

EDUCAÇÃO FINANCEIRA E AS CONTRIBUIÇÕES DO PENSAMENTO COMPUTACIONAL EM UMA PROPOSTA DE ATIVIDADE VOLTADA À TOMADA DE DECISÃO

FINANCIAL EDUCATION AND THE CONTRIBUTIONS OF THE COMPUTACIONAL THINKING IN A DECISION-MAKING ACTIVITY PROPOSAL

Ana Karina Cancian Baroni¹

Eliel Constantino da Silva²

Marcus Vinícius Maltempi³

Sueli Liberatti Javaroni⁴

Resumo: Neste artigo trazemos uma atividade voltada à tomada de decisão em Educação Financeira, realizada no ambiente Scratch. Identificamos e valorizamos as contribuições do Pensamento Computacional no processo de elaboração da atividade, diante da sua relação com o raciocínio e os processos de pensamento. O objetivo é discutir como a Matemática e as discussões financeiras se fazem presentes, a partir de uma atividade que considera o Pensamento Computacional como um caminho para a tomada de decisão, de forma reflexiva e crítica. A análise da atividade, em conjunto com as contribuições teóricas apresentadas, nos mostra que esse caminho pode contribuir com os processos de pensamento, raciocínio, resolução de problemas e organização de ideias, indispensáveis para a tomada de decisões financeiras.

Palavras-chave: Scratch; Educação Matemática; Educação Básica.

Abstract: In this article we bring an activity focused on decision making in Financial Education, carried out in Scratch software. We identify and value the contributions of Computational Thinking in the process of elaboration of the activity, in relation to its relation with reasoning and thought processes. The objective is to discuss how Mathematics and financial discussions are presented, from an activity that considers computational thinking as a path to decision making, reflectively and critically. The analysis of the activity, together with the theoretical contributions presented, shows that this path can contribute to the processes of thinking, reasoning, problem solving and idea organization, indispensable for financial decision-making.

Keywords: Scratch; Mathematics Education; Basic Education.

¹ Doutoranda em Educação Matemática pela Universidade Estadual Paulista (UNESP). Docente do Instituto Federal de São Paulo (IFSP), Capivari, São Paulo, Brasil. E-mail: anak@ifsp.edu.br

² Doutorando em Educação Matemática pela Universidade Estadual Paulista (UNESP). Universidade Estadual Paulista (UNESP), Rio Claro, São Paulo, Brasil. elielconstantinosilva@gmail.com

³ Livre Docente pela Universidade Estadual Paulista (UNESP). Docente da Universidade Estadual Paulista (UNESP), Rio Claro, São Paulo, Brasil. E-mail: marcus.maltempi@unesp.br

⁴ Doutora em Educação Matemática pela Universidade Estadual Paulista (UNESP). Docente da Universidade Estadual Paulista (UNESP), Bauru, São Paulo, Brasil. E-mail: sueli.javaroni@unesp.br

1 Introdução

Neste artigo, apresentamos e discutimos uma proposta de abordagem da tomada de decisão em Educação Financeira, a partir de uma atividade no ambiente Scratch, explorando as ideias sobre o desenvolvimento do Pensamento Computacional em sala de aula. O objetivo é discutir como a Matemática e as discussões financeiras se fazem presentes, a partir de uma atividade que considera o Pensamento Computacional como um caminho para uma reflexão crítica e a tomada de decisão.

Quando nos referimos ao desenvolvimento do Pensamento Computacional no âmbito escolar, estamos apoiados em Mannila *et al.*, (2014, p. 2, tradução nossa) que afirma que o Pensamento Computacional é “um termo que abrange um conjunto de conceitos e processos de pensamento da Ciência da Computação que ajudam na formulação de problemas e suas soluções em diferentes campos”⁵.

Porém esse termo tem suas origens em Papert (1996) que discutiu o uso do computador na resolução de problemas de modo a provocar o surgimento de novas ideias à medida que conexões entre as soluções eram encontradas. Um pouco mais de uma década depois, Jannette Wing apresentou uma nova definição, afirmando que se trata de processos de pensamento envolvidos na formulação de problemas e suas soluções, para que estas sejam representadas de uma forma que possa ser realizada por um agente de processamento de informações (WING, 2011). Embora Papert fale em forjar ideias e Wing fale em processos de pensamento, ambos os autores enfatizam que o desenvolvimento do Pensamento Computacional está relacionado ao raciocínio (MANILLA *et al.*, 2014) e é nesta perspectiva que situamos este artigo.

A Sociedade Internacional para a Tecnologia na Educação (ISTE) e a *American Computer Science Teachers Association* (CSTA) consideram que o Pensamento Computacional é um processo de resolução de problemas que inclui características como: formular problemas; organizar e analisar logicamente os dados; automatizar as soluções; identificar, analisar e implementar possíveis soluções, generalizar e usar o processo de resolução criado, em outros problemas. À essas características, estão relacionadas habilidades como: confiança, persistência, tolerância, comunicação e a capacidade de lidar com problemas abertos.

⁵ O texto original diz: is a term encompassing a set of concepts and thought processes from CS that aid in formulating problems and their solutions in different elds.

Visando o desenvolvimento dessa ideia na Educação, a ISTE e a CSTA elaboraram uma definição operacional de Pensamento Computacional, com o objetivo de auxiliar os professores quanto ao seu desenvolvimento em sala de aula. Essa definição operacional aborda nove conceitos essenciais ao desenvolvimento do Pensamento Computacional:

Coleta de dados: busca por dados essenciais à resolução do problema.

Análise de dados: criação de conexões/sentidos aos dados coletados, objetivando a resolução do problema.

Representação de dados: forma de organização dos dados coletados e analisados com a finalidade de contribuir para a visualização do problema proposto e a elaboração do processo de solução do problema.

Decomposição do problema: gerenciar/ “quebrar” o problema em partes menores de maneira que, ao final, consiga compreendê-lo. Esse conceito é essencial para problemas mais complexos que requerem um cuidado a mais e que envolvem muitos processos de resolução, na tentativa de obter a solução final.

Redução à abstração: esforço complexo de definir a ideia central do problema.

Algoritmos e procedimentos: sequência ordenada de etapas, visando alcançar um fim comum. Note que na decomposição do problema, o conceito de algoritmos e procedimentos estaria presente em cada parte em que o problema for decomposto.

Automação: usar máquinas para realizar tarefas e ações repetitivas e tediosas.

Simulação: executar o modelo de resolução elaborado para o problema, em outras situações, experiências ou problemas.

Paralelização: execução simultânea de tarefas, visando um objetivo comum.

Entendemos que tais conceitos se fazem presentes no desenvolvimento do Pensamento Computacional, podendo contribuir com os processos de pensamento, raciocínio, resolução de problemas e organização de ideias.

Procuramos observar esses conceitos durante o desenvolvimento da atividade que descrevemos na sequência, a qual foi realizada no *Scratch*, um software que possui uma linguagem de programação do tipo *drag and drop* (arraste e solte) e permite ao usuário criar histórias, animações, sequências interativas, etc., através de programação por blocos.

Esse software possui uma coleção de “‘blocos de programação’ em que crianças os encaixam para criar programas. Tal como acontece com os blocos Lego, os conectores nos blocos sugerem como eles devem ser conectados” (RESNICK *et al.*, 2009, p. 63, grifo

do autor, tradução nossa)⁶, passando por programas e algoritmos, desenvolvendo “as habilidades dos estudantes para pensar logicamente e algoritmicamente, o que leva à depuração planejada de código ao invés de apenas uma abordagem de tentativa e erro” (BERRY, 2013, p. 11, tradução nossa)⁷.

Em particular, a programação apoia o “pensamento computacional”, ajudando você a aprender importantes estratégias de resolução de problemas e design (como modularização e design iterativo) que se transportam para domínios não programados. E uma vez que a programação envolve a criação de representações externas de seus processos de resolução de problemas, a programação oferece oportunidades para refletir sobre seu próprio pensamento, mesmo para pensar em si mesmo (RESNICK *et al.*, 2009, p. 62, tradução nossa)⁸

A opção pela linguagem de programação *Scratch* se deve ao fato dela possuir um *piso baixo* (fácil de começar) e um *teto alto* (oportunidades para criar projetos cada vez mais complexos ao longo do tempo), permitindo que os estudantes se envolvam com programação e Matemática através de um conhecimento mínimo como pré-requisito, possibilitando explorar conceitos e relações além do que está presente no currículo do seu nível escolar. A linguagem de programação *Scratch* possibilita, ainda, um envolvimento com pessoas de diferentes localidades, níveis escolares e com grau de conhecimento diferenciado, caracterizando a dimensão *muros largos* (GADANIDIS, 2015).

Com base nos pressupostos e considerações apresentados, mostraremos nas seções seguintes: a proposta que nos motivou a elaborar a atividade sobre tomada de decisão em Educação Financeira, mostrando uma possibilidade de tratar esse tema no *Scratch* e observando as contribuições do Pensamento Computacional no processo de elaboração; apresentamos, em detalhes, a atividade desenvolvida, uma descrição da programação realizada no *Scratch*, passo a passo e os conceitos matemáticos que podem ser explorados; uma reflexão sobre a possibilidade de aplicar essa atividade em sala de aula e sobre os conceitos da definição operacional de Pensamento Computacional presentes na atividade proposta; as considerações finais.

⁶ O texto original diz: “programming blocks” children snap together to create programs.... As with Lego bricks, connectors on the blocks suggest how they should be put together.

