

DOI: <https://doi.org/10.48075/ReBECeM.2024.v.8.n.3.30965>

**ENTRELAÇAMENTOS ENTRE A MATEMÁTICA E O PENSAMENTO
COMPUTACIONAL NA CONSTRUÇÃO DE JOGOS ELETRÔNICOS POR
MEIO DO *SCRATCH* EM ATIVIDADES REMOTAS**

**INTERLACEMENT BETWEEN MATHEMATICS AND
COMPUTATIONAL THINKING IN THE CONSTRUCTION OF ELECTRONIC
GAMES THROUGH *SCRATCH* IN REMOTE ACTIVITIES**

Mateus Dauã de Morais¹Rodrigo Dalla Vecchia²

Resumo: Este artigo tem como objetivo apresentar um recorte de uma pesquisa de caráter qualitativo que buscou compreender possíveis entrelaçamentos entre a Matemática e as ações associadas ao Pensamento Computacional. Esta problemática foi conduzida pela seguinte pergunta: como os conceitos matemáticos emergem no processo de Decomposição de problemas, quando estudantes produzem jogos eletrônicos utilizando o *software Scratch*? Os dados foram produzidos a partir de encontros que, devido às restrições para conter a pandemia do Coronavírus (SARS-CoV-2), aconteceram remotamente com estudantes dos Anos Finais do Ensino Fundamental de uma escola pública no estado do Rio Grande do Sul. Como principal resultado, observamos que o Pensamento Computacional e a Matemática atuam em movimentos complementares na condução dos problemas inerentes ao processo de construção de jogos e que a Decomposição, um dos pilares do Pensamento Computacional, não ocorre de modo direto ou linear, podendo se mostrar de diferentes modos na busca pela solução do problema.

Palavras-chave: Educação Matemática; Decomposição; Ensino Remoto.

Abstract: This article presents an excerpt of qualitative research that sought to understand possible interlacements between mathematics and the actions associated with Computational Thinking. This problem was driven by the following question: how do mathematical concepts emerge in the problem Decomposition process when students produce electronic games using *Scratch software*? The data were produced from meetings that, due to restrictions to contain the SARS-CoV-2 pandemic, took place remotely with students in the final years of elementary school at a public school in the state of Rio Grande do Sul. With the main result, we observed that Computational Thinking and mathematics act in complementary movements in the conduction of the problems inherent to the game construction process and that Decomposition, one of the pillars of Computational Thinking, does not occur directly or linearly, being able to show itself in different ways in the search for a solution to the problem.

Keywords Mathematics Education; Decomposition; Remote Teaching.

¹ Mestre em Ensino de Matemática, UFRGS. SEDUC RS, Novo Hamburgo, Rio Grande do Sul, Brasil. mateusdaua@gmail.com

² Doutor em Educação Matemática, UNESP. UFRGS, Porto Alegre, Rio Grande do Sul, Brasil. rodrigovecchia@gmail.com

1 Introdução e justificativa

O campo investigativo associado à busca por potencialidades educacionais envolvendo a construção de jogos eletrônicos por meio do *Scratch* vem se consolidando desde o lançamento da primeira versão deste *software*. Trabalhos como os de Dalla Vecchia (2012), Corrêa (2021), Oliveira (2021) e Azevedo (2022) mostram algumas das possibilidades de utilização deste recurso no âmbito da Educação Matemática.

A riqueza de associações do uso do *Scratch* com campos investigativos da Educação Matemática também se mostra de modo múltiplo. Neste sentido, é possível encontrar na literatura trabalhos que associam este ambiente de programação à Modelagem Matemática (Dalla Vecchia, 2012), à Educação Financeira (Ferreira, 2020), à Investigação Matemática (Silva, 2020), à Formação de Professores (Barros, 2020), à construção de ideias geométricas (Horbach, 2020; Oliveira, 2021; Idem, 2022), entre outros.

É notável, no trabalho de Horbach (2020), algumas implicações práticas da exploração do *Scratch* para o ensino e aprendizagem de matemática. Com o objetivo de verificar a aprendizagem significativa do conteúdo de semelhança de triângulos em estudantes do 9º ano do Ensino Fundamental, a sequência didática elaborada pelo pesquisador foi iniciada com os estudantes explorando o *Scratch* para desenharem um quadrado e encerrada com eles livres para construírem algo relacionado o assunto.

Podemos observar na Figura 1 um exemplo de um bloco de programação construído por um dos estudantes para a desenho de um quadrado, no qual conceitos envolvendo distância e ângulos foram utilizados. Já na Figura 2, é evidenciado, junto ao bloco programado por outro estudante utilizando, além de distâncias e ângulos, variáveis e comandos de repetição, o desenho elaborado com triângulos semelhantes.

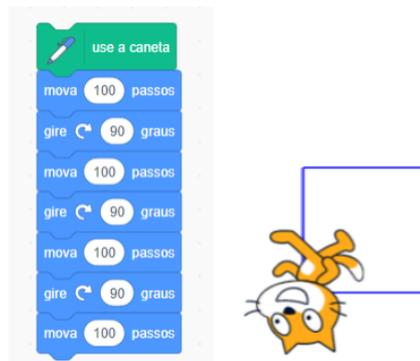


Figura 1: Bloco de programação utilizado para desenhar um quadrado no *Scratch*
Fonte: Horbach (2020).

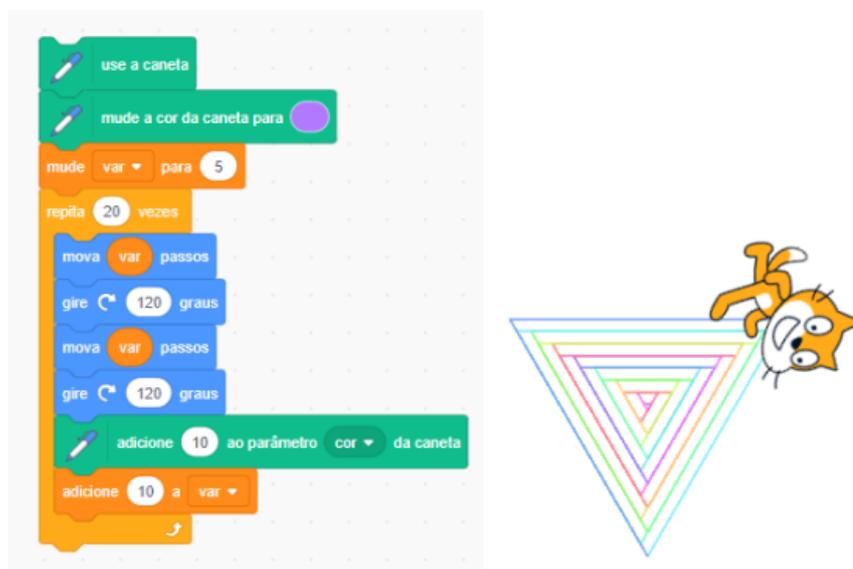


Figura 2: Bloco de programação construído para desenhar triângulos semelhantes
Fonte: Horbach (2020).

Assim como outros pesquisadores buscam relacionar o uso do *Scratch* com áreas de interesse investigativo, propomos, para este artigo, uma aproximação com o Pensamento Computacional (PC). Conforme Wing (2006), o Pensamento Computacional pode ser interpretado como um conjunto de processos mentais que envolvem a formulação e a busca de soluções de problemas por meio de conceitos fundamentais que emergem da Ciência da Computação, podendo ser interpretados por agentes humanos, máquinas ou pela combinação dos dois.

