagogia no Centro de Ciências Humanas (FAED) da Universidade do Estado de Santa Catarina (UDESC). Docente permanente do Programa de Pós-Graduação em Educação (PPGE) da UDESC, na linha de pesquisa Educação, Comunicação e Tecnologia; docente colaboradora do Programa de Pós-Graduação em Ensino de Ciências, Matemática e Tecnologias (PPGECMT) da UDESC na área de concentração Ensino de Matemática. Licenciada em Matemática pela UFPR, mestra e doutora em Educação - linha de Educação Matemática - pelo Programa de Pós-Graduação em Educação da UFPR. Líder do Grupo de Pesquisa NEPESTEEM - Núcleo de Estudo e Pesquisa em Tecnologia Educacional e Educação Matemática. E-mail: luciane.mulazani@udesc.br.

Katiane Cugik Couto

Mestra em Ensino de Ciências, Matemática e Tecnologias pelo Programa de Pós-Graduação no Ensino de Ciências, Matemática e Tecnologias (PPGECMT) da Universidade do Estado de Santa Catarina (UDESC), licenciada em Matemática pela Fundação Universidade Regional de Blumenau (FURB). Atua como instrutora de informática na rede municipal de ensino da Prefeitura de Itajaí/SC. Autora dos blogs infojoia.blogspot.com e programadordofuturo.blogspot.com nos quais compartilha experiências e materiais pedagógicos para serem usados na sala de informática educativa. E-mail: kcugik@yahoo.com.br