

Investigando a mediação teórica na aprendizagem de conceitos de programação com estudantes do Ensino Fundamental

Jean Miguel¹, Rafaela Melo¹, Fernanda Pires¹, Marcela Pessoa¹

¹Escola Superior de Tecnologia – Universidade do Estado do Amazonas (EST/UEA)
ThinkTEd Lab - Pesquisa, Desenvolvimento e Inovação em Tecnologias Emergentes

{jmcb.lic22, fpires, mspessoa}@uea.edu.br

rmelo@icomp.ufam.edu.br

Abstract. *This study investigates whether theoretical mediation, applied after the use of interactive platforms such as the Hour of Code, contributes to helping 5th-grade elementary school students understand and apply programming concepts in different contexts beyond merely following step-by-step instructions. The research involved a study with two groups: one received the theoretical intervention after using the platform, while the other did not. The experiment included a diagnostic assessment with logic questions, a hands-on programming activity, and finally, the creation of flowcharts representing the solution to a real-world problem. The results indicate that although both groups were able to follow instructions and complete the challenges, only the group that received mediation showed significant application of concepts such as loops and functions in new contexts. These findings suggest that theoretical mediation enhances conceptual understanding and the abstraction of programming logic.*

Resumo. *Este estudo investiga se a mediação teórica, aplicada após o uso de plataformas interativas como a Hora do Código, contribui para que estudantes do 5º ano do Ensino Fundamental compreendam e apliquem conceitos de programação em contextos diferentes, que não envolvam apenas a execução de instruções passo-a-passo. A pesquisa se trata de um estudo com dois grupos: um recebeu a intervenção teórica após o uso da plataforma, enquanto o outro não. O experimento incluiu uma avaliação diagnóstica envolvendo questões de lógica, uma atividade prática de programação e, por fim, a elaboração de fluxogramas representando a resolução de um problema cotidiano. Os resultados apontam que, embora ambos os grupos tenham conseguido seguir instruções e concluir os desafios, apenas o grupo com mediação demonstrou aplicação significativa de conceitos como repetição e função em novos contextos. Esses achados indicam que a mediação teórica potencializa a compreensão conceitual e a abstração da lógica de programação.*

1. Introdução

A Base Nacional Comum Curricular (BNCC), em seu novo complemento, tornou obrigatórios conteúdos de computação na Educação Básica [MEC 2018], onde os estudantes, desde os anos iniciais, deverão ter contato com esses assuntos, estruturados em três eixos: (i) Cultura Digital, que visa ao uso ético e responsável das tecnologias no cotidiano,

(ii) Mundo Digital, que tem como objetivo a compreensão do funcionamento da tecnologia e sua aplicação no cotidiano, e (iii) Pensamento Computacional (PC), que envolve o desenvolvimento de habilidades cognitivas que favorecem a resolução de problemas [Wing 2006].

No entanto, aprender programação não é uma tarefa trivial, pois conceitos como algoritmos, condicionais e repetição são considerados complexos de serem entendidos até por estudantes de nível superior, tendo em vista que exigem habilidades como alto nível de abstração, raciocínio lógico e construção de modelos mentais capazes de prever o resultado do algoritmo [Fincher and Robins 2019], trabalhadas nos pilares do Pensamento Computacional. Portanto, faz-se necessária a inserção de diferentes estratégias pedagógicas para tornar o aprendizado desses conteúdos mais simples para estudantes da Educação Básica [Scheffel and Motta 2022].

Diante desse contexto, pesquisadores têm buscado estratégias didáticas que tornem o aprendizado de computação mais lúdico e envolvente para crianças, especialmente por meio de ferramentas digitais com abordagens interativas [Narciso et al. 2024, Martins 2016]. Plataformas como o Hora do Código, Blockly Games, LightBot e Run Marco! têm sido utilizadas com esse propósito por adotarem programação em blocos, desafios progressivos e elementos de gamificação que estimulam a prática de programar, raciocínio lógico e a resolução de problemas [da Silva et al. 2019]. No entanto, embora tais ferramentas promovam engajamento e apresentem os conteúdos de forma dinâmica, a simples execução de tarefas não garante a compreensão dos conceitos envolvidos. Um estudante pode resolver um desafio apenas seguindo instruções visuais, sem necessariamente entender a lógica por trás da solução construída [Quequi 2021].