Historicamente, este conceito foi impulsionado pela pesquisadora Jeannette M. Wing que em 2006 publicou um artigo intitulado “Pensamento Computacional: um conjunto de atitudes e habilidades universalmente aplicáveis, que todos, não apenas cientistas da computação, estariam ansiosos para aprender e utilizar” (Wing, 2006, p. 33,

tradução nossa). O termo trazido pela pesquisadora é relacionado com um conjunto de habilidades que seriam fundamentais para a busca de soluções de problemas, a partir de conceitos fundamentais das Ciências da Computação, em que “[...] o poder das nossas ferramentas mentais é amplificado pelo poder das nossas ferramentas metálicas” (Wing, 2008, p. 3718, tradução nossa), como os computadores ou outras tecnologias digitais.

Algumas características do que seriam essas habilidades são apresentadas, como a reformulação de um problema difícil em problemas menores que podem ser resolvidos com mais facilidade; pensar recursivamente; utilizar-se da Abstração e Decomposição ao atacar um problema, separando-o em interesses menores; a correção de erros; planejar, aprender e agendar na presença da incerteza; entre outros (Wing, 2006).

A relação entre PC e o uso do *Scratch* já é algo que vem sendo estudado no campo da Educação Matemática. Pesquisas como as de Barcelos (2014), Vieira (2018), Barros (2020), Corrêa (2021), Ferreira (2020), Horbach (2020), Bozolan (2021) e Fernandes (2021) mostram as potencialidades deste entrelaçamento.

Apesar do campo já ser explorado, nossa inquietação parte do aprofundamento no processo determinado por Wing (2006) de Decomposição e que, de um modo geral, envolve o princípio de que para resolver um problema é preciso subdividi-lo em problemas menores.

Embora, empiricamente este processo possa parecer claro, a sua implementação em atividades é capaz de não ser, uma vez que pode ser influenciada tanto por aspectos objetivos na problemática envolvida, quanto pela subjetividade da interação e experiência dos envolvidos. Disso, surgiu nossa principal inquietação: como os conceitos matemáticos emergem no processo de Decomposição de problemas, quando estudantes produzem jogos eletrônicos utilizando o *software Scratch*?

Buscando respostas a este questionamento, convidamos alunos dos Anos Finais do Ensino Fundamental de uma escola pública no estado Rio Grande do Sul a participarem de um curso envolvendo a construção de jogos eletrônicos com o *Scratch*. O curso contou com a participação de quatro estudantes e logo nas primeiras interações observamos que a problemática associada ao Pensamento Computacional e, especificamente, ao processo de Decomposição estava sendo afetado não somente pelo próprio processo de construção, mas também pelas ideias matemáticas inerentes à

programação. Isto nos motivou a observar não somente o processo de Decomposição, mas também sua relação com a Matemática na busca por soluções.

Neste artigo, que é um recorte de uma pesquisa de caráter qualitativo que buscou compreender possíveis entrelaçamentos entre a Matemática e as ações relacionados ao Pensamento Computacional (Morais, 2022), apresentaremos alguns desdobramentos relacionados à Matemática e o processo de Decomposição.

Iniciaremos apresentando a conceituação do que é o Pensamento Computacional e quais são os pilares que sustentam as habilidades relacionadas a este conjunto de ideias. Depois, abordaremos a nossa metodologia de pesquisa, expondo o que entendemos sobre Pesquisa Qualitativa, explicando como os dados foram produzidos e como o *Scratch*, recurso tecnológico utilizado, funciona. Por fim, traremos a descrição e a análise de um recorte da pesquisa na qual buscamos compreender possíveis entrelaçamentos entre a Matemática e o Pensando Computacional, além das nossas considerações finais.

2 O Pensamento Computacional

De modo geral, o Pensamento Computacional emerge da Ciência da Computação e está associado aos processos mentais de formulação e busca de soluções de problemáticas. Esta solução pode ser interpretada por agentes humanos, máquinas ou pela combinação dos dois (Wing, 2006, 2008, 2011; NRC, 2010). O PC traz a ideia de que a identificação de um problema complexo que pode ser dividido em problemas menores aumenta a sua facilidade de resolução (Brackmann, 2017).

Cada um desses problemas pode ser analisado de forma separada, possibilitando a sua identificação com outros que já foram solucionados e focando nos detalhes que são mais importantes, em posição a informações que são irrelevantes para a solução. Além disso, um conjunto finito de passos pode ser organizado para que cada um dos subproblemas seja resolvido.

Esses processos abrangem algumas habilidades que são consideradas fundamentais para o desenvolvimento do Pensamento Computacional e que podem ser sustentadas por quatro pilares: Abstração, Decomposição, Reconhecimento de Padrões e Algoritmos, conforme a Figura 3.

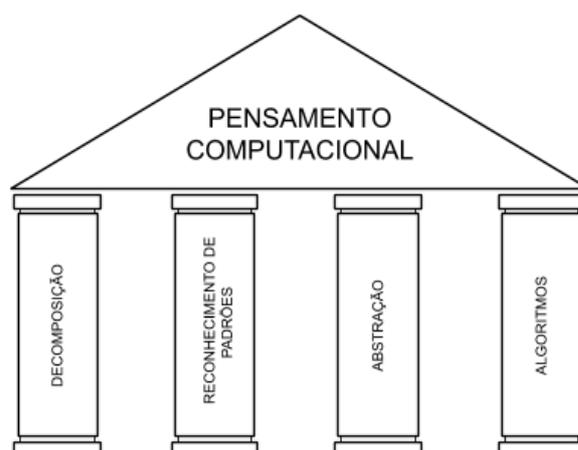


Figura 3: Os quatro pilares que sustentam o Pensamento Computacional
Fonte: Adaptada de Vicari, Moreira e Menezes (2018).

Visto por Wing (2006, 2008, 2011) como a principal habilidade do Pensamento Computacional, o pilar da Abstração pode ser definido como “[...] a ação ou o efeito de selecionar os aspectos de objetos ou de processos que devem ser considerados para satisfazer um determinado objetivo” (Brasil, 2019). É quando classificamos partes de um problema em categorias que podem ser consideradas mais ou menos importantes para a busca de sua solução. Também envolve a “[...] filtragem dos dados e sua classificação, essencialmente ignorando elementos que não são necessários para que se possa concentrar nos que são relevantes” (Brackmann, 2017, p. 38).

Uma das potencialidades da Abstração é a ideia de “[...] capturar as propriedades essenciais que são comuns em um conjunto de objetos enquanto escondemos distinções que são irrelevantes entre eles” (Wing, 2011, p. 2, tradução nossa), oportunizando-nos organizar e solucionar problemas complexos.

O segundo pilar, definido como Decomposição, apoia a habilidade de decompor um problema complexo em subproblemas que sejam mais fáceis de solucionar. De acordo com Brackmann (2017),

[...] trata-se de quebrar um problema ou sistema complexo em partes menores, que são mais manejáveis e mais fáceis de entender. As partes em menor tamanho podem, então, serem examinadas e resolvidas, ou concebidas individualmente, uma vez que são mais fáceis de trabalhar (Brackmann, 2017, p. 37).