⁷ O texto original diz: Thinking through programs and algorithms helps develop pupils’ abilities to think logically and algorithmically, which leads to planned debugging of code rather than just a trial-and-error approach.

⁸ O texto original diz: In particular, programming supports “computational thinking,” helping you learn important problem-solving and design strategies (such as modularization and iterative design) that carry over to nonprogramming domains. And since programming involves the creation of external representations of your problem-solving processes, programming provides you with opportunities to reflect on your own thinking, even to think about thinking itself.

2 Educação financeira e Pensamento Computacional: uma aproximação possível?

Os dois primeiros autores desenvolvem pesquisa de doutorado nas áreas de Educação Financeira e Pensamento Computacional, respectivamente, em Educação Matemática. Interessados em explorar uma atividade no *Scratch* que aproximasse essas duas áreas e que pudesse servir de sugestão para o trabalho em sala de aula, elegeram uma modalidade de financiamento comum na atualidade para explorar a tomada de decisão em Educação Financeira: a compra de um automóvel. A forma de explorar o tema, auxiliando na tomada de decisão, foi através da programação no software, atentando para o processo de construção do algoritmo, dos cálculos matemáticos envolvidos e, especialmente, para a presença dos conceitos envolvidos no desenvolvimento do Pensamento Computacional, desde a pesquisa de opções disponíveis no mercado financeiro para a realização do financiamento, até a tomada de decisão.

Os demais autores são orientadores dos primeiros e também se interessaram em discutir a atividade proposta, refletindo sobre uma possibilidade de aproximação entre as duas áreas de estudo, através da atividade desenvolvida.

Depois de pesquisar diferentes opções de financiamento, surgiu a curiosidade de criar uma programação que tornasse mais eficiente a realização dos cálculos envolvidos para encontrar o valor das parcelas a serem pagas e do montante final da operação financeira. Embora nossos cálculos e análises sejam específicos, ressaltamos que a tarefa de encontrar a opção de financiamento mais vantajosa também depende de fatores de ordem pessoal, principalmente da possibilidade de pagar o valor das parcelas referentes a cada caso, cada uma com suas taxas e prazos específicos.

A criação dessa programação no ambiente *Scratch* permitiu realizar os cálculos e cruzar os dados das diferentes opções de financiamento pesquisadas, contribuindo com o processo de tomada de decisão. Diante disso, percebemos a possibilidade de explorar diferentes conceitos matemáticos, se valendo das contribuições do Pensamento Computacional como um caminho para essa exploração.

Uma reportagem⁹ sobre o rendimento dos estudantes brasileiros no *Programme for International Student Assessment* (PISA) destaca os resultados obtidos pelos estudantes brasileiros na avaliação em 2015, em que mais da metade deles demonstraram não dominar conhecimentos financeiros básicos. Em Brasil (2015) encontramos uma

⁹ BBC Brasil. Disponível em: <http://www.bbc.com/portuguese/brasil-40081292>. Acesso em: 02 jan. 2019.

indicação da necessidade de promover a Educação Financeira na Educação Básica, pois o documento sugere que o problema está na forma como a Educação Financeira é tratada nas salas de aula, em geral, sem aprofundamento ou ligação com o cotidiano dos estudantes.

Outro aspecto que tem sido alvo de preocupação na Educação Matemática é a questão do consumo excessivo e sem planejamento, levando ao endividamento da população (KISTEMANN JR, 2011). Um dos objetivos importantes para a efetivação da Educação Financeira, segundo Campos, Teixeira e Coutinho (2015, p. 3) é “saber aproveitar convenientemente as oportunidades de financiamentos disponíveis”, analisando cautelosamente todos os detalhes envolvidos nas operações financeiras e no planejamento pessoal e familiar.

Mas como promover situações que favoreçam tais reflexões dos estudantes, levando-os a um posicionamento crítico diante de decisões financeiras cotidianas? Cunha e Laudares (2017, p. 4) entendem que, para a efetivação de uma Educação Financeira “há necessidade de uma transição do ensino da Matemática Financeira, para o exercício da reflexão e crítica acerca de situações que influenciam a vida das pessoas (...)”, apontando como um passo importante a incorporação de problemas abertos às rotinas da sala de aula, problemas esses que vão além da aplicação de fórmulas e realização de cálculos. São valorizadas outras competências, como pesquisar, buscar informações reais, falar sobre os problemas, considerando diferentes pontos de vista, entre outras.

Essas ideias parecem sintonizadas com as propostas de trabalho do Pensamento Computacional. No vídeo, intitulado *Mitch Resnick: Vamos ensinar crianças a escrever códigos*, um dos idealizadores do software *Scratch* explica que o propósito do software vai muito além de desenvolver e aprender códigos, privilegiando a criatividade, a interação entre o estudante e a tecnologia digital, a interação entre pessoas engajadas em um projeto no ambiente digital e a aprendizagem dinâmica e multifária que o software tem potencial para promover. Nos chamou a atenção, em especial, a questão da aprendizagem através de um projeto em ambiente virtual, sendo esse o propósito que identificamos para esta atividade voltada à Educação Financeira.

Apesar de escolhermos uma questão sobre o financiamento de um automóvel, entendemos que as discussões geradas são pertinentes a muitos outros bens de consumo, podendo a atividade proposta ser facilmente reelaborada para qualquer bem de interesse. Além disso, muitas outras questões são anteriores à ação de comprar um bem de consumo e, nas discussões em Educação Financeira, são questões até mesmo mais importantes do

que apenas promover uma abordagem funcional, preparando as pessoas para receber informações e proceder de forma esperada, consumindo.

Silva e Powell (2013) apresentam uma definição de Educação Financeira que é capaz de abarcar as demais questões a que nos referimos:

A Educação Financeira Escolar constitui-se de um conjunto de informações através do qual os estudantes são introduzidos no universo do dinheiro e estimulados a produzir uma compreensão sobre finanças e economia, através de um processo de ensino que os torne aptos a analisar, fazer julgamentos fundamentados, tomar decisões e ter posições críticas sobre questões financeiras que envolvam sua vida pessoal, familiar e da sociedade em que vivem (SILVA; POWELL, 2013, p. 13).

A Educação Financeira é por nós compreendida nesse sentido, essencialmente crítica e emancipadora. A nossa atividade não tem o objetivo de abarcar todas essas questões, mas sim de apresentar uma proposta que pode provocar discussões importantes em sala de aula, usando uma situação cotidiana na atualidade, que é o financiamento de um automóvel. A nossa ênfase está no encontro entre a Educação Financeira e o Pensamento Computacional, através da proposição de uma atividade prática.

Conforme já dito, a atividade pode ser adaptada para a aquisição de outro bem de consumo, conforme a percepção do professor e o interesse de cada turma. Além disso, recomendamos fortemente que, a partir da atividade proposta, a sua abordagem também inclua discussões importantes no contexto da Educação Financeira. Alguns exemplos de discussões podem ser encontrados em Baroni e Maltempi (2019, p. 262), quando tratam do financiamento de um imóvel:

[...] o desenvolvimento de uma consciência coletiva a respeito do endividamento e dos fatores de ordem política e econômica envolvidos nesse cenário; o engajamento para que as taxas praticadas no mercado financeiro sejam mais justas; as relações entre consumo e meio ambiente; a problemática do consumismo e do “ter para ser”; as possibilidades de uma comunidade se organizar para criar o seu próprio fundo de crédito, sem depender de bancos para a captação de recursos financeiros, entre tantos outros exemplos e possibilidades.

A atividade que trazemos é uma pequena contribuição, em um universo muito maior de ações que são importantes para a promoção da Educação Financeira. Ela trata, em especial, da tomada de decisão por uma modalidade de financiamento ou empréstimo pessoal para adquirir um bem, exige a comparação de diferentes alternativas, além de, indiretamente, levar o estudante à reflexões anteriores como o planejamento financeiro pessoal e/ou familiar necessário para o pagamento da dívida, um desejável conhecimento de como os juros se processam, a importância de uma análise conjunta das variáveis

envolvidas na operação financeira (taxa de juros, prazo e sistema de amortização da dívida), entre outras.

Tomar uma decisão financeira requer também, no mínimo, informar-se das taxas de juros praticadas no mercado, o que ultrapassa as paredes da Matemática formal, exigindo um conhecimento inter e transdisciplinar, pois a conexão da Matemática com outras áreas “pode propiciar práticas marcadamente interdisciplinares e transdisciplinares, promovendo a produção de significados que auxiliarão na regulação dos saberes necessários e suficientes para a tomada de decisão” (KISTEMANN JR, 2011, p. 285).

A parceria com as tecnologias na abordagem da Matemática Financeira também é apontada como positiva em alguns estudos. Os pesquisadores Campos, Teixeira e Coutinho (2015) acreditam que não é possível pensar em atividades com a Matemática Financeira abrindo mão do uso de Tecnologias Digitais da Informação e Comunicação, pois “o uso dessas tecnologias tende a motivar o estudante e a potencializar a aprendizagem, na medida em que permite trabalhar com diversas simulações e com valores reais, valorizando tanto o conteúdo como o método” (CAMPOS; TEIXEIRA; COUTINHO, 2015, p. 568).

A partir dessas considerações e pressupostos visualizamos uma possibilidade de aproximação entre a Educação Financeira e o Pensamento Computacional, favorecendo uma reflexão sobre as operações financeiras e os processos envolvidos na tomada de decisão.