Aprender programação vai além de executar comandos, pois exige a criação de representações internas dos conceitos, a abstração dos conteúdos utilizados e a capacidade de aplicá-las em diferentes contextos [Albuquerque et al. 2016]. Nesse sentido, a mediação teórica por parte de um facilitador torna-se fundamental para transformar a prática em aprendizagem significativa, pois relaciona as atividades realizadas com os conceitos teóricos que explicam como os algoritmos funcionam, tornando o conteúdo mais claro e auxiliando na compreensão dos estudantes [da Silva Pontes and Victor 2022].

Diante disso, este estudo de caso investiga se a mediação teórica por parte de um facilitador, realizada após o uso de plataformas como o Hora do Código¹, contribui para que estudantes do 5º ano do Ensino Fundamental compreendam e sejam capazes de aplicar conceitos de programação em contextos novos, indo além da simples execução de instruções. A pesquisa foi conduzida com duas turmas de 5º ano em uma escola particular do Estado [Omitido para revisão], divididas em grupo de controle e experimental, e composta por cinco etapas: i) aplicação de teste de lógica; ii) realização de teste prático de programação; e iii) intervenção pedagógica; iv) atividade desplugada “Ida ao mercado”; e v) análise dos resultados.

O restante do artigo está organizado da seguinte forma: a Seção 2 apresenta a Fundamentação Teórica e os Trabalhos Correlatos; a Seção 3 descreve a Metodologia; a Seção 4 apresenta a execução do estudo; a Seção 5 apresenta os Resultados; a Seção 6, as discussões; e, por fim, na Seção 7 encontram-se as considerações finais.

¹<https://hourofcode.com/pt/pt>

2. Referencial Teórico e Trabalhos Correlatos

O PC é um dos eixos dentro da BNCC computação, que tem como foco o desenvolvimento de habilidades relacionadas à resolução de problemas, sendo essencial para a formação dos estudantes no contexto digital. Embora não exista uma definição única e consensual do termo, o PC é amplamente entendido como a habilidade de organizar informações, estruturar soluções e expressá-las de forma que possam ser processadas por agentes computacionais [Kalelioglu et al. 2016, Wing 2011]. Habilidades como abstração, decomposição de problemas, reconhecimento de padrões e construção de algoritmos são características fundamentais do PC [ISTE and CSTA 2011]. No contexto escolar, essas habilidades podem ser trabalhadas por meio de estratégias como a programação em blocos, jogos digitais ou atividades desplugadas, que tornam o processo de aprendizagem mais lúdicos para os aprendizes [Moreno-León et al. 2018, Chiazese et al. 2019].

No entanto, apenas o uso de plataformas de programação em blocos não garante que os estudantes compreendam os conceitos envolvidos. Segundo Vigotsky et al. [1987], o aprendizado se torna mais efetivo quando ocorre dentro da Zona de Desenvolvimento Proximal (ZDP), ou seja, quando há a mediação de alguém com mais conhecimento, podendo ser um professor, tutor ou colega que atua como facilitador do processo de aprendizagem. Essa abordagem é fundamental para que o estudante ultrapasse o estágio de seguir instruções e comece a refletir sobre as escolhas de estratégias que fazem durante a resolução de problemas. No contexto de aprendizagem de programação, isso significa que o professor precisa atuar como o intermediário entre a experiência prática oferecida pelas plataformas e o conhecimento conceitual necessário para que o aprendiz compreenda o que está fazendo e consiga aplicar os mesmos princípios em novos desafios [da Silva Pinto and Mattos 2019]. Nesse sentido, a mediação teórica feita por professores capacitados em conteúdos de computação torna-se essencial para que habilidades relacionadas ao Pensamento Computacional sejam trabalhadas por estudantes da Educação Básica [Junior and Rivera 2024].

Nesse sentido, pesquisadores buscam investigar como apresentar conteúdos de computação para estudantes do ensino fundamental de forma lúdica, como no estudo de Sáez-López et al. [2021], onde foi realizada uma intervenção quase-experimental com 107 estudantes do 5º ano, que participaram de uma unidade didática de Ciências baseada no uso de robótica e programação em blocos. A mediação docente ocorreu por meio do planejamento e condução das atividades em sala de aula, utilizando ferramentas como robôs educacionais e ambientes de programação digitais adaptados ao nível dos estudantes. Os resultados apontaram que a mediação dos facilitadores contribuiu para o aprendizado dos conceitos computacionais, como condicionais, e favoreceu o envolvimento dos estudantes nas atividades.