Essa habilidade pode ser vista nas ideias construcionistas de Papert (2008). Segundo o pesquisador,

[...] é comum os estudantes falharem ao tentar resolver um problema, porque insistem em resolvê-lo por inteiro, de uma só vez; em muitos casos seria muito

mais fácil se reconhecessem que partes do problema podem ser resolvidas separadamente e depois reunidas para lidar com o todo (Papert, 2008, p. 90).

Assim, essa estratégia emerge como uma possibilidade para facilitar o processo de resolução. O problema, neste caso, pode ser decomposto em problemas menores que são resolvidos separadamente com a finalidade de encontrar a solução do principal.

Para Ribeiro, Foss e Cavalheiro (2020), definir as interfaces de entrada e saída de cada subproblema também é um fator importante e, que essa estratégia de decompor em subproblemas “[...] permite que se trabalhem cooperativamente para resolver problemas de forma organizada e eficaz” (Ribeiro; Foss; Cavalheiro, 2020, p. 25). Além disso, esta estratégia pode, de acordo com Brackmann (2017), aumentar a atenção aos detalhes do problema a ser solucionado.

Brennan e Resnick (2012) trazem a ideia de que, no *Scratch*, o desenvolvimento de um projeto não é um processo que acontece de forma linear, no qual é suficiente seguir uma sequência de passos para que ele funcione. “É um processo adaptativo, no qual o plano pode mudar em resposta à abordagem de uma solução em pequenos passos” (Brennan; Resnick, 2012, p. 7, tradução nossa). Deste modo, esta solução em pequenos passos também corrobora com a importância do pilar da Decomposição, no qual a divisão de um problema em problemas menores pode fazer com que a sua solução seja mais facilmente construída.

Outro pilar é o Reconhecimento de Padrões, que pode ser compreendido como “[...] a associação de algum objeto (ou parte dele), tangível ou conceitual, com padrões familiares que permitam identificá-lo e classificá-lo” (Brasil, 2019). É quando temos ideias de como lidar com um novo problema após identificar alguns padrões que são comuns em problemas que já conhecemos. Ribeiro, Foss e Cavalheiro (2020) destacam que é uma técnica que permite a construção de uma solução mais genérica a partir de uma preexistente, como um algoritmo, permitindo a sua utilização em outros contextos.

Por fim temos o pilar definido como Algoritmos, caracterizado pelo desenvolvimento de habilidades para a criação de “[...] uma sequência finita de etapas (passos), cada qual executável em um tempo finito, por um agente computacional, natural (humano) ou sintético (computador)” (Brasil, 2019). Assim que são construídos, os algoritmos “[...] seguirão sempre os mesmos passos pré-definidos, ou seja, podem ser repetidos quantas vezes forem necessárias, para a solução de um mesmo problema” (Vicari; Moreira; Menezes, 2018, p. 36).

Do mesmo modo que uma prova é o produto do raciocínio lógico, um algoritmo pode ser considerando o resultado do Pensamento Computacional, “[...] composto por instruções que devem ser executadas de uma forma e na ordem definida para atingir a solução desejada” (Ribeiro; Foss; Cavalheiro, 2020, p. 24). Portanto, junto ao pilar da Abstração, um processo de refinamentos sucessivos pode ser utilizado para tornar o Algoritmo mais rápido ao ignorar detalhes que não são relevantes para o objetivo.

3 Metodologia de pesquisa

Para a pesquisa realizada (Morais, 2022) assumimos um viés qualitativo, uma vez que este campo “[...] visa à compreensão interpretativa das experiências dos indivíduos dentro do contexto em que foram vivenciadas” (Goldenberg, 2004, p. 19).

Segundo Goldenberg (2004, p. 49-50), “[...] os métodos qualitativos enfatizam as particularidades de um fenômeno em termos de seu significado para o grupo pesquisa”, não defendendo um modelo único de pesquisa para todos os campos e não aceitando que os resultados se transformem em leis ou explicações gerais. Bogdan e Biklen (1994) indicam que a investigação qualitativa é descritiva, justificando assim os dados que foram produzidos a partir de registros das transcrições dos encontrados gravados, das notas de campo, das capturas de tela e de documentos que compilaram as atividades realizadas pelos estudantes.

Estes diferentes procedimentos para a produção de dados foram escolhidos pela importância da utilização de várias formas para a obtenção de dados, método denominado triangulação (Araújo; Borba, 2018; Goldenberg, 2004).

A produção de dados aconteceu junto a um projeto intitulado “Programação com *Scratch*”. Devido à pandemia do novo Coronavírus (*SARS-CoV-2*), a proposta foi ofertada de forma remota para todos os estudantes de uma escola pública da rede estadual do Rio Grande do Sul. Ao todo, 4 estudantes aceitaram o convite de participar do curso. Foram sete encontros que aconteceram semanalmente via *Google Meet* com duração de 1 hora e 30 minutos cada. Todos os participantes eram estudantes dos Anos Finais do Ensino Fundamental.

Durante a pesquisa, os cuidados éticos foram realizados, como o consentimento dos estudantes e dos seus responsáveis, registrado em termos específicos, a autorização

por escrito da direção da instituição onde foi realizada a coleta de dados e a identificação dos participantes por nomes fictícios.

O principal recurso utilizado para a produção de dados foi o *Scratch*. Entre os fatores que consideramos importantes para a escolha deste ambiente de programação, destacamos a sua interface gráfica, a programação por blocos e a possibilidade de compartilhar e acessar projetos de outros programados dentro da sua comunidade *on-line*.

Utilizamos o *Scratch 3*, versão mais recente no momento da pesquisa e disponibilizada no site na versão *off-line* ao ser instalada no computador. Na Figura 4 podemos observar a interface da linguagem de programação que faz parte do ambiente, que pode ser dividida em quatro áreas principais: a região dos códigos separados por categorias de cores diferentes (1), a parte central, onde o *script* do programa é montado com a construção de blocos de programação (2), o palco, que é a interface do programa que está sendo desenvolvido (3), e a área de seleção de atores e planos de fundo (4) que podem ser adicionados à interface. Os códigos utilizados na construção dos programas no *Scratch* estão representados, quando necessário, ao longo do texto entre colchetes. Por

exemplo, o código  é denotado como [quando bandeira verde for clicado].