3 A atividade e os cálculos envolvidos

A proposta trazida na atividade envolve a tomada de decisão sobre a melhor opção para financiamento de um automóvel, a partir de duas (ou mais) opções disponíveis no mercado. Com base nesse problema, a atividade proposta é usar o ambiente *Scratch* para programar a simulação de dois (ou mais) financiamentos de um automóvel e apresentar a proposta de financiamento que terá o menor valor a ser pago em juros.

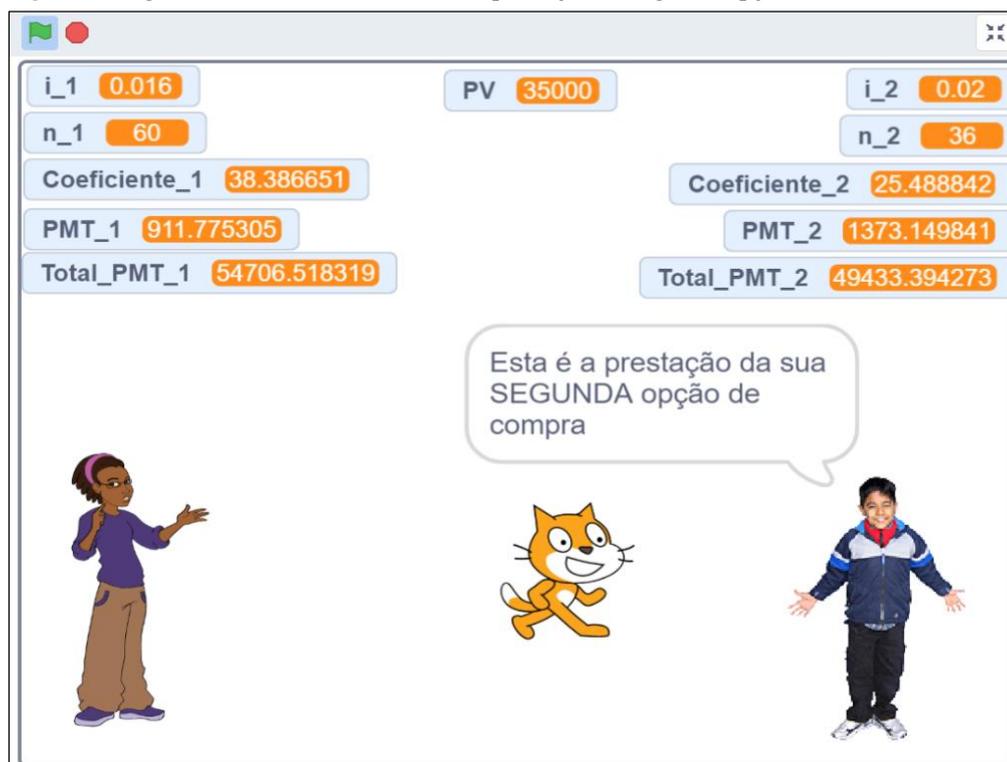
Essa atividade pode contribuir para a realização dos cálculos necessários, o cruzamento das informações referentes às diferentes opções de financiamento encontradas, colaborando na tomada de decisão. É importante que, inicialmente, os estudantes pesquisem situações reais para a realização da atividade e a tomada de decisão, a partir de investigações sobre as diferentes opções de financiamento de um automóvel

(em algumas agências concessionárias e bancárias), os valores de um automóvel de seu interesse, a expressão matemática utilizada para o cálculo de financiamento, entre outras.

Na próxima seção, apresentamos o passo a passo do desenvolvimento da atividade, disponível em: <https://scratch.mit.edu/projects/188251121/>. Objetivando dar uma ideia inicial de como procedemos, mostramos, a seguir, algumas imagens da atividade desenvolvida no software.

Observe que, na parte superior, de um lado encontram-se as informações relativas à primeira opção de financiamento, mostrando os valores da taxa de juros (i_1 , $0,016=1,6\%$ ao mês), o número de parcelas (n_1 , 60 meses) e o coeficiente de financiamento (c_1 , $38,386651$); da mesma forma, do outro lado, encontram-se as informações relativas à segunda opção (i_2 , $0,02=2\%$ ao mês; n_2 , 36 meses; c_2 , $25,488842$).

Figura 1: Segundo ator indicando o valor da prestação da segunda opção de financiamento



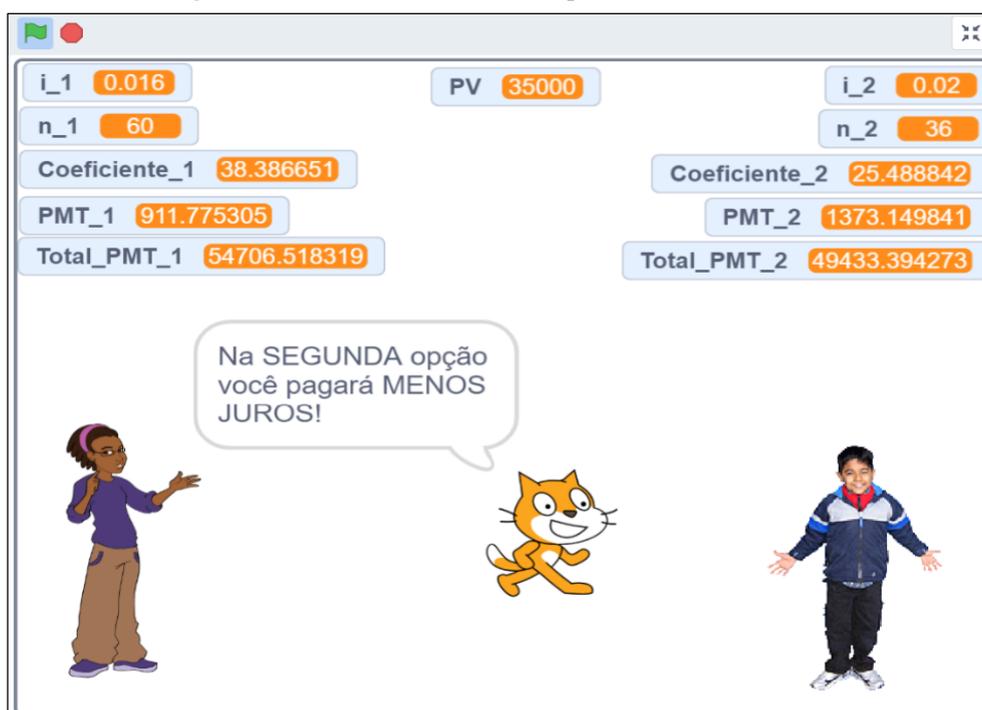
Fonte: Elaborada pelos autores, a partir do ambiente Scratch

O primeiro valor a ser informado é o valor do automóvel, PV, R\$ 35000,00. Após o cálculo da primeira prestação ser realizado pelo software, o primeiro ator dá a informação acima, mostrando o valor da variável PMT_1 (R\$ 911,775305), que se refere à primeira opção de compra.

Da mesma forma, assim que a segunda prestação é calculada, o segundo ator informa o valor da variável PMT_2 , $R\$ 1373,149841$ e, em seguida, também fornece o montante pago nas duas opções, $TOTAL_PMT_1$, no valor de $R\$ 54706,518319$ e $TOTAL_PMT_2$, $R\$ 49433,394273$. No software há a opção de arredondarmos os valores para o número inteiro mais próximo anterior ou posterior ao número, mas não há a opção de arredondarmos para duas casas decimais, tendo em vista que os valores obtidos representam um valor financeiro. Nesse sentido, optamos por permanecer com todos os algarismos pertencentes ao número, pois arredondar os números para o número inteiro mais próximo pode provocar erros nos cálculos. No entanto, consideramos importante apresentar essa discussão com os estudantes no momento de realização da atividade.

Ao final da execução do programa, o terceiro ator (gatinho, ao centro) dirige-se ao ator que representa a melhor opção de compra (definida como aquela em que será pago o menor valor em juros durante o período do financiamento) e dá o seu parecer.

Figura 2: Terceiro ator contribuindo para a tomada de decisão



Fonte: Elaborada pelos autores, a partir do ambiente Scratch

Assim, ilustramos de forma breve a ideia central da atividade. Antes de passar ao detalhamento pormenorizado da programação, apresentamos a equação matemática utilizada para o cálculo das parcelas. Como as opções de financiamento de automóvel encontradas no mercado brasileiro, na maioria dos casos, envolvem o pagamento através de uma sequência uniforme de prestações (geralmente mensais), a equação que utilizamos

é conhecida como a do valor atual de uma sequência uniforme postecipada, ou seja, uma sequência de n parcelas iguais e sem entrada. E por que “valor atual”? A razão é que conhecemos o valor à vista do automóvel, ou seja, o seu valor na data de hoje. O termo “postecipada” é usado para se referir ao fato de que o primeiro pagamento é realizado no primeiro período após a data da compra e assinatura do contrato de financiamento, ou seja, no período 1.

A equação, além do valor atual (PV), envolve as variáveis: “ i ” (taxa de juros), “ n ” (número de períodos/prestações) e “PMT” (pagamento/prestação):

$$PV = PMT \cdot \frac{(1+i)^n - 1}{(1+i)^n \cdot i}, \text{ que é equivalente a } PMT = PV \cdot \frac{(1+i)^n \cdot i}{(1+i)^n - 1}$$

Note que o termo $\frac{(1+i)^n - 1}{(1+i)^n \cdot i}$ é composto pelo valor da taxa de juros e do número de períodos/prestações e, assim, fornece o fator pelo qual o valor atual é dividido, gerando o valor das prestações. Esse termo é chamado *fator de valor atual*, conhecido também como o *coeficiente* da operação (PUCCINI, 1999). Por essa razão, também destacamos na programação esse coeficiente, pedindo que ele fosse mostrado em cima de cada ator, antes do cálculo do valor das prestações. Embora esse não seja o nosso foco nesse momento, destacamos que seria interessante discutir em sala de aula como essa expressão é gerada, através da soma dos termos de uma Progressão Geométrica.

