

ENSINO & PESQUISA

ISSN 2359-4381

O uso do Scratch no ensino de programação

DOI: <https://doi.org/10.33871/23594381.2025.23.1.9141>

Rudolph dos Santos Gomes Pereira¹, Paulo Roberto Anastácio², Willian Damin³, Barbara Nivalda Palharini Alvim Sousa⁴

Resumo: O objetivo deste é artigo é apresentar parte dos resultados encontrados em uma pesquisa de mestrado desenvolvida com alunos do primeiro ano do curso de Ciência da Computação de uma Universidade pública do Norte do Paraná, no qual foram abordados os conceitos de lógica de programação, estruturas de decisão e de repetição por meio da plataforma de programação em blocos Scratch. No desenvolvimento desta pesquisa, apresenta-se uma atividade para o uso da ferramenta Scratch e o desenvolvimento do Pensamento Computacional como modo de auxiliar o ensino de programação. Os resultados apresentados pelos alunos na atividade de Estruturas Lógicas de Condição foram analisados qualitativamente e indicam o desenvolvimento do Pensamento Computacional ao utilizá-lo para programar, tomar de decisão, elaborar e solucionar problemas que envolvem algoritmos.

Palavras-chaves: Ensino de Programação. Scratch. Pensamento Computacional.

The use of Scratch in programming education

Abstract: The purpose of this article is to present some of the findings from a master's research conducted with first-year Computer Science students at a public university in northern Paraná, Brazil. The research addressed the concepts of programming logic, decision structures, and repetition structures using the block programming platform Scratch. During this research, an activity using the Scratch tool was presented to facilitate the teaching of programming through the development of Computational Thinking. The results shown by the students in the Logical Condition Structures activity were qualitatively analyzed and indicate the development of Computational Thinking when used for programming, decision-making, and problem-solving involving algorithms.

Keywords: Teaching Programming. Scratch. Computational Thinking

Introdução

A capacidade de solucionar problemas é uma habilidade matemática que envolve algumas habilidades como raciocínio lógico, levantamento de hipóteses, planejamento de ações e formulação de problemas, as quais têm papel fundamental na formação integral do indivíduo, preparando-o para viver em sociedade e tomar decisões. Dentre essas habilidades, o raciocínio lógico é fundamental no processo de ensino de programação e se faz presente em qualquer área do conhecimento, sendo desenvolvido quando o

¹ Doutor em Educação, rudolph.matematica@gmail.com, <https://orcid.org/0000-0003-0504-7329>.

² Mestre em Ensino, paulinho.r.a@gmail.com, <https://orcid.org/0009-0004-0326-9731>.

³ Doutor em Ensino de Ciência e Tecnologia, wdamin@uenp.edu.br, <https://orcid.org/0000-0002-6795-9772>.

⁴ Doutora em Ensino de Ciências e Educação Matemática, barbara.palharini@uenp.edu.br, <https://orcid.org/0000-0002-3712-9663>

indivíduo se depara com situações-problema e necessita organizar o pensamento detalhadamente, buscando uma solução para revolvê-los (Tretin *et al.*, 2019).

Neste aspecto, as dificuldades comumente apontadas pelos alunos como por exemplo, a dificuldade em utilizar raciocínios lógicos, estão relacionadas aos conceitos fundamentais da Computação, os quais têm o papel de promover o desenvolvimento de habilidades e competências fundamentais para auxiliar na tomada de decisão e resolução de problemas (Brackmann, 2017).

Segundo os dados da *Programme for International Student Assessment* (Oecd, 2018), grande parte dos alunos da Educação Básica brasileira apresenta dificuldades na resolução de problemas nas diversas áreas do conhecimento, com destaque para o raciocínio lógico, a contagem, a capacidade de abstrair informações e o desenvolvimento do pensamento crítico.

Bittencourt *et al.* (2013) ressaltam que as dificuldades econtradas pelos alunos na disciplina de programação, se dá pela necessidade de aprender diversos conceitos em pouco tempo, desenvolver raciocínio lógico e capacidades específicas da programação como o Pensamento Computacional (PC). Como forma de auxiliar nas dificuldades encontradas nesses cursos, pode ser adequado a utilização de ferramentas que evidenciem o processo de solucionar problemas e estimulem o Pensamento Computacional por meio da programação (Mestre *et al.*, 2015).

O Pensamento Computacional é apontado por Mestre *et al.* (2015) como o domínio de habilidades diretamente relacionadas à resolução de problemas, envolvendo a capacidade de ler e interpretar textos além de compreender as situações contidas nos problemas e transcrevê-las em modelos matemáticos. Assim, para promover o Pensamento Computacional em sala de aula, é necessário incorporar estratégias no processo de ensino no qual o aluno participa da construção do processo de forma flexível, interligada e híbrida, podendo desfrutar dos benefícios de se utilizar as ferramentas educacionais como o *Scratch* (Brackmann, 2017).

Uma das vantagens dessa ferramenta é que ela não exige dos estudantes a sintaxe de uma linguagem de programação textual, uma vez que a ferramenta utiliza uma linguagem de programação visual por blocos lógicos, facilitando o ensino de programação.

Para que o aluno desenvolva e solucione um projeto utilizando o *Scratch*, é necessário que o mesmo conclua as quatro etapas do Pensamento Computacional: a decomposição, os padrões, a abstração e o algoritmo (Raabe *et al.*, 2018). Tais etapas são

responsáveis por garantir que o aluno seja capaz de dividir o problema em passos menores e concretizá-los por meio da utilização dos blocos de programação contidos na ferramenta, estimulando o aluno a utilizar a lógica matemática, estruturas de decisão, repetição e desenvolver o Pensamento Computacional (Tretin *et al.*, 2019).