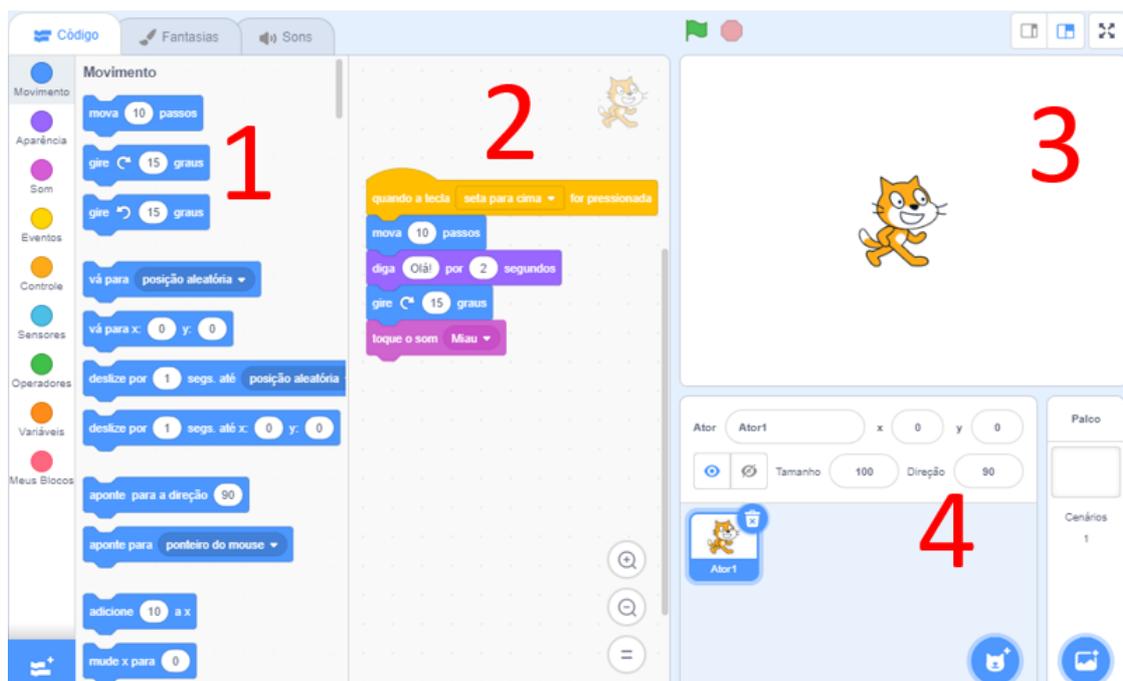


Figura 4: Interface do *Scratch* versão 3
Fonte: Morais (2022).

Além disso, por nos guiarmos em um viés construcionista, observamos o *Scratch* da mesma forma que Papert (1988, 2008) via o *LOGO*³, ou seja, o recurso fazendo parte de um micromundo, um “[...] ambiente onde falso ou verdadeiro, certo ou errado não são critérios decisivos” (Papert, 1988, p. 163), onde o estudante possa verdadeiramente usar, pensar e até brincar com a Matemática.

A partir disso, buscamos compreender as relações entre Matemática e Pensamento Computacional, norteados pelo questionamento: como os conceitos matemáticos emergem no processo de Decomposição de problemas, quando estudantes produzem jogos eletrônicos utilizando o *software Scratch*?

Na próxima seção traremos um recorte da pesquisa apresentando uma situação que permitiu encontrar indícios de respostas para esta pergunta (Morais, 2022). Esta situação envolve parte da trajetória do estudante Miguel.

Para ajudar na compreensão dos textos, consideramos, sempre que necessário, adicionar comentários às falas dos estudantes ou do professor. Esses comentários estão entre parênteses junto aos excertos. Além disso, alguns excertos possuem cortes para indicar a exclusão de algumas falas, algumas por não serem relevantes para a pesquisa e outras pela impossibilidade de entendê-las nas gravações. Esses cortes foram identificados por três pontos entre colchetes ([...]). Também, quando houver a presença de três pontos sem os colchetes (...), estamos indicando uma pausa na fala do participante.

4 Descrição e análise: a Decomposição não linear

Miguel foi um dos estudantes que, durante os sete encontros, desenvolveu três programas que envolveram a busca por soluções de diversos problemas que incluíram aspectos matemáticos e de programação. Para apresentar estes aspectos, trazemos o caso no qual o problema principal de Miguel era a programação de um algoritmo que fizesse com que um projétil partisse de um dos atores (circulado em vermelho na Figura 5) e fosse em direção à área na qual o ator principal se movimentava (circulado em verde na Figura 5).

³ Linguagem de Programação no qual os comandos são dados a uma tartaruga virtual que se desloca na tela do computador. A comunicação com o computador é feita em uma linguagem que tanto o estudante iniciante quanto o computador possam entender (Papert, 1988).

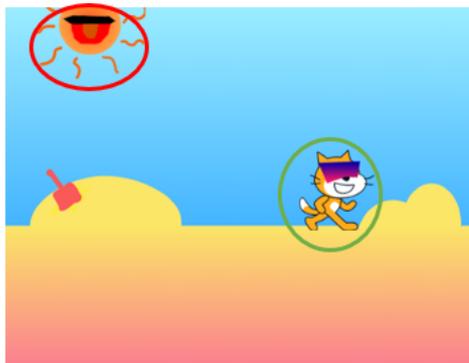


Figura 5: Interface do programa que estava sendo programado por Miguel
Fonte: Morais (2022).

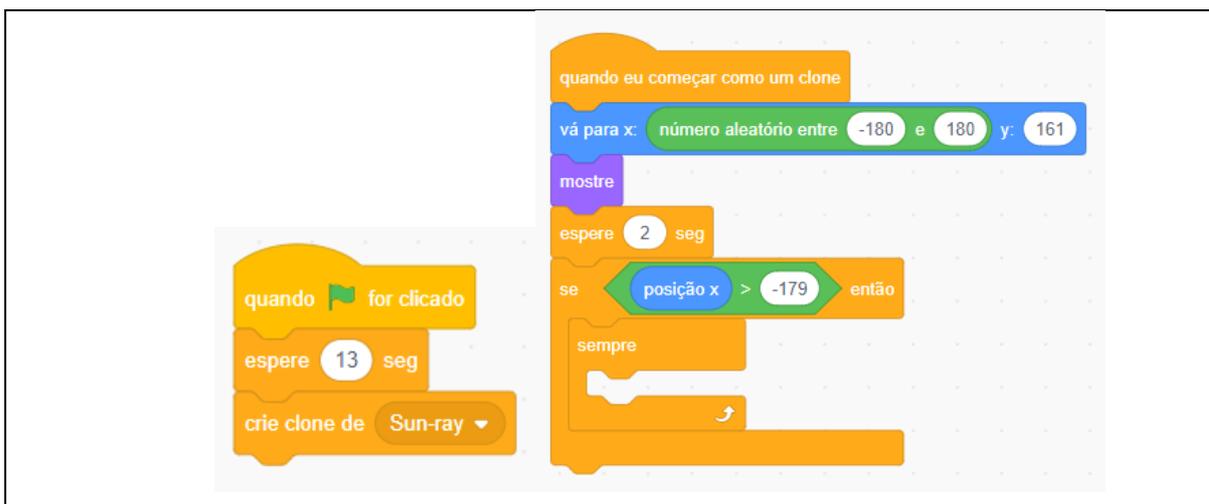
O processo de construção desse programa se desmembrou em alguns procedimentos. Para facilitar a explanação, apresentaremos a resolução do problema em três momentos, mostrando como foi a evolução do processo até Miguel encontrar a solução. As discussões principais iniciam no terceiro encontro, no qual Miguel já havia apresentado dois blocos de programação na busca da solução do problema do projétil, que podem ser vistos no Quadro 1. O primeiro bloco, que iniciava com [quando a bandeira verde for clicado], criava um clone de um “raio de sol” após 13 segundos junto ao ator circulado em vermelho na Figura 5. O segundo bloco era ativado após o surgimento do clone e guiava o movimento deste clone pelo cenário.

Excerto referente aos códigos no bloco abaixo

Professor: por que você colocou -180 a 180?

Miguel: porque se eu botasse de 0 até 180 ele ia aparecer só daqui até aqui... então eu tenho que botar de -180 até 180 positivo que é para os dois lados. (Miguel fala isso enquanto mostra, com o mouse, as posições que está falando na interface do programa criado no *Scratch*).