Em Motta [2020], houve um estudo exploratório envolvendo o Pensamento Computacional (PC), por meio de uma combinação entre atividade desplugada e atividade digital com o jogo LightBot. O objetivo era promover a resolução de problemas de forma algorítmica em turmas da educação básica. A proposta envolveu a criação de algoritmos em grupo usando a própria movimentação pela sala para simular comandos de movimentação, seguidos por desafios no ambiente digital, com foco em conceitos como sequência e repetição. Embora a atividade tenha sido aplicada a um público restrito, os resultados indicaram alto envolvimento e potencial de generalização da abordagem, so-

bretudo quando associada à mediação docente e à contextualização pedagógica.

Semelhante aos trabalhos anteriores, apesar de não ser o objetivo principal da pesquisa, este trabalho também aborda o desenvolvimento do pensamento computacional no Ensino Fundamental por meio de atividades práticas baseadas em desafios interativos e programação em blocos. No entanto, diferencia-se ao se tratar de um estudo de caso que incluiu a aplicação de uma avaliação diagnóstica com questões envolvendo raciocínio lógico, aplicação de exercícios práticos de programação com o Hora do Código e a aplicação de uma intervenção pedagógica teórica estruturada com um dos grupos do estudo, visando analisar se os estudantes conseguem aplicar conceitos de computação em outros contextos sem precisar seguir instruções explícitas.

3. Metodologia

Este é um estudo de caráter quali-quantitativo, onde a abordagem utilizada foi o estudo de caso [Creswell and Creswell 2017]. O contexto da pesquisa é uma escola particular do Estado do Amazonas, que possui em sua grade curricular do Ensino Fundamental uma disciplina de “Robótica Educacional”. Na disciplina de Robótica Educacional da escola, observam-se práticas alinhadas a algumas habilidades previstas na BNCC de Computação, como nos eixos de Cultura Digital e Mundo Digital, trabalhadas de forma transversal com outras disciplinas, como Geografia e Física. No entanto, embora envolva aspectos técnicos e práticos da robótica, a disciplina foca diretamente os conceitos centrais do eixo de Pensamento Computacional, tais como decomposição de problemas, abstração, reconhecimento de padrões e elaboração de algoritmos, que são relevantes para a consolidação da área de Computação na Educação Básica.

A pesquisa foi conduzida por um estudante de Licenciatura em Computação da Universidade do Estado do Amazonas, com a supervisão da professora de “Robótica Educacional”. Participaram da pesquisa 46 estudantes de duas turmas do 5º ano do Ensino Fundamental, sendo 22 do 5º A e 24 do 5º B. O estudo de caso foi dividido em quatro etapas: i) avaliação diagnóstica, por meio da aplicação de um teste de lógica (Bebras²); ii) atividade prática de programação (Hora do Código); iii) intervenção pedagógica; iv) atividade desplugada “Ida ao mercado”; e, por fim, v) análise dos dados coletados nas etapas anteriores, como pode ser visualizado na Figura 1.

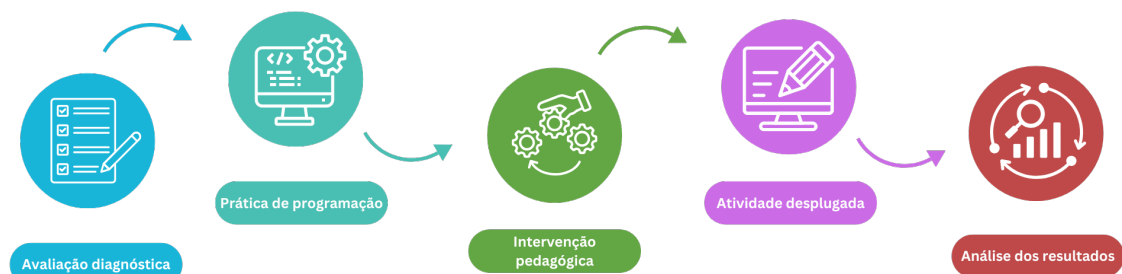


Figura 1. Etapas da metodologia.