O objetivo deste artigo é apresentar os resultados encontrados em uma atividade de Estruturas Lógicas de Condição, a partir do uso do Scratch como recurso tecnológico no ensino de programação, tendo em vista o desenvolvimento do Pensamento Computacional e suas etapas: a Decomposição, o Reconhecimento de Padrões, a Abstração e o Algoritmo.

Ensino de Programação

Valente (2016) afirma que programar é um trabalho que objetiva fornecer instruções ao computador a fim de solucionar um conjunto de problemas tradicionais. Aprender uma linguagem de programação é uma atividade desafiadora, contudo, tornar a programação disponível para um maior número de indivíduos é algo significativo, já que pode estimular várias capacidades cognitivas como pensar logicamente e solucionar problemas (Scaico *et al.*, 2012).

De acordo com Forbellone e Ebrspacher (2005) o raciocínio é a forma mais complexa do pensamento, sendo que a lógica estuda a “correção do raciocínio”, organizando o pensamento. Assim, segundo esses autores apontam, a importância da lógica no dia a dia das pessoas, pois essa prática está presente no pensar, falar, escrever e agir corretamente, ordenando os pensamentos a partir do uso da lógica. Para que ocorra a aprendizagem da programação é fundamental que o aluno tenha a capacidade de abstrair, para que assim consiga compreender, propor e solucionar problemas por meio de raciocínios lógicos. Os autores ainda propõem a utilização de ferramentas educacionais que estimulem os alunos por meio da ludicidade e usabilidade, possibilitando o encontro de subsídios para a ensino e a aprendizagem de programação.

Zaharija, Mladenovic e Boljat (2013) defendem que o ensino de programação poderia estar incluso nas disciplinas do Ensino Fundamental e Médio, possibilitando integrar às metodologias curriculares, ferramentas auxiliadoras como Scratch, que objetivam facilitar a aprendizagem de programação.

Diante da demanda de profissionais relacionados as áreas da computação, a falta de profissionais capacitados e as dificuldades encontradas no ensino de programação, surge a necessidade de motivar crianças e adolescentes a desenvolverem novas formas de

pesamento como a Lógica de Programação (Ventura, 2018). Almejando que em um futuro próximo estes alunos não sejam apenas usuários das tecnologias, mas também desenvolvedores dessas, capazes de solucionar problemas e pensar computacionalmente (Ventura, 2018).

O aprendizado da programação, possibilita ao aluno o desenvolvimento de habilidades, abstração, reflexão e planejamento, garantindo a este a possibilidades de compreender e solucionar problemas dentro e fora das salas de aulas (Ventura, 2018).

Blikstein (2008) ressalta que o ensino de programação não é uma prática exclusiva dos cursos da área da Computação. De acordo com as exigências da atualidade, os profissionais de diversas áreas como por exemplo economia e ciências, devem saber utilizar ferramentas a fim de criar modelos computacionais que facilitem, aprimorem e aperfeiçoem trabalhos e atividades do cotidiano (Blikstein, 2008).

Pereira *et al.* (2012) revelam em suas pesquisas que a disciplina de lógica de programação é responsável pela reprovação ou desistências de 75% dos alunos da Área da Computação, pois apresentam dificuldades nos exercícios de raciocínio lógico e matemático, apontando o sistema educacional como responsável por este obstáculo, pois o aprendizado ocorre pela memorização de conteúdos e os alunos não desenvolvem aspectos relacionados a autonomia e criatividade.

Albertin e Albertin (2008) afirmam que dominar a habilidade da programação é tão importante quanto foi aprender a ler no século passado, nessa perspectiva o autor almeja que o usuário não apenas utilize os aplicativos e jogos do seu celular, mas ajude a desenvolve-los.

A sociedade atualmente passou a utilizar as tecnologias de informação no dia a dia, e para atender a essa demanda as pessoas buscam por melhorias em *softwares* e *hardwares* a fim de agilizar tarefas e aperfeiçoar desempenhos. Deste modo, é essencial incentivar pessoas da área da Computação a aprender programação, já que esta é a base para o desenvolvimento de novos *softwares* que busca atender a demanda da população (Ventura, 2018). E para que profissionais e alunos dominem e utilizem a Programação, se faz necessário o estudo da Lógica de Programação para o desenvolvimento de algoritmos coerentes e válidos (Forbellone; Eberspächer, 2005).

Um algoritmo pode ser estabelecido como uma sequência de passos que visam atingir um objetivo delineado, essa lógica investiga a ordenação do pensamento e do raciocínio lógico podendo ser determinada como a “arte de pensar” e corrigir o pensamento constantemente (Forbellone; Eberspächer, 2005).

Quanto aos os cursos da Computação, há o hábito de iniciarem o ensino de programação por meio de uma disciplina introdutória, na qual são abordados conceitos fundamentais, tais como: tipos de dados, constantes, variáveis, sintaxes de entrada e saída, operadores aritméticos e as estruturas sequenciais. Assim, esta disciplina introdutória tem o objetivo de guiar o aluno a dar os passos iniciais para que ocorra o ensino e aprendizagem da Lógica de Programação.

Vislumbrando uma possibilidade de minimizar as dificuldades no ensino da lógica de programação, Ventura (2018) aponta para a utilização atividades lúdicas, já que essas além de propiciar o aprendizado de iniciantes na disciplina de programação também despertam interesse pelo assunto (Ventura, 2018).

Neste contexto, quando um aluno é estimulado a solucionar um problema, busca por estratégias que permitam utilizar a criatividade a fim de encontrar diferentes maneiras de resolvê-lo, promovendo o pensamento computacional.