Blocos de programação apresentados por Miguel



Quadro 1: Entendendo a problemática de Miguel
Fonte: Morais (2022).

Temos, nos dados apresentados no Quadro 1, a ideia de Miguel de utilizar um número aleatório dentro de um certo intervalo para a posição do clone, que podemos notar no bloco [vá para x: número aleatório entre -180 e 180 y: 161] e a utilização da relação maior/menor no bloco [se posição x > -179 então], além de reflexões sobre pontos no plano cartesiano no excerto: "porque se eu botasse de 0 até 180 ele ia aparecer só daqui até aqui... então eu tenho que botar de -180 até 180 positivo que é para os dois lados".

Observamos que já houve a ação de dividir o problema inicial (construção de um algoritmo para ação do clone) em subproblemas, pois o estudante iniciou com dois blocos de programação de forma separada; um para a criação do clone e outro para a movimentação deste clone (o da esquerda e o da direita, respectivamente, que podem ser visualizados no Quadro 1). Portanto, a estratégia de Decomposição, no qual um problema pode ser dividido em problemas menores de mais fácil resolução, evidenciada por Wing (2006), já começa a ser desenvolvida. Porém, na especificidade da situação, o modo como o problema foi dividido não permitiu que o estudante encontrasse uma solução, já que o algoritmo construído com os dois blocos de programação não foi suficiente para solucionar o problema. Isso precisa fazer sentido para a problemática e tanto a Matemática quanto a própria situação podem ajudar a dar esse sentido, direcionando o estudante para solução do problema.

No quarto encontro, o estudante voltou ao assunto sobre a busca da solução do problema anterior e explicou que havia melhorado o seu programa. Compartilhando a tela do seu computador pelo *Google Meet*, Miguel novamente apresentou dois blocos de programação, mas com algumas modificações em relação aos anteriores. Os dois

iniciavam com o mesmo código [quando eu começar como um clone], mas agora o bloco à esquerda trazia o código [mude calor para 10], ou seja, Miguel utilizou uma variável denominada “calor” para controlar quando o seu clone aparecia e desaparecia. Já o bloco à direita estava com algumas alterações após o código [mostre], quando comparado ao anterior.

No Quadro 2 , podemos ver, além dos blocos de programação, um excerto da conversa do professor com o estudante depois que foi pedido para que ele explicasse como conseguiu encontrar a solução do problema que ficou em aberto.

Blocos de programação referentes à programação do projétil	
Excerto referente aos blocos acima	
<p>Miguel: eu fiquei tentando um monte de coisa aleatória, eu não sabia o que tinha dado errado, o porquê de não estar funcionando. Eu só fui tentando juntar, misturar... e eu juntei o... lembra que eu não estava conseguindo fazer ele, quando tocasse na areia, sumir?</p> <p>Professor: Sim, o projétil, isso?</p> <p>Miguel: Sim, eu criei uma variável. Eu fiz uma gambiarra. Eu criei uma variável e chamei ela de calor. Eu dei um valor de 10 para ela. Eu botei pra quando começar como um clone, que é para todos os clones que criar, ele começar com calor 10 (bloco à esquerda neste quadro).</p> <p>Professor: ok.</p>	

Miguel: eu tive que, pra fazer ele sumir, eu tive que juntar o *script* de colisão, que é para quando ele tocar na areia, com o *script* dele cair [...].

Professor: e tu entendeu o porquê disso? Que as vezes ele vai muito rápido e as vezes ele vai muito devagar?

[...]

Miguel: ah, eu acho que já sei. Porque como vai para posições aleatórias, e eu botei um certo tempo para ele percorrer essa distância.

Professor: perfeito. Se for uma posição mais longe, ele vai mais rápido ou mais devagar?

Miguel: mais rápido.

Professor: isso, porque é o mesmo tempo. Se for uma posição mais pertinho?

Miguel: ele vai mais devagar.

Quadro 2: Blocos de programação e excerto sobre a solução da problemática de Miguel

Fonte: Morais (2022).

Quando Miguel diz “eu fiquei tentando um monte de coisa aleatória, eu não sabia o que tinha dado errado, o porquê de não estar funcionando. Eu só fui tentando juntar, misturar... e eu juntei o... [...]”, observamos que o estudante notou que o *bug* que estava tentando corrigir relacionava-se a um dos clones que desaparecia da tela de interface do programa. Mesmo com ele tendo dividido o problema em dois problemas menores, ao separá-los em dois blocos, a solução ainda não havia sido encontrada.

O problema foi resolvido quando o estudante resolveu reunir os blocos. Este fato pode ser observado na seguinte fala de Miguel: “eu tive que, pra fazer ele sumir, eu tive que juntar o *script* de colisão, que é para quando ele tocar na areia, com o *script* dele cair”.

Assim, temos evidências de que o processo de reformulação de problemas em problemas menores, associado ao pilar da Decomposição do PC, não é um movimento que necessita acontecer de forma linear em um só sentido. O estudante construiu seus algoritmos com blocos de programação separados e, para que eles funcionassem e fizessem sentido para a problemática, precisou uni-los em um mesmo bloco.

Além disso, observa-se que este uso da estratégia de Decomposição em um fluxo contrário aconteceu junto ao uso da Matemática na criação da variável “calor” pois, como pode ser visto no Quadro 2, o estudante explica o seguinte:

Miguel: [...] eu criei uma variável. Eu fiz uma gambiarra. Eu criei uma variável e chamei ela de calor. Eu dei um valor de 10 para ela. Eu botei pra quando começar como um clone, que é para todos os clones que criar, ele começar com calor 10.

Nos dados do Quadro 2, também temos a ideia de grandezas proporcionais emergindo nas falas do estudante ao discutir, junto ao professor, a velocidade do projétil:

Professor: e tu entendeu o porquê disso? [...]

Miguel: ah, eu acho que já sei. Porque como vai para posições aleatórias, e eu botei um certo tempo para ele percorrer essa distância [...]

Professor: perfeito. Se for uma posição mais longe, ele vai mais rápido ou mais devagar?

Miguel: mais rápido,

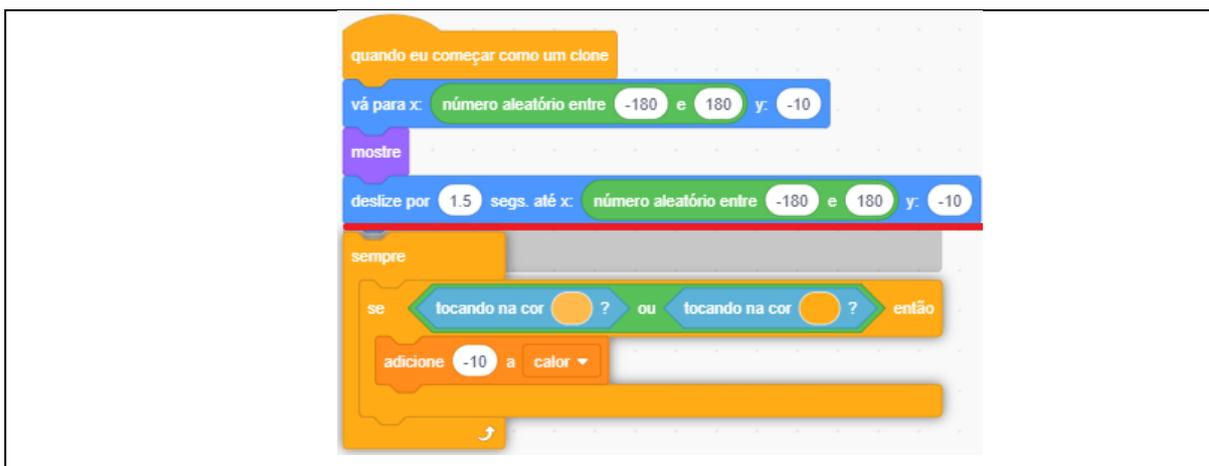
Professor: isso, porque é o mesmo tempo. Se for uma posição mais pertinho?