Para o cálculo da prestação optamos pela divisão do valor de PV pelo coeficiente $\frac{(1+i)^n - 1}{(1+i)^n \cdot i}$, embora seja também possível, de forma equivalente, multiplicá-lo pelo inverso dessa razão. Na próxima seção, detalhamos como a programação foi realizada, bem como as reflexões que estiveram presentes durante a sua realização.

4 Programando

Há diversas maneiras de desenvolver a atividade no *Scratch*. O estudante poderá iniciar, por exemplo, programando o cálculo do coeficiente, mas perceberá que, para realizá-lo, precisará ter o valor a ser financiado, a taxa de juros e o número de parcelas. Outro exemplo é se o estudante começar programando a tomada de decisão. Ele perceberá que, para isso, precisará dos valores da parcela, em cada opção de financiamento e que, para obter esse valor, é necessário calcular o coeficiente, que requer que saibamos, novamente, o valor a ser financiado, a taxa de juros e o número de parcelas.

Assim, iniciamos com esses valores que precisam ser informados, como pré-requisitos para o cálculo das prestações. O valor a ser financiado (PV), a taxa de juros (i)

e o número de parcelas (n) são os valores a serem informados, de acordo com cada opção de financiamento, podendo variar. Já de início podemos explorar o conceito de variável com os estudantes, bem como as relações de dependência entre elas. É um momento propício para verificar se os estudantes compreendem tal conceito, abrindo possibilidades para a discussão.

Matematicamente, a variável é utilizada para representar a substituição de um termo desconhecido que pode assumir diferentes estados de grandeza. No *Scratch*,

[...] as variáveis são usadas para acompanhar as coisas que podem mudar enquanto um programa está sendo executado. Eles são um pouco como x ou y em álgebra, na medida em que os valores podem não ser conhecidos inicialmente. As variáveis não são usadas apenas para números. Eles também podem manter o texto, incluindo frases inteiras ('strings'), ou os valores lógicos 'true' ou 'false'. ... as variáveis são como caixas, na medida em que o computador pode usá-las para armazenar informações que podem ser alteradas pelo usuário, pelo programa ou por outra variável (BERRY, 2013, p. 11)¹⁰.

Dessa maneira, teremos que criar três variáveis: uma variável para o PV (que será o mesmo para as duas opções de financiamento), uma variável para a taxa de juros (chamamos de i_1 , para indicar a taxa de juros da primeira opção) e uma variável para o número de parcelas (chamamos de n_1 , número de parcelas da primeira opção de financiamento).

Criadas as variáveis, o próximo passo é informar os seus valores, já pesquisados junto às instituições financeiras. O programa os solicitará ao usuário e, antes de programarmos esse pedido, precisamos escolher qual ator fará essa pergunta. A escolha desse ator poderá ser realizada através das opções presentes na biblioteca de atores disponível no software, aba "ateliê", podendo também ser uma foto pessoal captada no momento da programação, uma imagem disponível no computador ou, ainda, uma imagem capturada da internet.

Definido o ator, podemos realizar a programação referente ao pedido desses valores, utilizando o bloco *pergunte e espere a resposta* para pedir ao usuário o valor do automóvel a ser financiado, a taxa de juros e o número de parcelas. Quando informado o valor do automóvel, a resposta do usuário será armazenada no sensor *resposta*. Ao perguntar a taxa de juros, a nova resposta do usuário será armazenada no mesmo sensor,

¹⁰ O texto original diz: Variables are used to keep track of the things that can change while a program is running. They are a bit like x or y in algebra, in that the values may not initially be known. Variables are not just used for numbers. They can also hold text, including whole sentences ('strings'), or the logical values 'true' or 'false'. For our quiz we would use variables to keep track of the player's score and the number of questions they attempt. Variables are like boxes, in that the computer can use them to store information that can be changed by the user, the program or by another variable.

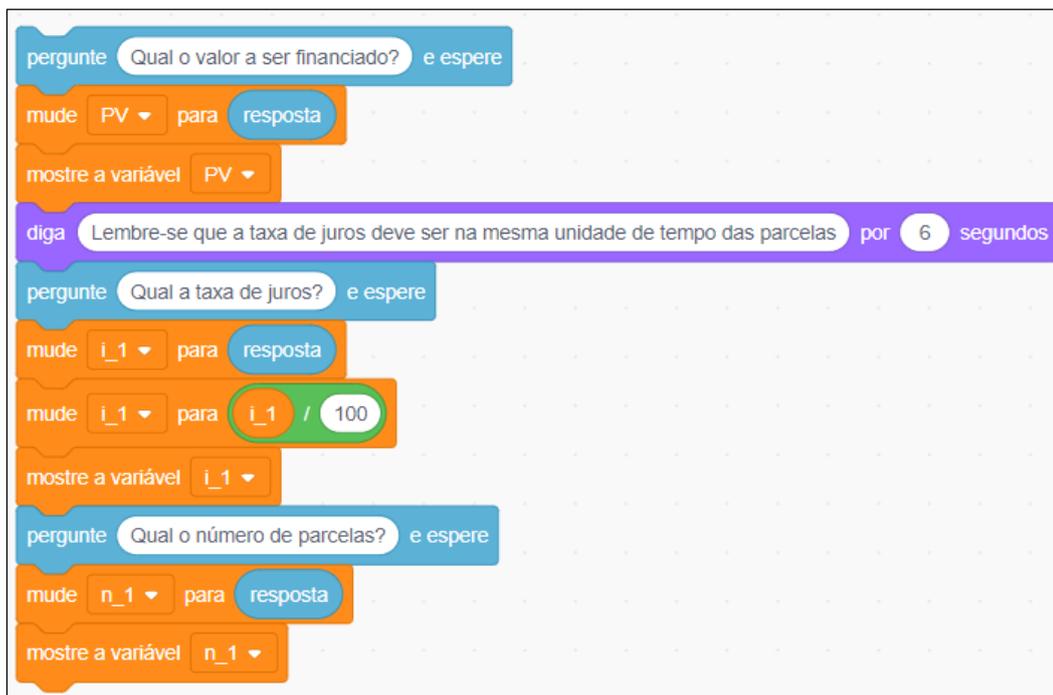
mantendo a última resposta guardada. Por isso é importante utilizar as variáveis para armazenar essas respostas.

Ao responder o valor a ser financiado, precisamos programar de maneira que essa resposta seja armazenada na variável PV, o Valor Presente do automóvel. Realizamos essa programação utilizando o bloco *mude PV para* conectando o sensor *resposta* a ele.

Neste momento não nos atentamos sobre as diferentes possibilidades para a entrada do valor da taxa de juros (forma unitária ou percentual, com ou sem o símbolo %) e nem para o fato de que o período da taxa e do número de parcelas devem coincidir (por exemplo, para o pagamento em 48 parcelas mensais, a taxa de juros deve ser mensal, e não anual). São detalhes e conceitos que poderão ser questionados durante o ato de programar, configurando como oportunidades para a discussão junto aos estudantes.

Dessa forma, decidimos inserir uma mensagem lembrando o usuário que a taxa de juros deve ser na mesma unidade de tempo das parcelas e que a taxa de juros seria inserida na forma percentual, programando de maneira que ela fosse mudada para a taxa unitária, ou seja, após mudar a variável i_1 para a resposta do usuário, precisamos mudá-la novamente para o valor informado, dividido por 100 (fato que também pode ser explorado junto aos estudantes), ou seja, usaremos novamente o bloco *mude para* e precisaremos conectar a ele o *operador de divisão*. O operador será preenchido com a variável i_1 no numerador e o número 100 no denominador. Por último, de forma análoga programamos para seja pedido o número de parcelas do financiamento, usando a variável n_1 . Portanto, a programação inicial é:

Figura 3: Programando para que o primeiro ator peça ao usuário os valores necessários ao cálculo. O procedimento para a alteração da taxa foi inserido



Fonte: Elaborada pelos autores, a partir do ambiente Scratch

A partir desse momento, podemos programar o cálculo do coeficiente. Como faremos isso? Note que, como o coeficiente depende da taxa de juros e do número de parcelas, que são variáveis, o coeficiente também será uma variável, uma vez que ele poderá ser alterado pelas variáveis anteriores. Portanto, criamos a variável Coeficiente_1 e também vamos defini-la recorrendo ao bloco *mude para*. Para programar o cálculo do coeficiente, é preciso analisar esse termo da equação de maneira minuciosa. Primeiramente, percebemos que se trata de uma razão, ou seja, teremos o numerador e o denominador; portanto, o primeiro passo será utilizar o *operador de divisão*.