Pensamento Computacional

O Pensamento Computacional é apontado por Wing (2016) como a capacidade de reformulação e resolução de problemas do mundo real.

O desenvolvimento do PC exige a criatividade do aluno para que ele seja capaz de utilizar diferentes estruturas lógicas a fim de solucionar problemas (Resnick, 2017). Esse pensamento oferece às pessoas a possibilidade de desenvolverem o pensamento abstrato, pensamento algorítmico, pensamento lógico e pensamento dimensionável (Wing, 2016). Assim, quando o PC é desenvolvido pelos jovens, promove grande influência no futuro desta pessoa, pois capacita-os no desenvolvimento das TDIC utilizadas na sociedade moderna (Ventura, 2018).

Blikstein (2008) salienta que o pensamento computacional é uma ferramenta de poder cognitivo e operacional humano, que possibilita aumentar a produtividade, inventividade e criatividade.

Já para Furber (2012, p. 29):

Pensamento computacional é o processo de reconhecer aspectos da computação no mundo que nos rodeia, e aplicar ferramentas e técnicas da Ciência da Computação para entender e raciocinar sobre sistemas e processos tanto naturais quanto artificiais.

Para a *International Society for Technology in Education* em colaboração com a *Computer Science Teachers Association* (Csta/Iste, 2011), o pensamento computacional (PC) é um processo de solução de problemas que inclui, mas não está limitado às

seguintes características: a) Formulação de problemas de uma forma que nos permita usar um computador e outras ferramentas para ajudar a resolvê-los; b) Organização e análise lógica dos dados; c) Representação de dados através de abstrações como modelos e simulações; d) Automatização de soluções por meio de pensamento algorítmico (uma série de etapas ordenadas); e) Identificação, análise e implementação de possíveis soluções com o objetivo de alcançar a combinação mais eficiente e eficaz de etapas e recursos; f) Generalização e transferência deste processo de resolução de problemas para uma ampla variedade de problemas.

Assim, o PC pode ser definido como uma abordagem que tem o intuito de resolver problemas utilizando processos de análises de dados, criação de modelos e construção de algoritmos que possibilitem auxiliar e facilitar o trabalho das pessoas.

Raabe *et al.* (2018, p. 57) afirmam que o PC é constituído por 4 pilares:

- I) decomposição: identificação e redução de um problema em partes menores para facilitar a compreensão;
- II) reconhecimento de padrões: verificar se uma solução que atende a uma parte pode ser repetida em outras situações dentro de um problema macro através de pontos que se repetem;
- III) abstração: manter o foco no que é importante e excluir o que pode ser descartado e;
- IV) algoritmo: conjunto de passos que compõe uma solução replicável.

O Pensamento Computacional envolve identificar um problema complexo e quebrá-lo em pedaços menores e mais fáceis de gerenciar (DECOMPOSIÇÃO). Cada um desses problemas menores pode ser analisado individualmente com maior profundidade, identificando problemas parecidos que já foram solucionados anteriormente (RECONHECIMENTO DE PADRÓES), focando apenas nos detalhes que são importantes, enquanto informações irrelevantes são ignoradas (ABSTRAÇÃO). Por último, passos ou regras simples podem ser criados para resolver cada um dos subproblemas encontrados (ALGORITMOS) (Brackmann, 2017).

Nessa perspectiva o *Scratch* é uma das ferramentas utilizada no Brasil que possibilita trabalhar o PC, propiciando com que o aluno adquira habilidades como a lógica de programação e tomada de decisão. Deste modo, quando o aluno domina as habilidades do PC, apresenta facilidades na compreensão e interação de diferentes assuntos, encontrando soluções para lidar com a complexidade dos problemas do cotidiano (Pereira; Franco, 2018).

O *Scratch* é uma ferramenta educacional que utiliza linguagem de programação visual, desenvolvida pelo grupo Lifelong Kindergarten do Instituto de Tecnologia de Massachusetts MIT Media Lab. A ferramenta é um *software* gratuito que foi construído em 2007 sob o slogan “imaginar, programar, compartilhar”, cujo objetivo é motivar pessoas de qualquer faixa etária a se tornarem produtoras do conhecimento com o auxílio do computador (Rezende; Bispo, 2018).

A ferramenta apresenta algumas potencialidades como liberdade de criação, comunicação, criatividade e compartilhamento, aprendizagens de conceitos escolares, manipulação de mídias, troca de projetos via internet, permitindo que os usuários compartilhem projetos entre si, reutilizar e adaptar projetos já existentes, e integração de objetos do mundo físico (*Scratch*, 2017).

Dessa forma, ela se mostra promissora no ensino de programação devido a sua ludicidade, podendo ser utilizada para introduzir a lógica de programação e desenvolver o pensamento computacional nos anos iniciais dos cursos da computação, auxiliando a aprendizagem do aluno (Rezende; Bispo, 2018).

Metodologia

Para a construção desse trabalho, foram utilizados os procedimentos metodológicos da pesquisa bibliográfica de cunho qualitativo exploratório, a qual proporcionou a fundamentação teórica necessária para a elaboração da atividade e preocupou-se com os aspectos que envolviam a realidade do ensino de programação na Universidade.

Para realizar a análise qualitativa utilizou-se de um questionário semi-estruturado com os estudantes participantes, composto por questões dissertativas que buscavam verificar as percepções sobre o uso da ferramenta *Scratch* no ensino da Lógica de Programação.

Para o desenvolvimento prático da atividade, foram selecionadas as estruturas de lógica de programação, decisão e repetição, apontadas por Pereira *et al.* (2012) como sendo fundamentais na disciplina de programação. Já a análise das atividades seguiu os pilares do Pensamento Computacional, o qual objetiva identificar se o aluno foi capaz de concluir todas as etapas das atividades (Raabe *et al.*, 2018).