As duas primeiras etapas foram realizadas em ambas as turmas de 5º ano, em dias separados por conta dos horários disponíveis e do tamanho das salas em que ocorreram

²<https://www.bebasbrasil.com.br/>

as aplicações. Para a intervenção pedagógica, uma das turmas foi selecionada para ser o grupo experimental, ou seja, que receberia a intervenção, enquanto a outra foi o grupo de controle. A seleção das turmas para participar de cada grupo foi aleatória.

4. Execução do Estudo

Esta seção apresenta a descrição do estudo, de acordo com as etapas da metodologia.

4.1. Avaliação diagnóstica

A avaliação diagnóstica foi realizada com o objetivo de coletar dados sobre o raciocínio lógico dos estudantes antes da realização dos outros testes. Para isso, foi utilizado o desafio internacional Bebras. Os estudantes receberam gabaritos personalizados para anotarem as respostas das questões. A aplicação ocorreu na sala de Robótica da escola, um ambiente com projetor, onde as questões foram espelhadas em *slides* para facilitar a leitura das questões. A atividade foi realizada em tempo médio de 50 minutos. Os estudantes receberam orientações para resolver as questões individualmente, sem auxílio externo, e visualizaram as questões na tela durante todo o tempo da aplicação. Ao final, todos os gabaritos foram recolhidos para ocorrer a análise e tabulação dos dados.

4.2. Prática de programação

A segunda etapa do estudo consistiu na realização de uma atividade prática de programação utilizando a plataforma digital Hora do Código, baseada em desafios de programação em blocos com diferentes temas, nesse caso foram utilizados os desafios inspirados no jogo *Minecraft*³. A aplicação foi realizada na sala de informática da escola, onde havia computadores conectados à internet, permitindo que cada estudante acessasse a plataforma nos dispositivos disponíveis. Durante 50 minutos, os estudantes foram incentivados a explorar livremente os desafios propostos pela plataforma, avançando conforme seu próprio ritmo. Devido à falta de computadores para todos os estudantes no laboratório para resolver os desafios de programação, eles se juntaram em duplas, utilizando a função de *pair programming* presente no Hora do Código. A ementa desses testes trabalha com uma sequência didática gradual, onde os exercícios aumentam o grau de dificuldade a cada nível e novos conteúdos surgem. O responsável pela explicação acompanhou os estudantes para tirar quaisquer dúvidas que eles tivessem, sem fornecer respostas dos exercícios.

4.3. Intervenção pedagógica

A intervenção pedagógica foi aplicada apenas com o grupo experimental, com o objetivo de fornecer suporte teórico aos conteúdos explorados nos exercícios do Hora do Código. A intervenção consistiu em uma aula de 50 minutos dividida em parte teórica e uma parte prática conduzida pelo licenciando responsável pelo estudo, com apoio da professora da disciplina de “Robótica Educacional”. A aplicação ocorreu em sala de aula. O início da aula foi a parte teórica, onde foram abordados de forma simplificada os conceitos de algoritmo, repetição e função, sendo esses conteúdos utilizados pelos estudantes durante o teste com o Hora do Código, mas agora ocorrendo a explicação dos conteúdos de forma mais simples.

³<https://www.minecraft.net/pt-br>

Para explicar o que são algoritmos, houve a comparação com uma sequência de passos para realizar um objetivo, sendo apresentado como exemplo a rotina dos estudantes para chegar à escola, indo desde acordar, escovar os dentes, se arrumar e ir para a escola, onde cada ação representa um passo no algoritmo para executar a tarefa. A explicação sobre repetição envolveu a comparação com a repetição de ações semelhantes durante o dia-a-dia, como subir uma escada, onde é realizada uma sequência de passos para realizar a ação. Para explicar funções, houve a analogia com uma receita de bolo, que pode ser seguida N vezes para fazer quantos bolos forem necessários, ou seja, a receita está presente no algoritmo e pode ser chamada quando for preciso, remetendo à ideia da função. O grupo controle não recebeu essa mediação teórica, tendo em vista investigar se a interação com a plataforma Hora do Código seria suficiente para o entendimento desses conteúdos.

4.4. Atividade desplugada

Após a intervenção pedagógica, foi passado para os estudantes, tanto da turma experimental, quanto da turma de controle, o exercício “Ida ao Mercado”. O objetivo de aplicar com as duas turmas foi identificar se a mediação teórica influenciaria no desempenho dos estudantes durante a execução da tarefa.