Iniciando pelo numerador, percebemos que ele possui uma operação de subtração; logo, utilizaremos o *operador de subtração* para compor o numerador. O primeiro termo da subtração é uma soma, então, usaremos o *operador de soma* e o colocaremos no primeiro termo do operador de subtração. Precisamos fazer algumas observações: a primeira é chamar a atenção para a leitura que o *Scratch* faz dessa junção de blocos. Como o operador de adição está sobreposto aos outros operadores, seguido do operador de subtração e operador de divisão, respectivamente, o software calcula os termos de cada operador seguindo essa ordem, respeitando, portanto, a ordem da expressão algébrica em questão; a segunda observação é que para que o *Scratch* entenda que se deve calcular a expressão matemática na ordem correta, respeitando os parênteses, potências, etc.,

precisa-se sempre construir a programação “de baixo para cima” (do operador que ficará por baixo ao operador que ficará sobreposto aos outros), ou seja, dividir a expressão matemática em diversas partes e ir construindo-a passo a passo. Não podemos deixar de registrar a valiosa possibilidade de discutir detalhes como esses, tão presentes em sala de aula e que muitas vezes passam despercebidos.

Construídos os blocos que compõem o numerador, é hora de preenchê-los, lembrando que, quando informar a taxa de juros, é preciso atrelá-la a variável i_I . Agora é preciso elevar o operador soma à enésima potência. Clicando no conjunto de blocos de sensores no *Scratch*, você notará que não há a opção de potência enésima e é aqui que surge outra grande possibilidade de se explorar um pouco mais a Matemática.

Nós, autores deste trabalho, sentimos dificuldade quando nos deparamos com essa situação. À priori, exploramos o software em busca de uma alternativa. Sem sucesso, decidimos olhar para a Matemática em busca de uma solução e recorrendo à nossa base de conhecimento matemático, reescrevemos a expressão de forma equivalente, de tal forma que ela pudesse ser implementada no software, considerando sua sintaxe:

$$\begin{aligned}(1 + i)^n = ? \Rightarrow y = (1 + i)^n \Rightarrow \ln y = \ln (1 + i)^n \Rightarrow \ln y = n \cdot \ln (1 + i) \Rightarrow \\ \Rightarrow e^{\ln y} = e^{n \cdot \ln (1 + i)} \Rightarrow y = e^{n \cdot \ln (1 + i)}\end{aligned}$$

Então, conseguimos reescrever a expressão para deixá-la em termos de operadores que o *Scratch* reconhece. Ou seja, as coisas não estão todas prontas para o usuário, tendo sido necessário recorrer aos conhecimentos matemáticos que possuímos, o que analisamos como positivo diante dos objetivos educacionais também presentes na atividade. A nossa falta de sucesso na busca por alternativas com os blocos do *Scratch* não significa que não há possibilidades disso ser feito, muito pelo contrário; apenas queremos mostrar o passo a passo de como nós construímos e desenvolvemos a atividade.

Curiosamente, por distração, programamos o cálculo acima usando o comando *log*, ao invés de *ln*, considerando $y = e^{n \cdot \log (1 + i)}$; passamos algum tempo testando e reformulando nossa programação, pois o valor obtido não estava coincidindo com o valor pesquisado inicialmente, o que também destacamos como positivo para a produção de conhecimento matemático. O erro nos fez depurar sobre o conceito definidor de cada passo presente na construção dessa nova forma de reescrever a expressão $(1 + i)^n$, o que possibilitou discussões valiosas. Ainda que o estudante programe usando o *log*, a programação permitirá que ele reconheça o erro através da depuração e ressignifique o seu conhecimento sobre exponencial e logaritmo.

Desse modo, precisaremos alterar o primeiro termo da subtração, pois a expressão $(1 + i)^n$ será reescrita como $e^{n \cdot \ln(1+i)}$. Para isso, usaremos o *sensor e[^]*. Observe que o expoente da exponencial é a multiplicação de dois termos, logo, iremos juntar ao sensor anterior, o operador de multiplicação e substituir os valores de ambos os termos da multiplicação.

Note que o logaritmando é agora a adição de dois termos, por isso, juntaremos o *operador de adição* e preencheremos ambos os termos deste operador. Observe quantos conceitos e termos matemáticos foram explorados na programação apenas do numerador do coeficiente, ou seja, ao programar estamos produzindo conhecimento.

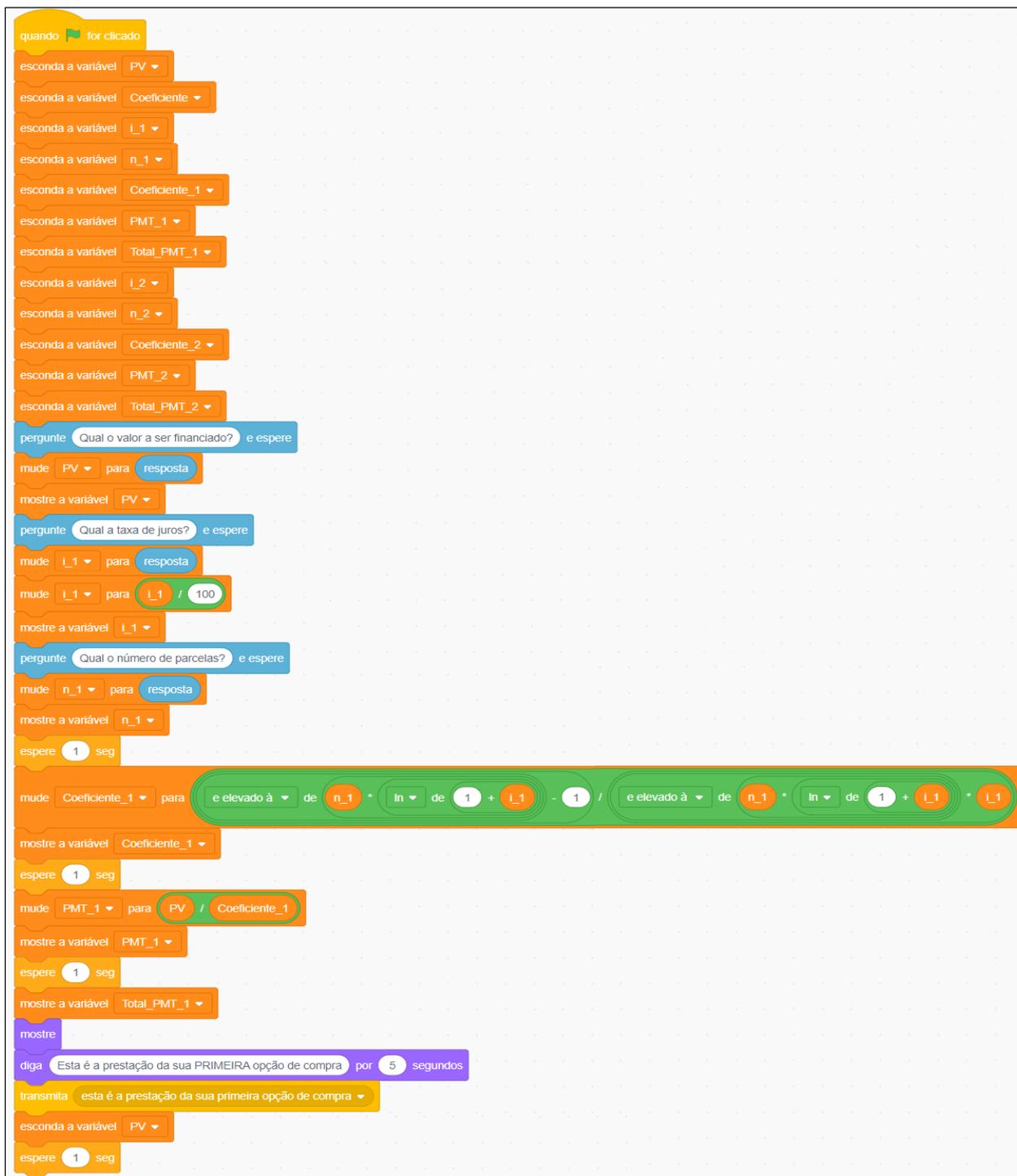
Continuando a programação do coeficiente, o próximo passo é programar a expressão do denominador, que seguirá o mesmo raciocínio do numerador, alterando apenas o primeiro operador que será colocado no denominador, sendo, nesse caso, o *operador de multiplicação*.

Solicitado o valor do PV e calculado o coeficiente, podemos, finalmente, fazer a programação do PMT e, novamente, fazendo-se a discussão sobre o conceito de variável e recorrendo a operadores matemáticos.

Após pedir os valores de PV, taxa de juros e número de parcelas ao usuário, tendo obtido o valor do coeficiente e da variável PMT, pedimos para que o primeiro ator escolhido mostre esses dois últimos valores na tela, dando ênfase ao valor da prestação relativa à primeira opção de compra. Essa ação está relacionada com a aparência e, assim, usaremos o bloco *diga por segundos*. Optamos por escrever a frase que será dita pelo primeiro ator: *Esta é a prestação da sua PRIMEIRA opção de compra*, aparecendo na tela por 5 segundos.

É habitual, no *Scratch*, iniciar a programação com algum bloco de evento. Optamos por iniciar com o bloco *quando clicar em bandeira verde*, porém, salientamos que fica a critério de cada programador essa escolha. Esses blocos não possuem entrada de encaixe em sua parte superior, exatamente para indicar que se refere a algo que está iniciando. A seguir, é possível observar como está nossa programação até o momento. Devido as dimensões da página e tamanho do algoritmo não é possível visualizar com nitidez os blocos utilizados, mas a programação pode ser observada através do link <https://scratch.mit.edu/projects/188251121/>, no modo “Ver Interior” no ambiente *Scratch*.