Assim, foi associada a ferramenta *Scratch* ao ensino de programação por meio de um curso envolvendo conteúdos teóricos, de vídeos e atividades sobre a temática. O conteúdo programático do curso foi definido junto aos professores da Área da Computação

de modo a contribuir com o conhecimento teórico e prático dos alunos sobre o ensino de programação por meio do *Scratch*.

O curso obteve carga horária total de vinte horas, que foram divididas em dez encontros presenciais de duas horas, em três módulos: I – Lógica de Programação; II – Estrutura de Decisão e; III) Estrutura de Repetição. A atividade apresentada aqui pertence ao Módulo II e teve o objetivo de explorar exercícios de programação que envolvessem estruturas de condição, sendo necessário utilizar a lógica de programação explorada no Módulo I.

Quadro 1 - Atividade avaliativa do Módulo II

Módulo	Ativ.	Descrição
II	2	<p>Desenvolva um algoritmo que calcule a média de três notas as quais serão inseridas pelo usuário: Nota 1; Nota 2; Nota 3.</p> <p>Calcule, analise a média total do usuário e aponte se ele está aprovado ou de exame, considerando que a média é 7.</p> <p>Caso o aluno fique de exame, analise se o mesmo tem o direito de efetuar uma nova prova, caso seja possível, o usuário insere a nota do exame e o programa executa um novo cálculo a fim de saber se ele será aprovado ou não. Considerando que a média mínima do exame é 4 e a mínima para passar no exame é 5.</p> <p>Ao final, o algoritmo exibe ao usuário se ele está aprovado, aprovado com exame ou reprovado.</p>

Fonte: Autores

A atividade apresentada no Quadro 1 envolve uma situação problemática que exige do aluno habilidades de análise, interpretação, cálculo, lógica matemática e generalização da informação, além da necessidade de utilizar estruturas lógicas de programação por meio do *Scratch* e o Pensamento Computacional.

A seguir, a representação por meio da Figura 1 uma alternativa para a resolução da atividade 2.

Figura 1 – Resolução atividade 2.



Fonte: Autores

Após conhecerem e utilizarem a ferramenta *Scratch* para desenvolver atividades

foram convidados a responder: Qual seu ponto de vista em relação ao *Scratch*?

Os alunos foram codificados com a vogal A e um número ordinal: A1, A2, A3 ... A6. Os excertos foram descritos na íntegra, uma vez que não houve a correção gramatical.

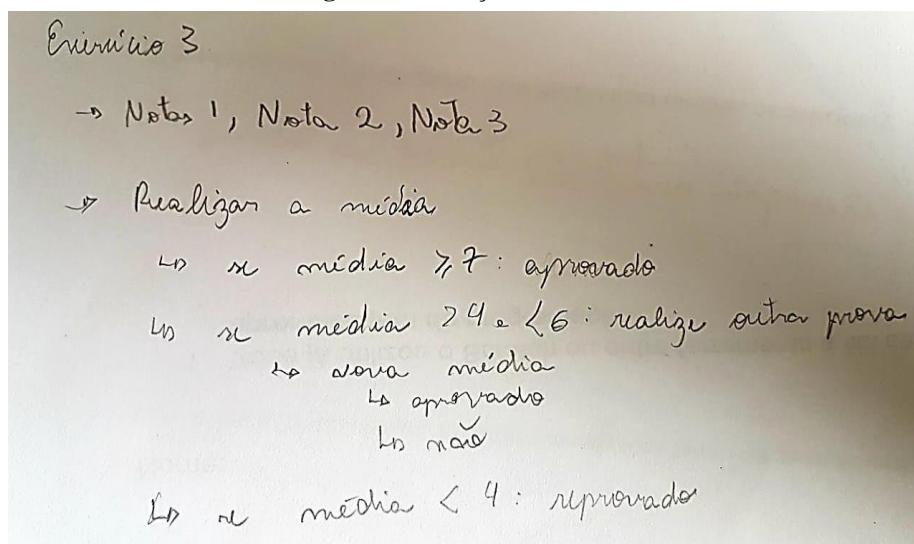
Resultados e Discussão

Neste capítulo, apresenta-se os dados coletados dos alunos durante o curso, bem como as análises dos exercícios. Para a coleta desses dados utilizou-se o Scratch e o software Atube®, o qual foi responsável por gravar a resolução e o desenvolvimento dos exercícios do início ao fim, auxiliando e enriquecendo a análise das atividades.

Nessa categoria foram analisados os dados referentes às estruturas lógicas de condição que foram coletadas no Módulo II. Tais estruturas se fazem presente na maioria dos algoritmos de programação, sendo apontado por Pereira *et al.* (2012) como uma das estruturas fundamentais para o desenvolvimento de um algoritmo.

Seguindo as etapas do Pensamento Computacional com o objetivo de avaliar as atividades, foram replicados por meio de figuras os rascunhos desenvolvidos pelos alunos A2, A3, A5 e A6. Os alunos A1 e A4 não utilizaram rascunho.