Miguel: ele vai mais devagar.

Desta maneira, apontando uma mobilização ou até mesmo uma possível construção do conhecimento relacionado a grandezas proporcionais a partir da experimentação realizada com o programa ao longo da decomposição, junto a reflexão orientada pelo professor.

No terceiro momento destacado nesta análise, observamos, no Quadro 3, a explicação de Miguel referente ao que programou.

Excerto
<p>Professor: e esse é o <i>script</i> do projétil? (bloco apresentado neste quadro).</p> <p>Miguel: esse é o <i>script</i> que faz ele se mover (nesse momento, Miguel divide o script em dois, para explicar o que cada parte faz, conforme pode ser visto na imagem abaixo. Está se referindo primeiro ao código acima da linha vermelha) e esse aqui era o <i>script</i> que tava separado dele que fazia que, quando ele tocasse aqui, ele sumisse (código abaixo da linha vermelha na imagem abaixo). Eu juntei com esse aqui e criei uma variável chamada calor que só se aplica ao ator do projétil. Quando o calor for 0, então o projétil se some. Seria a mesma coisa que programa colisão.</p>
<p>Bloco de códigos que Miguel se refere no excerto acima</p>



Quadro 3: Reflexões de Miguel
Fonte: Morais (2022).

Além dos aspectos matemáticos envolvendo o plano cartesiano, os números inteiros, a utilização de números aleatórios dentro do intervalo $[-180, 180]$ e o operador [ou], os dados do Quadro 3 nos mostram a estratégia que o estudante utilizou para a construção do seu programa e como isso fez com que estes aspectos matemáticos emergissem.

Retomando todos os aspectos apresentados, consideramos haver evidências de que Miguel, tanto para a construção do seu programa quanto nas ações para a busca da solução do problema apresentado, utilizou-se de diversos aspectos que estão relacionados à Matemática. Para mais, o estudante se envolveu com pelo menos dois dos pilares do Pensamento Computacional: Decomposição e Algoritmos.

Como um meio para sintetizar os aspectos que se relacionaram com o problema principal, trazemos, na Figura 6, um fluxograma que mostra como foi a trajetória do estudante para encontrar a solução dos problemas e os aspectos que emergiram durante este percurso.

Temos, no primeiro momento, a apresentação de um algoritmo envolvendo dois blocos de programação no qual já conseguimos observar a relação do estudante com pontos do plano cartesiano, números aleatórios em um certo intervalo e o uso da relação maior/menor na construção do seu algoritmo de forma fragmentada, aspecto relacionado ao PC.

No segundo momento, temos a apresentação de outro algoritmo também decomposto em dois blocos que foi construído a partir do processo de depuração. Diferente do anterior, neste observamos uma variável criada por Miguel em uma estrutura

de decisão sempre-se-então. Ademais, reflexões sobre grandezas proporcionais foram feitas, como quando Miguel responde ao professor no Quadro 2 que, se o projétil tiver que ir mais longe, ele irá mais rápido, e, se for mais perto, irá mais devagar, já que o tempo para percorrer essas distâncias será o mesmo. É neste segundo momento que notamos a não linearidade do processo de Decomposição relacionado ao PC, pois, apesar do estudante ter iniciado a construção do algoritmo dividindo o código em alguns blocos, o problema só é solucionado ao uni-los. Enfim, evidenciamos o surgimento de outros aspectos matemáticos e relacionados ao PC nas reflexões que aconteceram no terceiro momento destacado nesta análise.

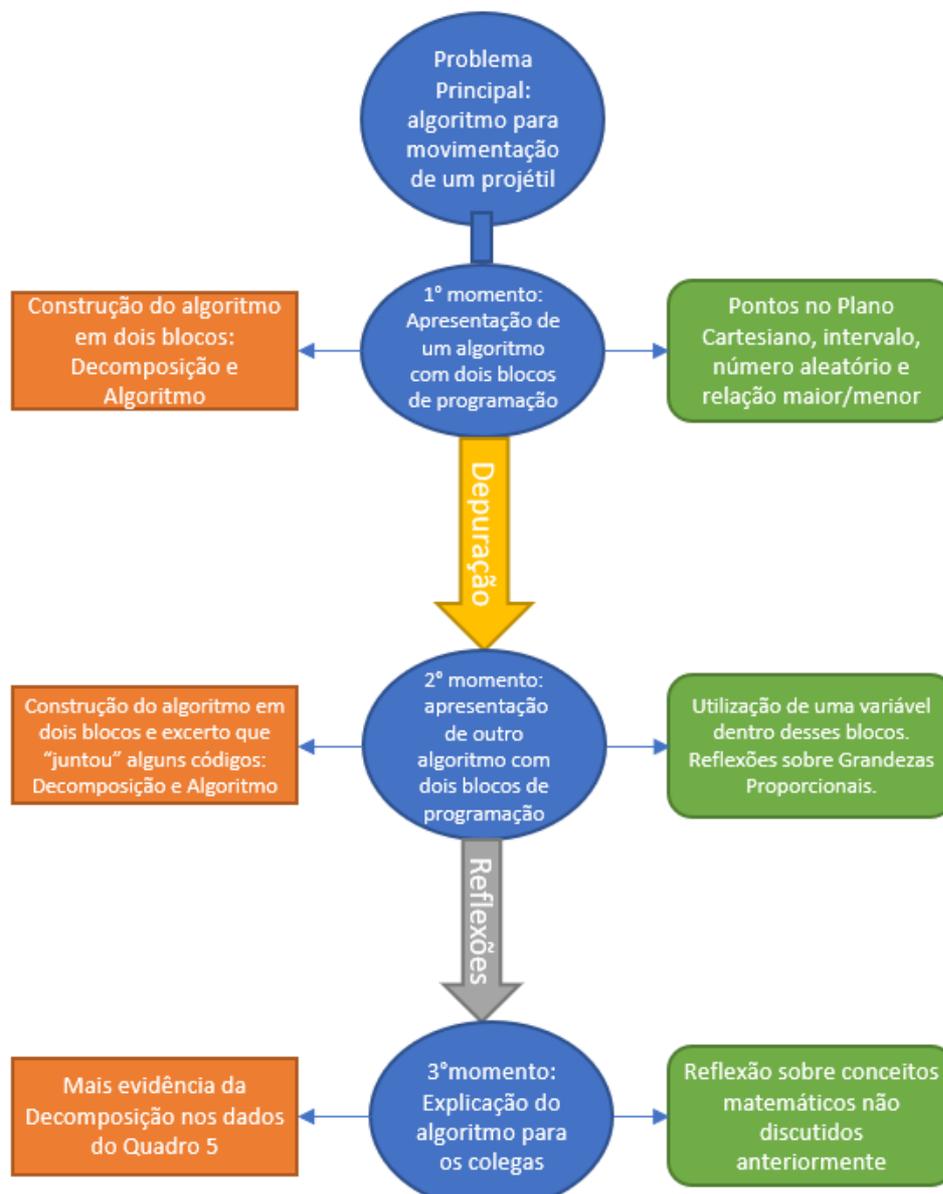


Figura 6: Fluxograma com os aspectos matemáticos e do PC que emergiram
Fonte: Morais (2022).