Figura 4: Programação para a ação do primeiro ator



Fonte: Elaborada pelos autores, a partir do ambiente Scratch

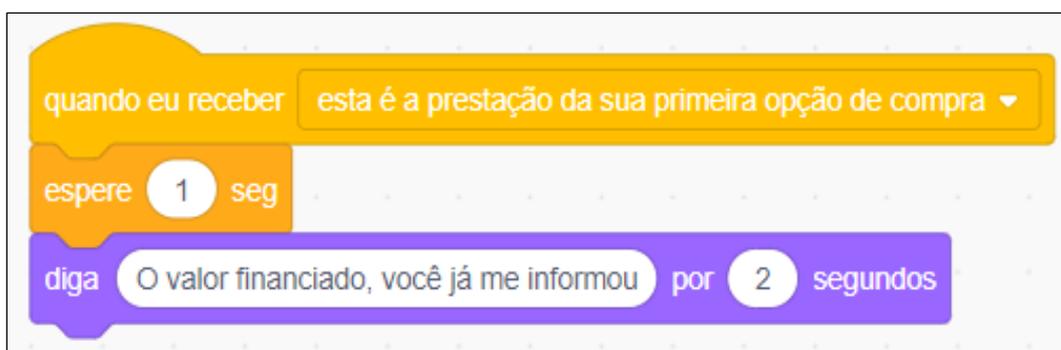
Criada a programação do primeiro ator, que indica os resultados relativos à primeira opção de financiamento, precisamos criar a programação referente à segunda opção e, para isso, é necessário decidir se o programa continuará com o mesmo ator ou se será criado um segundo. Nós optamos por criar um novo ator e iniciamos novamente a programação.

Precisamos pensar quando o segundo ator começará a interagir com o usuário, que será quando terminar a execução da programação referente ao primeiro financiamento. Por isso, o evento que escolhemos para iniciar a programação do segundo financiamento, dentro das opções presentes no *Scratch*, é *quando receber mensagem*. Então, voltamos à programação do primeiro ator e acrescentamos no final o envio da mensagem ao segundo ator: *Esta é a prestação da sua PRIMEIRA opção de compra*, dando início ao estudo do segundo financiamento. Acrescentamos o bloco de *espera* para que a mensagem seja enviada e haja uma pausa de 1 segundo para iniciar a execução da segunda programação.

Esses detalhes são construídos durante a programação; não são ações possíveis de serem previstas e esse ir e vir estimula o pensamento, de forma estruturada, além de provocar uma organização de ideias por quem está programando. Essas características são inerentes ao que entendemos por Pensamento Computacional, valorizando o raciocínio e o pensar sobre o pensar, presentes na rotina de programação.

Dando continuidade, o segundo ator precisa interagir com o usuário, solicitando as variáveis de entrada para o segundo financiamento. Observe que o PV será o mesmo do primeiro caso e, logo, não há necessidade de solicitá-lo novamente. Optamos, assim, por enviar uma mensagem ao usuário lembrando-o que o PV já foi informado. Para isso, usamos o bloco *diga por 2 segundos* e o início da nossa programação foi composto pelos blocos:

Figura 5: Início da programação do segundo ator



Fonte: Elaborada pelos autores, a partir do ambiente Scratch

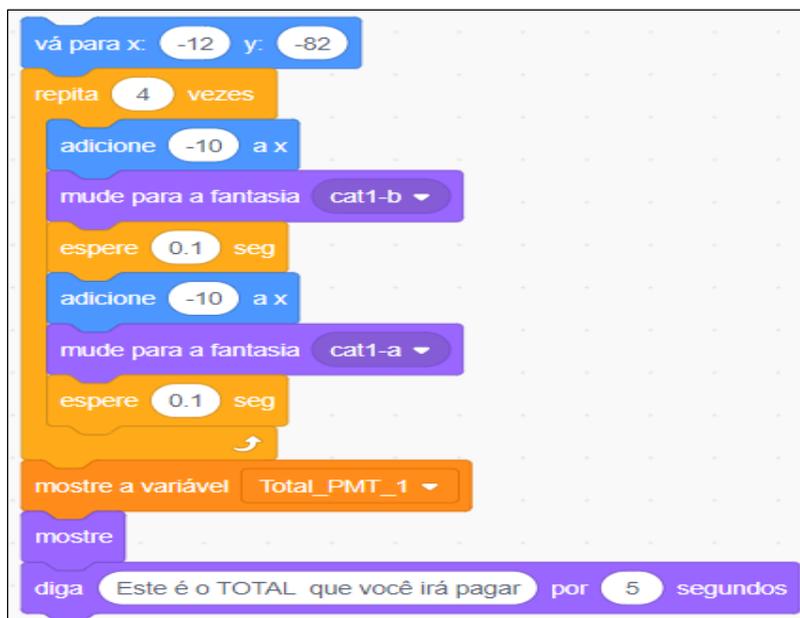
O próximo passo é solicitar que o usuário indique a taxa de juros e o número de parcelas da segunda opção de financiamento e, então, calcule o coeficiente e o PMT com esses novos valores, indicando qual é a prestação da segunda opção de compra. Ou seja, a programação do primeiro financiamento se repete, sendo possível, inclusive, copiar o anterior, fazendo ajustes quando necessário.

Até aqui, a primeira parte da atividade foi realizada. Agora, precisamos programar a segunda parte, que se refere à tomada de decisão. O cálculo do montante poderia ter sido programado imediatamente após o cálculo das prestações, mas nós optamos por criar um terceiro ator, programando para que ele direcione o cálculo do montante total. Para isso, novamente, precisamos determinar quando iniciará a execução da próxima programação, podendo utilizar o mesmo recurso usado anteriormente.

Por consequência, a programação do terceiro ator se iniciará com o bloco *quando receber* com a mensagem: *Esta é a prestação da sua SEGUNDA opção de compra*. É nesse momento que o montante que será pago nas duas opções de financiamento será calculado. Esse montante é obtido multiplicando o PMT pelo número de parcelas. Lembre-se que tanto o PMT quanto o número de parcelas são variáveis e, logo, o montante também será uma variável, que nós chamamos de TOTAL_PMT_1 e TOTAL_PMT_2, respectivamente, para as duas opções de compra, utilizando os blocos *mude para e operador de multiplicação*.

Em seguida, queremos que o terceiro ator indique ao primeiro o total que será pago, movendo-se até ele e mostrando o valor. Programamos esse trajeto percorrido pelo terceiro ator, da seguinte maneira:

Figura 6: Comandos usados para a movimentação do terceiro ator, no sentido negativo do eixo das abscissas



Fonte: Elaborada pelos autores, a partir do ambiente Scratch

Como na nossa programação, o ator 1 está à esquerda do usuário e desejamos que o terceiro ator caminhe até ele, esse terceiro ator precisará andar no sentido negativo do

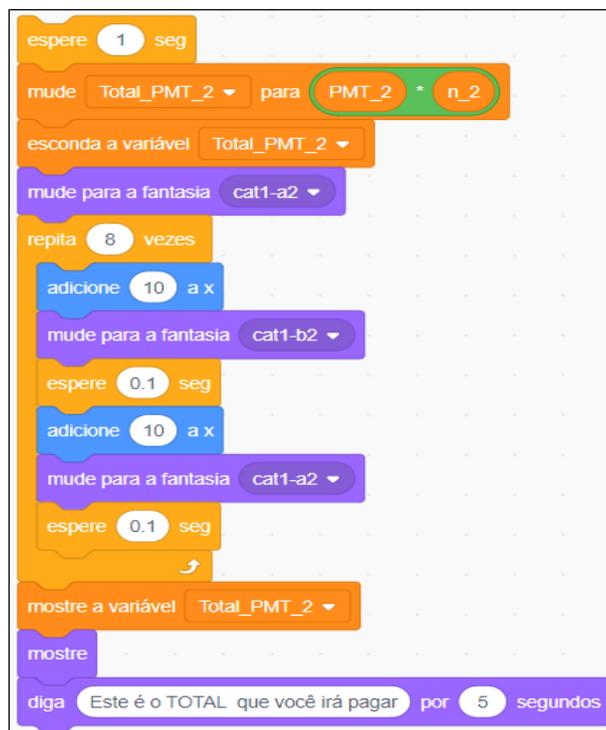
eixo das abscissas; por isso, em seguida, colocamos o bloco que adiciona -10 à posição anterior x . Na *aba fantasias* há a imagem do ator selecionado em duas posições, a e b . O uso em conjunto dessas duas posições, possibilita visualizar o ator caminhando, pois mostra o movimento das suas pernas. Por isso, após adicionar -10 à posição x (para que o ator ande para o lado esquerdo do usuário), pedimos para trocar a fantasia e , assim, sucessivamente, até chegar próximo ao primeiro ator. Veja que esse “sucessivamente” será a mescla entre a fantasia a (fantasia inicial) e a fantasia b , possibilitando que usemos o comando *repita* pela quantidade de vezes que achamos necessário. Entre a troca de fantasia e os passos que o ator dará, inserimos um bloco de espera de 0.1 segundos para que seja visível, ao usuário, o caminhar do ator.

Sabendo que a posição inicial do terceiro ator pode variar e isso acarretará em mudanças na quantidade de repetições dos comandos que indicam a troca de fantasia e os passos do ator à esquerda, definimos, no início, o ponto de origem dele $(-12, -82)$, abrindo outra possibilidade de exploração de conceitos matemáticos, pois para usar esse bloco é necessário que o estudante compreenda o que significam as coordenadas cartesianas de um ponto, a localização de pontos no plano e as partes positivas e negativas de um eixo. Mais uma vez, esse entendimento e a necessidade de inserir esse bloco surgiu decorrente de um pensar sobre a ação, da depuração e organização das ideias.