Figura 2 – Esboço A2 e A3

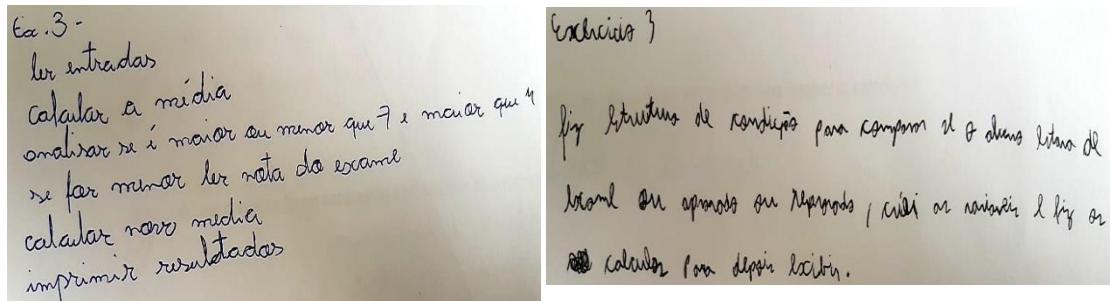


Fonte: Autores

Ao explorar os esboços apresentados, evidenciou-se as duas primeiras etapas do PC – a Decomposição e o Reconhecimento por Padrões. Os alunos A2 e A3 praticamente seguiram o mesmo raciocínio, visto que primeiramente dividiram as variáveis de entrada, ou seja, as notas que o usuário irá inserir no programa, posteriormente fizeram um esquema de critérios utilizando numerais e operadores lógicos a fim de classificar a média do usuário e, por fim, classificaram o aluno como aprovado, aprovado com exame e

reprovado. Nota-se ainda que seguiram uma ordem lógica por padrões a fim de dividir o problema em partes menores. Já A5 e A6 desenvolveram um esquema mais textual, a fim de esboçar a problemática proposta pela atividade.

Figura 3 – Esboço A5 e A6



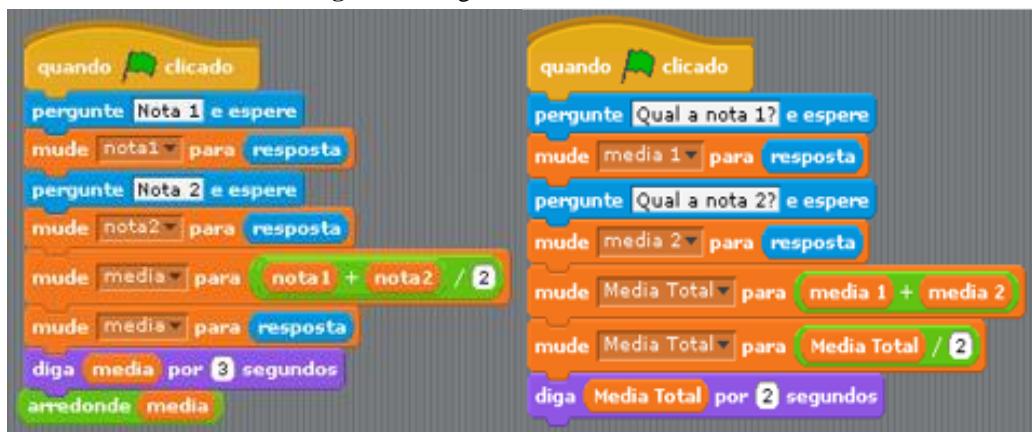
Fonte: Autores

Ainda, estipularam padrões e seguiram uma ordem lógica, dividindo o problema em partes menores, representando a Decomposição e o Reconhecimento por Padrões, assim como afirma (Resnick, 2017).

Para sondar as etapas da Abstração e do Algoritmo foram replicados por meio das figuras abaixo e os códigos desenvolvidos respectivamente por A1, A2, A3, A4, A5 e A6.

A partir dos algoritmos exibidos no Figura 4, percebe-se que A1 e A5 não atingiram os objetivos exigidos pela atividade. Porém, vale apontar que os mesmos conseguiram efetuar uma parte da atividade, calculando a média e exibindo um resultado, importantes para o processo de programar.

Figura 4 – Algoritmo do A1 e do A5



Fonte: Autores

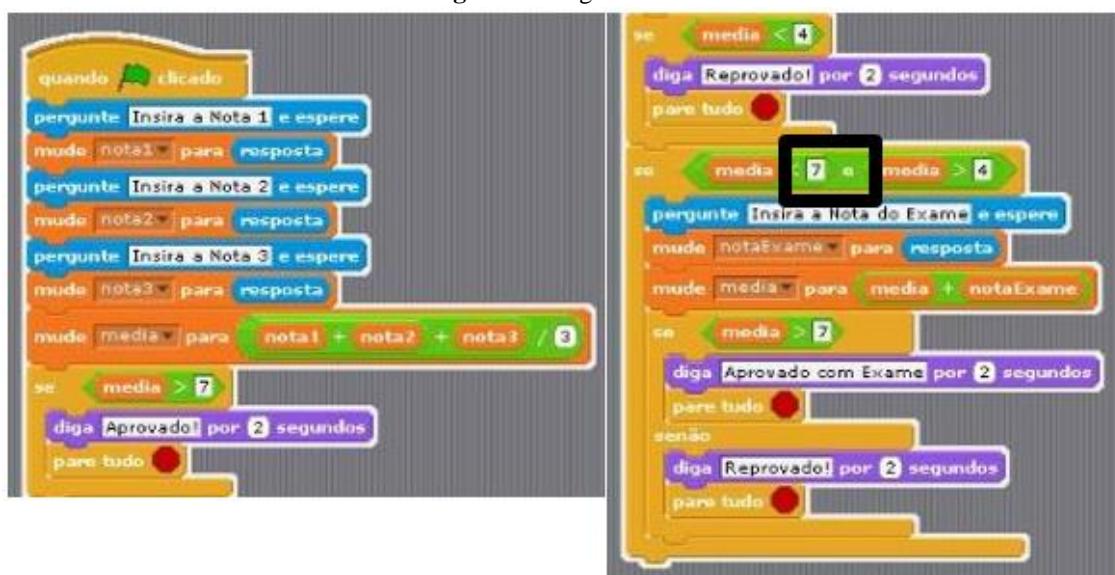
O algoritmo desenvolvido por eles funciona perfeitamente, ficou claro que as etapas do pensamento computacional estão contidas no algoritmo, visto que o mesmo funciona. Porém, o objetivo dessa atividade era avaliar se os alunos conseguiriam utilizar estruturas de condição, as quais eram fundamentais para solucionar o problema proposto. Assim, o algoritmo desenvolvido e apresentado por A1 e A5 é funcional, utiliza os passos

do PC, mas não satisfaz o objetivo proposto pela atividade devido a não utilização de estruturas de condição para tratar a média das notas do usuário.