A construção de um algoritmo que representa a solução de um problema que pode ter sido desenvolvido junto à Decomposição é uma das competências fundamentais do Pensamento Computacional (CSTA, 2011), além de ser considerada o produto dele (Ribeiro; Foss; Cavalheiro, 2020). A articulação de símbolos e códigos, o qual o estudante possa traduzir uma certa situação de uma linguagem para outra, é, de acordo com Barcelos e Silveira (2012), uma das competências esperadas no ensino de Matemática, na qual o aluno possa “expressar suas respostas e sintetizar conclusões, utilizando diferentes registros e linguagens” (Brasil, 2018, p. 267).

Estas ações apresentadas ao longo dessa análise permitem evidenciar que a linguagem visual do estudante foi traduzida para um algoritmo na linguagem do *Scratch* que funcionou por meio da utilização da Matemática.

Mesmo com tudo o que foi analisado, observamos que não podemos apenas pensar em uma relação de causa e consequência. Não é apenas o PC que gerou a necessidade do uso da Matemática. A utilização e a manipulação dos conceitos matemáticos também geraram *insights* para que o processo de Decomposição do problema evoluísse continuamente na construção de um algoritmo que fosse eficiente para o que se desejava.

Como quando observamos os blocos de programação do Quadro 1 e notamos que a Decomposição do algoritmo na busca da solução do problema já havia acontecido, porém ele ainda não havia sido solucionado. Já no Quadro 2, vemos que existem poucas diferenças entre esses blocos e os apresentados anteriormente, entretanto, o problema foi solucionado quando o estudante criou e adicionou ao seu algoritmo a variável [calor] e programou, de forma fragmentada, outro bloco que foi posteriormente ligado ao bloco anterior, conforme visto no Quadro 3.

Portanto, a Matemática foi utilizada para fazer com que o Algoritmo, construído a partir da Decomposição, fosse a solução do problema.

5 Considerações finais

Tivemos como objetivo neste artigo responder como os conceitos matemáticos emergem no processo de Decomposição de problemas, quando estudantes produzem jogos eletrônicos utilizando o *software Scratch*. A partir da produção de dados que foi

realizada em sete encontros, conseguimos observar este processo de Decomposição e a presença de outros aspectos do PC e da Matemática nos programas construídos pelos estudantes e nas reflexões que foram realizadas nos momentos de conversa.

É importante observar que, no caso das ações de Miguel, a Matemática e o Pensamento Computacional parecem atuar como movimentos que se complementam na condução e determinação da solução. Enquanto o PC se mostrou como uma possibilidade para resolver o problema, a Matemática, em alguns casos, aparece como um problema a ser resolvido, fosse para traduzir alguma ação desejada entre diferentes linguagens ou para programar algum aspecto relacionado com o algoritmo que estava sendo elaborado.

Também observamos que a Decomposição de problemas não é algo direto, fixo ou linear, pois, apesar de Miguel ter programado blocos fragmentados e ter reunido eles para a solução, construindo assim um algoritmo, não conseguimos afirmar que essa era a intenção dele desde o começo.

Logo, notamos que os aspectos relacionados ao PC são processos que se dão ao longo de muitas mudanças e que a Matemática, neste caso, pode contribuir para auxiliar no processo de determinação do encaminhamento da solução do problema.

O Pensamento Computacional e os conceitos matemáticos emergiram por meio de projetos de caráter aberto, ou seja, a partir do desenvolvimento de artefatos ou objetos, neste caso os programas no *Scratch*, que partiram das ideias dos estudantes e que foram frutos do ambiente construcionista que foi criado para eles as materializar.

Ao observarmos os aspectos matemáticos, evidenciamos como eles emergem neste recorte: o plano cartesiano, utilizado diversas vezes na construção dos blocos de programação e ponderado pelo estudante no Quadro 1 quando foi questionado pelo professor o motivo de ter utilizado -180 e 180, respondendo o seguinte: “porque se eu botasse de 0 até 180 ele ia aparecer só daqui até aqui... então eu tenho que botar de -180 até 180 positivo que é para os dois lados”. Intervalos, números aleatórios e relação maior/menor para definir a posição na qual o “raio de sol” apareceria, evidenciada nos códigos [vá para x: número aleatório entre -180 e 180 y: 161] e [se posição $x > -179$ então]. Variável, denominada “calor”, junto a uma estrutura de decisão sempre-se-então, utilizada após um processo de depuração para corrigir um dos algoritmos relacionado ao mesmo “raio de sol” e grandezas proporcionais, realçado na fala de Miguel, no Quadro 2, sobre a velocidade do projétil dependendo da distância, já que o tempo era fixo.

Ao refletirmos sobre as limitações que tivemos durante a realização da pesquisa, destacamos que não tivemos como afirmar que todos os conceitos observados foram realmente formalizados pelos estudantes, pois, devido às atividades remotas, os estudantes construíram grande parte de seus projetos fora dos encontros. Mesmo assim, consideramos que isso não deve ser menosprezado, pois eles tiveram a oportunidade de lidar com uma Matemática que potencialmente pode ter sentido para eles, uma vez que parte de temas do interesse de cada um dos participantes, características importantes para a construção dos conhecimentos (Papert, 1986).

A partir disso, consideramos que a utilização do *Scratch*, guiadas pelas ideias construcionistas de Papert (1986, 1988, 2008), podem, mesmo por meio de atividades remotas, fazer com que conceitos matemáticos emergjam e sejam discutidos pelos estudantes e professores a partir de projetos que partam de suas próprias ideias.

Além disso, o desenvolvimento de habilidades relacionados ao Pensamento Computacional destacadas pela Base Nacional Comum Curricular (BNCC) (Brasil, 2018), como a Abstração, a Decomposição, o Reconhecimento de Padrões e a utilização e construção de Algoritmos, por estarem diretamente ligadas à busca de soluções de problemas, também podem estar presentes nesse processo.

Refletimos também que diversos aspectos que foram apresentados neste recorte podem aparecer como indicações para pesquisas futuras. Queremos buscar conceitos teóricos-filosóficos que possam nos ajudar a embasar e compreender, de forma mais profunda, as relações entre o Pensamento Computacional e a Matemática para que consigamos observar diferentes formas que aspectos relacionados a estas duas áreas podem se complementar e emergir na busca por soluções de problemas.

Por fim, espera-se que outros profissionais utilizem as ideias apresentadas neste artigo, visando a uma Educação Matemática na qual as tecnologias digitais não sejam vistas apenas como facilitadoras dos processos de ensino e de aprendizagem, mas sim como um meio que possa potencializar a construção de saberes que sejam relevantes para os estudantes e os ajudem na busca por soluções de problemas que aparecerão dentro ou fora da escola.