Seguindo o mesmo raciocínio, programamos o terceiro ator para caminhar até o segundo (que está à sua direita) e indicar para ele o total que será pago, levando em consideração que, nesse caso, o terceiro ator caminhará no sentido positivo do eixo das abscissas e deverá caminhar o dobro da distância já percorrida para a esquerda.

Entre a programação do caminhar no sentido negativo e positivo do eixo das abscissas, inserimos o comando de esperar 1 segundo, para que seja dada uma pausa que permita ao usuário compreender as dinâmicas das ações. Acompanhe como ficou a programação:

Figura 7: Comandos usados para a movimentação do terceiro ator, no sentido positivo do eixo das abscissas



Fonte: Elaborada pelos autores, a partir do ambiente Scratch

Observe que foi preciso aplicar uma reflexão horizontal no terceiro ator, devido à necessidade de que ele estivesse virado para o lado direito do usuário, o que também abre possibilidades para explorar mais esse conceito com os estudantes. Em sala de aula, indicamos que o professor questione ao máximo sobre as possibilidades em como colocar o ator na direção correta para caminhar em direção ao segundo ator, para que eles conjecturem e possam perceber os conceitos matemáticos que eles possuem sobre reflexão horizontal, vertical ou outros relacionados.

Ainda seguindo o mesmo raciocínio, programamos para que o terceiro ator, caminhando, volte à posição inicial e se direcione ao usuário. O algoritmo criado para esse movimento pode ser observado na Figura 8. Salientamos que esse momento da programação foi realizado a partir de uma escolha nossa para tornar o programa criado mais dinâmico e interativo com o usuário.

Figura 8: Comandos usados para que o terceiro ator volte à posição inicial

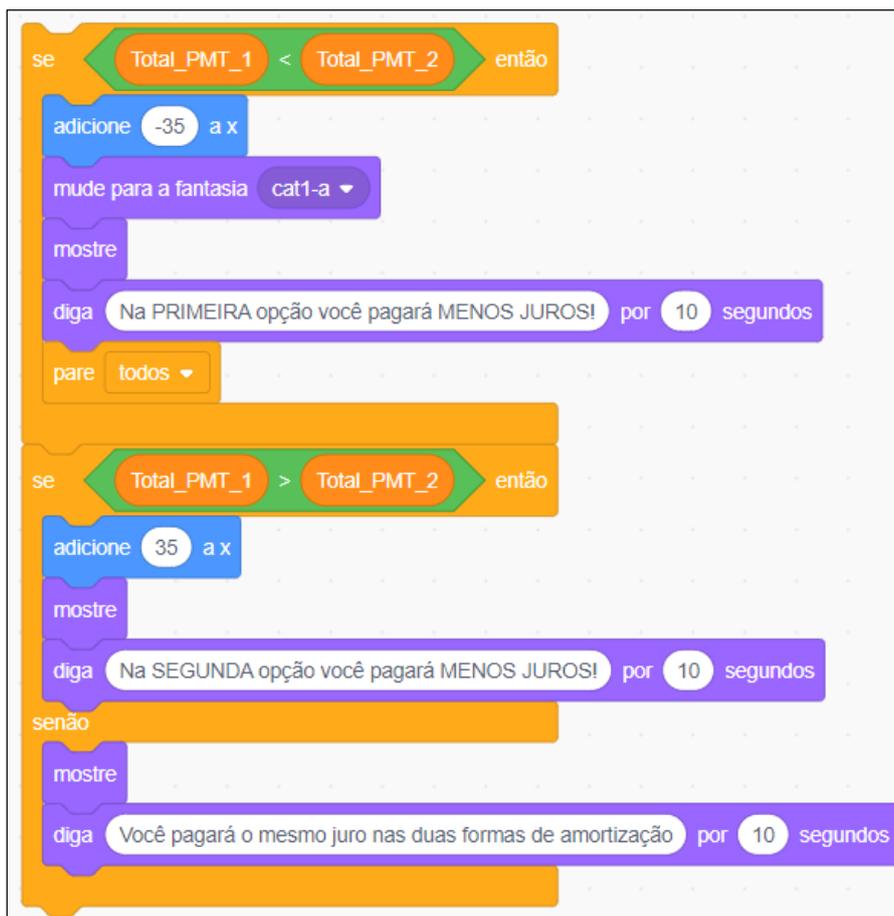


Fonte: Elaborada pelos autores, a partir do ambiente Scratch

Iniciamos, então, a programação para a tomada de decisão quanto à proposta de financiamento que terá o menor valor a ser pago em juros, o que está diretamente ligado ao menor montante a ser pago. Para isso, recorreremos ao bloco de controle “*se, então*” por se tratar de uma condição. Se o Total_PMT_1 for menor que o Total_PMT_2, significa que o montante do primeiro financiamento terá menos juros. Dessa maneira, programamos o terceiro ator para se dirigir ao primeiro e dizer que a primeira opção é a que envolve menos juros, terminando a programação uma vez que a tomada de decisão aconteceu. Note que aqui também foi necessário alterar a fantasia do ator, conforme já mencionamos.

Se o Total_PMT_1 não for menor que o Total_PMT_2, precisamos criar outra condição, indicando que a segunda opção de financiamento terá menos juros. Porém, pode acontecer de ambos os financiamentos terem o mesmo montante e, conseqüentemente, o mesmo total de juros a pagar. Nesse caso, precisaremos criar uma terceira condição ou utilizar o bloco de controle *se – então - senão* para as duas últimas condições, como pode ser observado na Figura 9.

Figura 9: Programação da tomada de decisão reformulada



Fonte: Elaborada pelos autores, a partir do ambiente Scratch

Assim, finaliza-se a atividade, observando a oportunidade de, em sala de aula, gerar discussões sobre várias questões voltadas à Educação Financeira.

5 Implicações para a promoção da Educação Financeira em sala de aula

Como professores, visualizamos inúmeras possibilidades para o trabalho com atividades como a que expomos aqui, nas salas de aula da Educação Básica, gerando discussões importantes tanto para a exploração de conceitos matemáticos, quanto para a Educação Financeira.

Para ilustrar, citamos dois casos. Um em que é dada uma entrada e, assim, o valor pago em juros é menor em virtude da diminuição do valor financiado; simulações podem mostrar que os juros decaem consideravelmente à medida em que o valor da entrada aumenta. Outro caso já poderia mostrar o contrário, que os juros podem aumentar consideravelmente quando existe uma carência para o início dos pagamentos, situação comum no final do ano, em que o primeiro pagamento é postergado para o ano seguinte,

geralmente após a festividade popular do carnaval. Nesses casos, o valor presente precisará ser corrigido para, só então, dar início aos cálculos do modelo matemático descrito na atividade. Assim, vemos a possibilidade de intensificar a discussão sobre o valor do dinheiro ao longo do tempo, em conjunto com os cálculos já descritos na atividade.

Ao simular financiamentos de outros bens de consumo que sejam de interesse dos estudantes, o professor poderá promover uma discussão sobre algumas reflexões importantes, desde o questionamento sobre a necessidade de comprar, passando pelas modalidades para se realizar um financiamento e o planejamento financeiro para que se possa honrar os pagamentos devidos, discussões essas que parecem ausentes na sociedade, quando observamos os altos índices de endividamento familiar nos últimos anos. Em fevereiro de 2019, 49,9% da população brasileira possuíam algum tipo de dívida¹¹.

Voltando à atividade apresentada, em sala de aula o professor poderia gerar outros questionamentos importantes, como: Para a tomada de decisão, basta olhar para o montante final que será pago? Essa análise é suficiente? Que outras variáveis precisam ser levadas em consideração, para que o indivíduo não fique endividado? E quanto à decisão de poupar, ao invés de financiar ou emprestar? É fato que, muitas vezes, financiar ou emprestar pode ser necessário diante de imprevistos, projetos que não podem esperar ou outras possíveis razões. Mas guardar dinheiro também pode ser possível em muitas situações, inclusive para a compra de um automóvel.

Uma possibilidade de expansão da atividade proposta é fazer uma simulação do tempo necessário para adquirir o automóvel, caso a pessoa guarde exatamente o valor que seria pago através das prestações, ainda que em aplicações de baixo rendimento, como a poupança. Dependendo da taxa de juros que se pagaria no financiamento, a diferença pode ser muito significativa, pois os juros, mesmo sendo baixos, atuam a favor da pessoa, aumentando o seu capital aplicado. Pode ser também uma oportunidade valiosa para falar sobre o planejamento para a velhice e convidar os estudantes a pensar e pesquisar sobre essa possibilidade.

Um momento interessante para propor aos estudantes a atividade apresentada neste artigo é no Ensino Médio, quando são estudadas algumas funções como a

¹¹ Em janeiro de 2019 o percentual de famílias endividadas era de 49,9%, sendo 18,7% do total com contas em atraso. Dados fornecidos pela PEIC – Pesquisa de Endividamento e Inadimplência do Consumidor. Disponível em: <http://www.fecomercio.com.br/pesquisas/indice/peic>. Acesso em: 19 fev. 2019.

exponencial e a logarítmica. Além da exploração dos conceitos ligados aos cálculos envolvidos nos financiamentos, também poderiam ser estudados os sistemas de amortização de dívidas, abrindo os valores das prestações para a compreensão de como os juros e as amortizações se compõem nas sequências uniformes de capitais. São conteúdos e competências apontados na Base Nacional Comum Curricular (BNCC) para o Ensino Médio, aprovada no final de 2018.