Nesse contexto, A1 e A5 apresentaram fragilidades em utilizar estruturas de condição. Tais dificuldades, segundo Resnick (2017), podem estar associadas à falta de criatividade, falta de estímulo do Pensamento Computacional, incapacidade de utilizar e estruturar algoritmos por meio da lógica de programação, deficiência no processo de aprendizagem dessas estruturas, entre outras, as quais podem ser minimizadas com a utilização do *Scratch*.

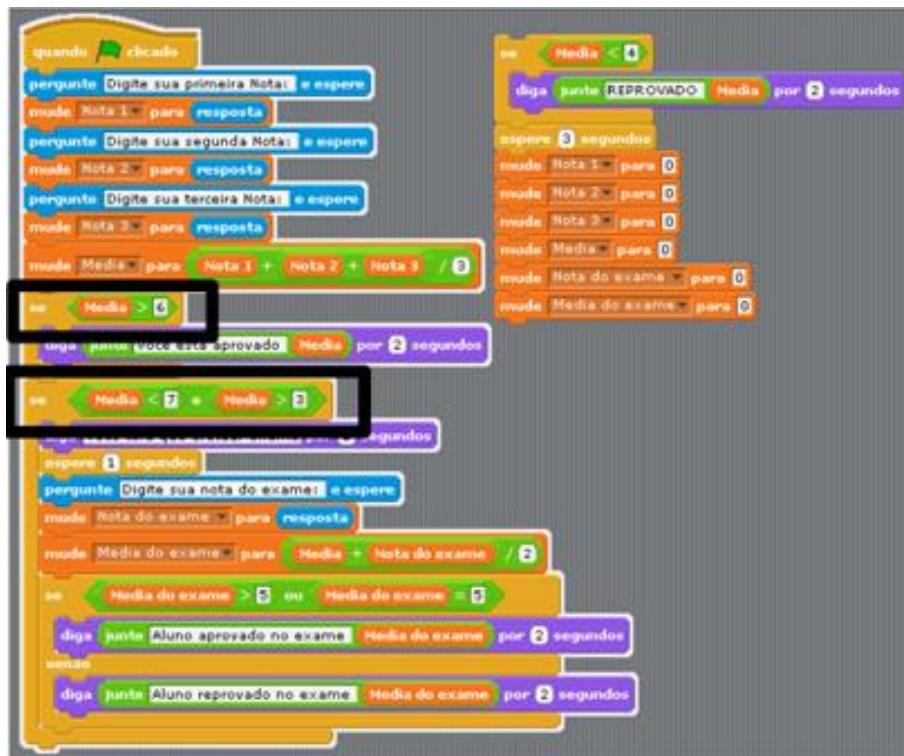
Ao explorar os códigos desenvolvidos por A2 e A4 notou-se uma problemática semelhante as quais foram destacadas por meio de retângulos com bordas pretas, nas respectivas figuras.

Figura 5 – Algoritmo A2



Fonte: Autores

No caso do A2, ao tratar da nota do exame, o mesmo utilizou a seguinte condição (**SE Média > 7**), validando que se a nota referente ao exame inserida pelo usuário estivesse entre 5 e 6.9, apontaria que o usuário está REPROVADO, contudo a nota mínima exigida pela atividade para que o usuário seja APROVADO no exame é 5. Destarte, para solucionar o problema, o A2 poderia ter utilizado as seguintes condições: **SE (Média < 5)**, ou poderia ter utilizado: **SE ((Média = 5) ou (Média > 5))**; deste modo, o algoritmo teria funcionado perfeitamente, atendendo aos requisitos propostos pela atividade 2. Já o A4 cometeu dois erros, os quais estão destacados por meio de retângulos com bordas pretas em destaque, apresentados na Figura 6.

Figura 6 – Algoritmo A4**Fonte:** Autores

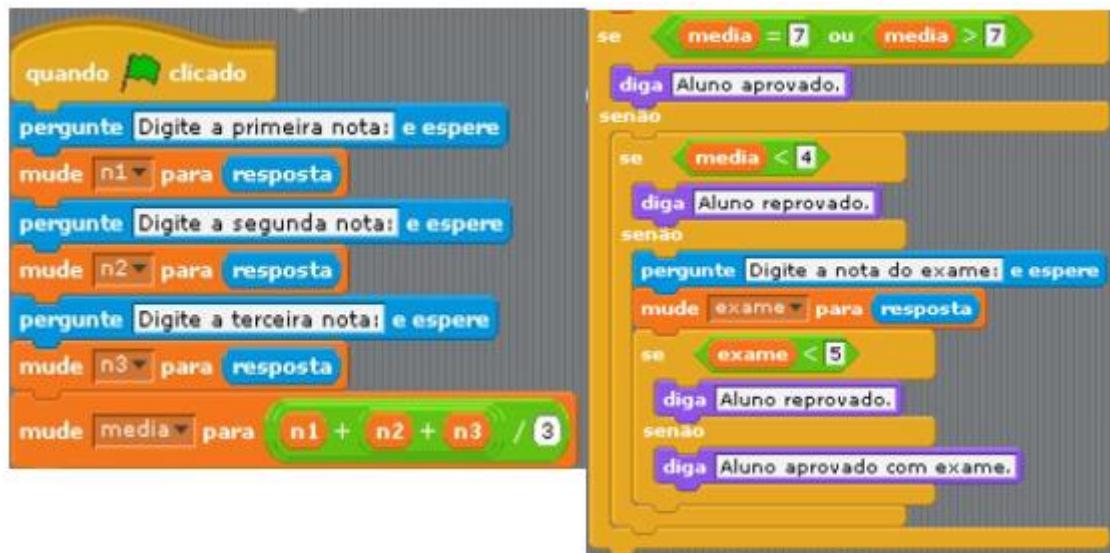
O primeiro, ao categorizar se o usuário pegaria exame ou não, visto que A4 utilizou a seguinte condição para fazer a seleção: **SE (Média > 6)**. Considerando que a média necessária para que o usuário seja APROVADO diretamente é de no mínimo 7, com a condição desenvolvida por A4, o usuário seria aprovado com notas maiores que seis, descumprindo um objetivo determinado pela atividade 2. A estrutura correta a fim de selecionar o aluno seria: **SE ((Média = 7) ou (Média > 7))**, assim o usuário só seria aprovado se obtivesse média maior ou igual a sete.