Referências

- ARAÚJO, J. L.; BORBA, M. C. Construindo pesquisas coletivamente em Educação Matemática. In: BORBA, M.C; ARAUJO, J. L. (orgs.). **Pesquisa qualitativa em educação matemática**. 5ª. ed. Belo Horizonte: Autêntica, 2018. p. 31-51.
- AZEVEDO, G. T. **Processo Formativo em Matemática: invenções robóticas para o Parkinson**. 213 f. Tese (Doutorado em Educação Matemática), Universidade Estadual Paulista, Rio Claro, 2022.
- BARCELOS, T. S. **Relações entre o Pensamento Computacional e a Matemática em atividades didáticas de construção de jogos digitais**. 276 f. Tese (Doutorado em Ensino de Ciências e Matemática), Universidade Cruzeiro do Sul, São Paulo, 2014.
- BARCELOS, T. S.; SILVEIRA, I. F. **Pensamento Computacional e Educação Matemática: relações para o Ensino de Computação na Educação Básica**. XX Workshop sobre Educação em Computação. Curitiba, 2012. p. 141-150.
- BARROS, T. T. T. **Formação em Pensamento Computacional utilizando Scratch para Professores de Matemática e Informática da Educação Fundamental**. 174 f. Tese (Doutorado em Informática da Educação), Universidade Federal do Rio Grande do Sul, Porto Alegre, 2020.
- BOGDAN, R.; BIKLEN, S. K. **Investigação qualitativa em educação: uma introdução à teoria e aos métodos**. 1ª. ed. Portugal: Porto, 1994.
- BOZOLAN, S. M. **Pensamento Computacional, Educação Maker e Cultura digital, aplicados aos alunos do Ensino Fundamental I**. 127 f. Tese (Doutorado em Tecnologia da Inteligência e Design Digital), Pontifícia Universidade Católica de São Paulo, São Paulo, 2021.
- BRACKMANN, C. P. **Desenvolvimento do Pensamento Computacional Através de Atividades Desplugadas na Educação Básica**. 226 f. Tese (Doutorado), Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017.
- BRASIL. Ministério da Educação. **Base Nacional Comum Curricular: Educação é a Base**. 2018. Disponível em: <http://basenacionalcomum.mec.gov.br/a-base>. Acesso em: 10 nov. 2019.
- BRASIL. Ministério da Educação. *Introdução ao Pensamento Computacional*. 2019. Disponível em: <https://avamec.mec.gov.br/#/instituicao/seb/curso/3801/informacoes>. Acesso em: 03 nov. 2024.
- BRENNAN, K.; RESNICK, M. **New frameworks for studying and assessing the development of computational thinking**. AERA 2012, Vancouver, Canadá, 2012. Disponível em: https://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf. Acesso em: 10 out. 2021.
- CORRÊA, B. S. **Programando com Scratch no Ensino Fundamental: uma possibilidade para a construção de conceitos matemáticos**. 171 f. Dissertação (Mestrado em Ensino de Matemática) – Instituto de Matemática e Estatística, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2021.

CSTA. *K-12 Computer Science Standards: Revised 2011*. New York: Association for Computing Machinery, 2011. Disponível em: <https://dl.acm.org/doi/pdf/10.1145/2593249>. Acesso em: 03 nov. 2024

DALLA VECCHIA, R. **A Modelagem Matemática e a Realidade do Mundo Cibernético**. 275 f. Tese (Doutorado em Educação Matemática), Instituto de Geociências e Ciências Exatas, Universidade Estadual Paulista, Rio Claro, 2012.

GOLDENBERG, M. **A arte de pesquisar: como fazer pesquisa qualitativa em Ciências Sociais**. 8 ed. Rio de Janeiro: Record, 2004.

FERNANDES, K. T. **Game Criativo: Desenvolvendo Habilidades de Pensamento Computacional, Leitura e Escrita através da Criação de Jogos**. 325 f. Tese (Doutorado em Ciência da Computação), Universidade Federal do Rio Grande do Norte, Natal, 2021.

FERREIRA, F. P. D. S. C. **Educação Financeira com o Scratch: Potencialidades e Dificuldades**. 115 f. Dissertação (Mestrado em Ensino de Ciências e Matemática), Universidade do Estado de Mato Grosso, Barra do Bugres, 2020.

HORBACH, I. C; **Semelhança de Triângulos: um estudo propositivo através do Scratch**. 70 f. Dissertação (Mestrado Profissional em Matemática), Universidade Federal da Fronteira Sul, Chapecó, 2020.

IDEM, R. C. **Compreensões sobre a resolução de problemas com tecnologias digitais na construção de padrões dinâmicos no Scratch por estudantes do Ensino Fundamental**. 229 f. Tese (Doutorado em Educação Matemática), Universidade Estadual Paulista, Rio Claro, 2022.

MORAIS, M. D. **Pensamento Computacional e matemática: como emergem em projetos com o Scratch no Ensino Remoto? 137 f. Dissertação (Mestrado em Ensino de Matemática)**. Universidade Federal do Rio Grande do Sul, Porto Alegre, 2022.

NRC. *Report of a Workshop on the Scope and Nature of Computational Thinking*. Washington: National Academies Press, 2010. Disponível em: <https://www.computacional.com.br/files/Geral/Report%20of%20a%20Workshop%20on%20The%20Scope%20and%20Nature%20of%20Computational%20Thinking.pdf>. Acesso em: 03 nov. 2024.

OLIVEIRA, C. D. S. **Scratch no Ensino e Aprendizagem de Geometria: um panorama de pesquisas brasileiras**. 166 f. Dissertação (Mestrado), Universidade Federal do Pampa, Mestrado Profissional em Matemática em Rede Nacional, Caçapava do Sul, 2021.

PAPERT, S. **Constructionism: a new opportunity for elementary science education**. Massachusetts Institute of Technology, 1986.

PAPERT, S. **Logo: computadores e educação**. 3. ed. São Paulo: Brasiliense, 1988.

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

RIBEIRO, L.; FOSS, L.; CAVALHEIRO, S. A. C. Entendendo o pensamento computacional. In: RAABE, A.; ZORZO, A. F.; BLIKSTEIN, P. **Computação na educação básica: fundamentos e experiências**. Porto Alegre: Penso, 2020.

SILVA, J. C; **Uso do Scratch para investigação matemática e os Números Mágicos de Ball.** 69 f. Dissertação (Mestrado), Universidade Federal do Tocantins, Arraias, 2020.

VICARI, R.; MOREIRA, A.; MENEZES, P. B. **Pensamento Computacional:** Revisão Bibliográfica. Monografia. Instituto de informática, UFRGS, 2018.

VIEIRA, M. F. V. **Pensamento Computacional com Enfoque Construcionista no Desenvolvimento de Diferentes Aprendizagens.** 182 f. Tese (Doutorado em Educação), Universidade do Vale do Itajaí, Itajaí, 2018.

WING, J. Computational Thinking. **Communications of the ACM**, v. 49, n. 3, p. 33-35, 2006. Disponível em: <http://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>. Acesso em: 7 out. 2018.

WING, J. Computational thinking and thinking about computing. **Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences**, v. 366, n. 1881, p. 3717-3725, 2008.

WING, J. **Computational thinking** - What and why? The Link Magazine, Spring, 2011.

Recebido em: 19 de abril de 2023

Aceito em: 15 de novembro de 2023