Entendemos que em qualquer nível de ensino são bem-vindas as iniciativas que rompem as rotinas de sala de aula, muitas vezes pautadas em exercícios de fixação e alheias às situações reais. Atividades como a que propomos aproximam o estudante de situações financeiras vividas pelos cidadãos, abrindo caminho para uma abordagem que favorece não apenas a aprendizagem da Matemática Financeira, mas também a autonomia do estudante e a tomada consciente de decisão. A parceria com o desenvolvimento do Pensamento Computacional, propiciada através do ambiente Scratch, mostrou-se valiosa para desenvolver habilidade importantes para a tomada de decisão, conforme discutimos a seguir.

6 Os conceitos da definição operacional de Pensamento Computacional observados no desenvolvimento da atividade

Com base na definição operacional de Pensamento Computacional apresentada na introdução deste artigo, apresentamos a seguir o nosso entendimento de como tais conceitos se fizeram presentes na atividade apresentada.

A *coleta de dados* foi observada na busca por alternativas para o financiamento do automóvel, bem como na pesquisa pelo modelo matemático que permite obter o valor das prestações e calcular o montante que será pago.

A *análise dos dados* se deu através da compreensão sobre os financiamentos, a expressão matemática e a identificação das variáveis envolvidas na resolução do problema.

A *representação dos dados* esteve presente em nossas reflexões ao buscar compreender o processo de financiamento de automóveis e nas discussões matemáticas, levando-nos a recorrer à elaboração de tabelas e diagramas de fluxo de caixa, para a compreensão dos modelos de financiamentos estudados. Este processo foi importante para a compreensão matemática da questão e, conseqüentemente, para a programação dos cálculos no software.

A *decomposição do problema* foi visivelmente necessária, principalmente para a definição inicial das variáveis e, depois, para a obtenção do coeficiente de financiamento, o cálculo do montante a ser pago em cada caso, a elaboração final para o cruzamento das informações e parecer com a tomada de decisão.

A *abstração* foi verificada diante da necessidade de analisar e identificar as características essenciais em uma proposta, bem como as relações entre elas, a ponto de compreender todo o processo e conseguir estender essa compreensão à outras propostas de financiamento. Embora a taxa de juros fosse um pouco maior na segunda opção, em virtude do número de parcelas ser reduzido, esta era a mais viável, dado que a outra opção envolvia uma taxa menor, porém por um período de tempo significativamente maior.

No caso dos *algoritmos e procedimentos*, estes são observados em todo momento, evidenciados na programação com o Scratch, que requer fortemente um detalhamento minucioso de cada ação que se deseja realizar.

A *automação* é observada através da realização dos cálculos envolvidos em cada opção de compra, substituindo o que seria realizado de forma repetitiva e tediosa.

Por fim, observamos que na atividade também está presente o conceito de *simulação*, através da execução de outros tipos de financiamentos, usando o modelo criado.

A *paralelização* não esteve presente em nossa programação, porém, entendemos que se após a entrada das variáveis de ambos os financiamentos, programássemos o cálculo dos dois montantes em um único usuário e ao mesmo tempo, estaríamos trabalhando com a paralelização.

Todos esses conceitos observados na atividade mostram, em consonância com as ideias já relatadas sobre o desenvolvimento do Pensamento Computacional, a valorização do raciocínio no processo de resolução do problema, em detrimento das abordagens em que apenas os conceitos matemáticos são valorizados e, muitas vezes, de maneira tediosa nas salas de aula.

7 Considerações finais

Entendemos que o Pensamento Computacional está relacionado com o modo como o indivíduo desenvolve seu pensamento de forma estratégica e organizada sobre os problemas, valorizando o raciocínio e a combinação de diferentes formas de investigação e expressão, em especial as inerentes à computação. Em relação aos nove conceitos

elaborados pela ISTE e CSTA na definição operacional de Pensamento Computacional, existe a possibilidade destes e ainda outros conceitos estarem envolvidos em atividades dessa natureza.

No caso da atividade que realizamos, percebemos que, ao discutir o processo de tomada de decisão na escolha de um financiamento para a compra de um automóvel, foi pertinente a criação de um algoritmo, levando em conta as diferentes variáveis envolvidas, o que demandou a coleta de dados advindos de diferentes instituições financeiras, a abstração do problema em conjunto com a busca de soluções, além da programação para que o software calculasse e comparasse os montantes devidos em cada caso, apresentando a solução matematicamente mais viável.

Três fatores ficaram visíveis na atividade: as inúmeras possibilidades de ensino subjacentes ao uso de linguagem de programação; a presença do Pensamento Computacional nesse ambiente de programação; e a contribuição do desenvolvimento do Pensamento Computacional para a exploração, verificação e ressignificação de conceitos matemáticos.

Percebemos também que o estudante pode ser engajado em reflexões posteriores à tomada de decisão; por exemplo, pode acontecer que a opção em que se paga menos juros contemple o pagamento de uma prestação mensal superior às possibilidades de uma pessoa e, então, não seja possível de se encaixar no orçamento pessoal. Também pode ser necessária a busca por outras alternativas de pagamento, sendo necessário voltar à coleta de dados e à pesquisa de campo, estando uma pessoa insatisfeita com as opções analisadas, o que é extremamente positivo para a conscientização dos estudantes sobre a necessidade de pesquisar sempre, antes de uma tomada de decisão que comprometerá a renda familiar durante meses ou anos.

É essencial que as possibilidades de reflexão com os estudantes sobre a Educação Financeira se façam presentes em todo o processo de desenvolvimento da atividade proposta. Em nossa atividade, essa reflexão foi potencializada pelo desenvolvimento do Pensamento Computacional, no ambiente *Scratch*. É possível que outras discussões e reflexões envolvendo financiamentos surjam em decorrência desta atividade, o que consideramos mais um ganho. É bom e desejável que isso aconteça e que elas sejam exploradas com os estudantes, ampliando ainda mais o nível de conhecimento e criticidade diante da necessidade de tomada de decisão sobre problemas financeiros.

Referências

- BARONI, A. K. C.; MALTEMPI, M. V. Os espaços da Educação Financeira na formação do professor de Matemática em uma instituição federal de São Paulo. **Revemop**, Ouro Preto, v. 1, n. 2, p. 248-265, maio./ago. 2019.
- BRASIL, Série Cidadania Financeira: **Estudos sobre Educação, Proteção e Inclusão**. 2. ed. Brasília: Banco Central do Brasil, Novembro de 2015.
- BERRY, M. **Computing in the national curriculum**: a guide for primary teachers. 1ed. Bedford: Newnorth Print, 2013.
- CAMPOS, C. R.; TEIXEIRA, J.; COUTINHO, C. Q. S. Reflexões sobre a educação financeira e suas interfaces com a educação matemática e a educação crítica. **Educação Matemática Pesquisa**, São Paulo, v.17, n.3, p. 556-577, 2015.
- CUNHA, C. L.; LAUDARES, J. B. Resolução de Problemas na Matemática Financeira para Tratamento de Questões da Educação Financeira no Ensino Médio. **Bolema**, Rio Claro, v. 31, n. 58, p. 659-678. 2017.
- GADANIDIS, G. Coding as a Trojan Horse for mathematics education reform. **Journal of Computers in Mathematics and Science Teaching**, Chesapeake, v. 34, n. 2, p. 155-173, abr. 2015.
- KISTEMANN JR, M. A. **Sobre a produção de significados e a tomada de decisão de indivíduos-consumidores**. 2011. Tese (Doutorado em Educação Matemática) - Instituto de Geociências e Ciências Exatas, Universidade Estadual Paulista, Rio Claro, 2011.
- MANILLA, L. *et al.* **Computational thinking in K-9 education**. *In*: Proceedings of the working group reports of the 2014 on innovation & technology in computer science education conference. ACM, 2014. p. 1-29.
- MITCH RESNICK. **Vamos ensinar crianças a escrever códigos**. TEDxBeaconStreet. Brookline, MA-Estados Unidos, 2012. Disponível em: https://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code?language=pt-br#t-754207. Acesso em 23 maio. 2019.
- PAPERT, S. An exploration in the space of mathematics educations. **International Journal of Computers for Mathematical Learning**, Boston, v. 1, n. 1, p. 95-123, 1996.
- PUCCINI, A. de L. **Matemática Financeira**: Objetiva e Aplicada. 6. ed. São Paulo: Saraiva, 1999.
- RESNICK, M. *et al.* Scratch: Programming for all. **Communications of the ACM**, Nova York, v. 52, n. 11, p. 60-67. 2009.
- SILVA, A. M.; POWELL, A. B. Um programa de Educação Financeira para a Matemática Escolar da Educação Básica. *In*: ENCONTRO NACIONAL DE EDUCAÇÃO MATEMÁTICA, 11., 2013, Guarapuava. **Anais...** Curitiba: SBEM-PR, 2013. p. 01-17. Disponível em: http://sbem.iuri0094.hospedagemdesites.ws/anais/XIENEM/pdf/2675_2166_ID.pdf. Acesso em: 03 jan. 2019.

DOI: <http://dx.doi.org/10.33238/ReBECeM.2019.v.3.n.2.22627>

WING, J. M. **Computational Thinking: What and Why.** Recuperado de <http://www.cs.cmu.edu/link/researchnotebook-computational-thinking-what-and-why>. Cambridge, 2011. Acesso em 30 maio. 2019.

Recebido em: 16 de junho de 2019.

Aceito em: 08 de agosto de 2019.