O segundo erro cometido por A4, está localizado na próxima estrutura de condição, onde o mesmo não soube utilizar a lógica de programação a fim de selecionar o usuário. A condição elaborada por A4 para selecionar se o usuário tinha direito de efetuar o exame ou não necessitava seguir a seguinte condição: Se a média do usuário for maior ou igual a quatro, ele tem o direito de efetuar o exame, mas se a média do usuário for menor que quatro, o mesmo não tem direito de efetuar o exame, sendo reprovado direto. A condição elaborada por A4 para fazer essa seleção foi a seguinte: **SE ((Média < 7) e (Média > 3))**, assim sendo, se o usuário obtivesse a média maior que três, poderia efetuar o exame, o

que não era permitido, levando em consideração que a média mínima é 4. A forma correta de utilizar tal condição a fim de seguir os objetivos propostos pela atividade seria: **SE** (**Média < 7**) e (**(Média = 4)** ou (**Média > 4**)).

Os algoritmos desenvolvidos por A3 e A6, foram os únicos que atingiram todos os objetivos propostos pela Atividade 2. Ambos alunos souberam lidar com os pilares do PC, efetuando cada uma delas de maneira satisfatória, garantindo a solução da atividade, conforme Figura 7.

Figura 7 – Algoritmo A3 e A6



Fonte: Autores

De acordo com Wing (2016), o aluno que domina a prática do PC consegue desenvolver o pensamento abstrato, pensamento algorítmico, pensamento lógico e o pensamento dimensionável. Assim, ele consegue decompor, abstrair, desenvolver padrões e elaborar algoritmos por meio de diferentes estruturas lógicas que possibilitam solucionar problemas (Raabe et al., 2018).

De maneira geral, todos os alunos utilizaram as etapas do PC, até mesmo A1 e A5 que fizeram apenas uma parte da atividade. O problema foi que os mesmos não apresentaram conhecimento suficiente a fim de utilizar cada etapa de maneira lógica e eficaz, devido a falta do desenvolvimento e estímulo do PC, essa habilidade abrange inúmeros processos e etapas a serem seguidos, dentre elas, tomada de decisões, organização de informações até o desenvolvimento do algoritmo (Wing, 2016).

Levando em consideração as etapas do PC, nota-se que A1 e A5, apresenta dificuldades em todas as etapas, visto que apenas iniciaram o algoritmo. A2 e A4 apresentaram falhas ao desenvolverem padrões e ao efetuarem a abstração dos dados. Já A3 e A6 conseguiram utilizar todas as etapas satisfatoriamente.

Nessa perspectiva, Valente (2016) relata que o *Scratch* é uma ferramenta que auxilia o processo de ensino de programação, visto que é uma ferramenta que aborda conceitos fundamentais para a aprendizagem de programação inicial, podendo ser utilizada como uma forma alternativa para introduzir o ensino de programação em sala de aula.

Considerações finais

Em vista do objetivo deste artigo, discutiu-se as principais características do pensamento computacional à formação de licenciando em Computação, ao longo de uma atividade que visava a construção de um algoritmo com a ferramenta *Scratch* para o cálculo de média. Nesse processo verificou-se o algoritmo, a decomposição, a abstração e o reconhecimento de padrões. Ao analisar essas características do pensamento computacional, entende-se que elas se estruturam como habilidades importantes à formação de conceitos específicos de computação dos alunos, pois incentivam o processo de reflexão-argumentação, o desenvolvimento do pensamento lógico-racional e favorece a compreensão de conceitos como uma rede de significados.

Observou-se a Decomposição e o Reconhecimento de Padrões, constituintes do pensamento computacional, quando os alunos dividiram as variáveis de entrada, organizaram um esquema de critérios, dentre eles o textual. Utilizaram numerais, operadores lógicos e construção de condicionais.

Abstração e algoritmo se evidenciam nas variáveis que compõem a execução do programa em construção, o que gerou a busca de novas estratégias para a solução, aprofundando a compreensão lógico-analítica. A construção do algoritmo mobilizou-se em um processo de abstração que comprehende mecanismos sequenciais e lógicos. Ao resolverem o problema elaborando um algoritmo de solução com códigos de programação desenvolveram a lógica, a curiosidade e, por conseguinte, a concatenação de ideias. Esses derivados procedurais e habilidades, tiveram como sustentação, a interface oferecida pelo *Scratch*, com sua linguagem de programação em blocos e esquemas lógicos pré-definidos. O seu uso alinhado ao pensamento computacional favoreceu a resolução do problema e a tomada de decisões.

Notou-se que a maioria não apresentou dificuldade ao elaborar um esquema mental para solucionar a atividade, visto que foram realizados por meio de desenhos e rascunhos, esquematizando uma proposta de solução para a atividade. Contudo, pode-se observar que ao transcrever esse esquema para um algoritmo que exige pensar

computacionalmente, os participantes demonstraram dificuldades.

Referências

- ALBERTIN, A. L.; ALBERTIN, R. M. de M. Benefícios do uso de tecnologia de informação para o desempenho empresarial. **Revista de Administração Pública- RAP**, Rio de Janeiro, v. 42, n. 2, p. 275-302, 2008.
- BITTENCOURT, R. A. *et al.* Aprendizagem de Programação Através de Ambientes Lúdicos em um Curso de Engenharia de Computação: Uma Primeira Incursão. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 23, 2013, Maceió. **Anais**[...]. Maceió: SBC, 2013, v.1, p. 749–758.
- BLIKSTEIN, P. **O pensamento computacional e a reinvenção do computador na educação.** [S. l.] 2008. Disponível em: http://www.blikstein.com/paulo/documents/online/ol_pensamento_computacional.html. Acesso em: 28 jan. 2018.
- BRACKMANN, C. P. **Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica.** 2017. 226f. Tese (Doutorado em Informática da Educação) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017.
- FORBELLONE, A. L. V.; EBERSPÄCHER, H. F. **Lógica de programação:** a construção de algoritmos e estruturas de dados. 3. ed. São Paulo: Pearson, 2005, 232 p.
- FURBER, S. **Shut down or restart? The way forward for computing in UK schools.** Londres: The Royal Society, 2012.
- ISTE (International Society for Techonology in Education). **Operational definition od computational thinking for K-12 education.** 2011. Disponível em: www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf. Acesso em: 25 Mai 2020.
- MESTRE, P. *et al.* Pensamento Computacional: Um estudo empírico sobre as questões de matemática do PISA. In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 4, 2015, Maceió. **Anais** [...]. Maceió: SBC, 2015. p. 1281- 1289.
- OECD. **PISA (Programme for International Student Assessment).** 2018. Disponível em: <http://www.oecd.org/pisa/>. Acesso em: 20 nov. 2018.
- PEREIRA, A. C.; FRANCO, M. E. Desenvolvendo o pensamento computacional no ensino fundamental com Arduino e *Scratch*. In: ENCONTRO NACIONAL DE COMPUTAÇÃO DOS INSTITUTOS FEDERAIS, 5, 2018, Natal. **Anais** [...]. Natal: SBC, 2018, p. 1-4.
- PEREIRA, P. S. *et al.* Análise do SCRATCH como ferramenta de auxílio ao ensino de programação de computadores. In: CONGRESSO BRASILEIRO DE EDUCAÇÃO EM ENGENHARIA, 40, 2012, Belém. **Anais** [...]. Belém: ABENGE, 2012. p.1-9.

RAABE, A.; BRACKMANN, C.; CAMPOS, F. **Curriculum de Referência em Tecnologia e Computação:** da Educação Infantil ao Ensino Fundamental. São Paulo: CIEB, 2018. E-book. Disponível em: http://curriculo.cieb.net.br/assets/docs/Curriculo_de_Referencia_em_Tecnologia_e_Computacao.pdf. Acesso em: 20 Nov 2018.

RESNICK, M. Fulfilling Papert's Dream: "Computational Fluency for All". In: Technical Symposium on Computer Science Education, 5, 2017, Seattle. **Anais** [...]. Seattle: SIGCSE, 2017.

REZENDE, C. M. C.; BISPO, E. L. Comparação entre o Uso de Pseudocódigo e a Programação Visual no Ensino de Programação: Uma avaliação a partir da ferramenta *Scratch*. In: Conferência Ibérica de Sistemas e Tecnologias de informação, 13, 2018, Caceres. **Anais** [...]. Caceres: CISTI, 2018, p. 1-5.

SCAICO, P. D. et al. Programação no ensino médio: uma abordagem de ensino orientado ao design com *Scratch*. In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 23, 2012, Rio de Janeiro. **Anais** [...]. Rio de Janeiro: SBC, 2012, p. 1-10.

SCRATCH. Imagine, program, share. 2017. Disponível em: <http://Scratch.mit.edu>. Acesso em: 11 Jun 2019.

VALENTE, J. A. Integração do Pensamento Computacional no currículo da Educação Básica: diferentes estratégias usadas e questões de formação de professores e avaliação do aluno. **e-Curriculum**, São Paulo, v. 14, n. 3, p. 864-897, 2016.

VENTURA, L. M. **A lógica de programação e os jogos digitais:** uma experiência com a ferramenta *SCRATCH*. 2018. 109f. Dissertação (Mestrado em Metodologias para o Ensino de Linguagens e suas Tecnologias) – Universidade do Norte do Paraná, Londrina, 2018.

WING, J. Pensamento Computacional: Um conjunto de atitudes e habilidades que todos, não só cientistas da computação, ficaram ansiosos para aprender e usar. **Revista Brasileira de Ensino de Ciência e Tecnologia**, Ponta Grossa, v. 9, n. 2 , p. 1-10, 2016.

ZAHARIJA, G.; MLADENOVIC, S.; BOLJAT, I. Introducing basic Programming Concepts to Elementary School Children. **Procedia - Social and Behavioral Sciences**, Turkey, v. 106, p. 1576-1584, 2013.

Submissão: 26/04/2024. **Aprovação:** 03/09/2024. **Publicação:** 25/04/2025.