Durante o exercício, os participantes precisaram criar um algoritmo em forma de fluxograma para resolver o problema de fazer compras no mercado. O conceito de fluxograma já era conhecido por conta de uma aula ministrada pela professora das turmas. Então, durante a tarefa, os estudantes precisaram detalhar os passos para comprar os produtos, indo desde acordar até chegar no mercado, pegar o produto e colocar no carrinho. Além disso, utilizaram conceitos de repetição (por exemplo, produtos comprados em maior quantidade) e funções (uma situação que poderia ser realizada várias vezes em produtos diferentes, como uma função para checar a validade do produto).

4.5. Análise dos resultados

A última etapa do estudo consistiu na análise de evidências coletadas nas etapas anteriores, sendo: i) as respostas dos estudantes no teste do Bebras; ii) as respostas dos exercícios do Hora do Código, tanto código em blocos quanto a versão na linguagem *Javascript*; e por fim; iii) as respostas dos estudantes ao exercício “Ida ao Mercado”, sendo os fluxogramas de ambas as turmas. Esses dados foram armazenados em uma planilha eletrônica com a ferramenta *Google Sheets*.

5. Resultados

Nesta seção são apresentadas as análises dos dados coletados durante a aplicação do experimento.

5.1. Avaliação diagnóstica com o Bebras

Apesar das diferenças dos acertos totais de questões por turma, o desempenho foi relativamente semelhante no que diz respeito ao tipo de questão que os estudantes conseguiram resolver com mais facilidade, como pode ser visto na Figura 2. A análise dos dados revelou que, tanto no grupo experimental quanto no grupo controle, as questões com maior número de acertos eram as que os aprendizes precisavam seguir um caminho ou sequência

lógica de ações para alcançar um objetivo, como localizar árvores a partir de posições fixas (Questão 1) ou identificar o trajeto para coletar cenouras em um mapa (Questão 4). Esses dados podem indicar que os participantes conseguem executar instruções organizadas em sequência para atingir um resultado final; entretanto, isso não indica aprendizado de fato. Ademais, quando as questões exigem raciocínio lógico mais abstrato ou a manipulação de variáveis do problema, o número de acertos tende a diminuir, sugerindo maior dificuldade na construção de estratégias autônomas em problemas complexos.

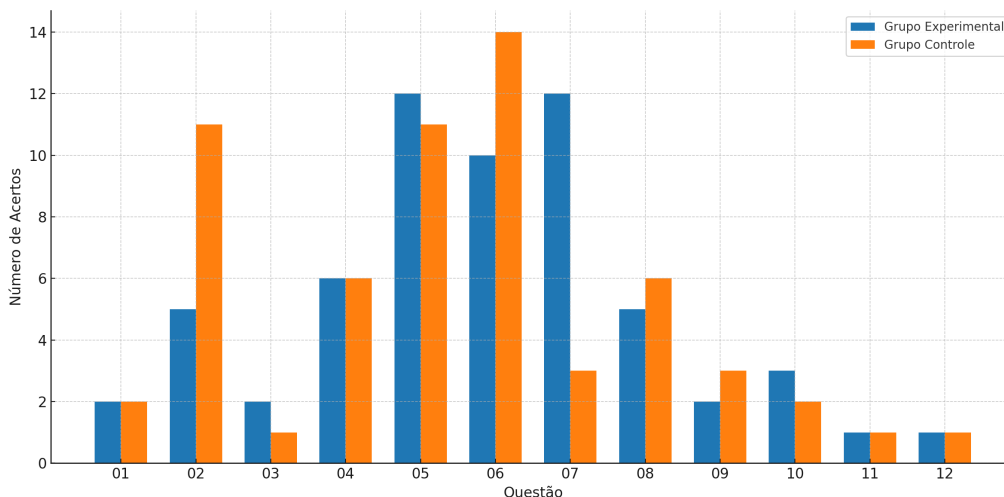


Figura 2. Total de Acertos no Bebras por Turma.

5.2. Desempenho no Hora do Código

Dentre os dados coletados no teste prático, o grupo experimental produziu 80 códigos, dos quais 33,75% utilizaram estruturas de repetição e 17,5% utilizaram funções. A turma de controle, por sua vez, produziu 70 códigos, com 20,0% usando repetição e apenas 5,71% funções, como pode ser visualizado na Tabela 1.

Tabela 1. Análise dos códigos produzidos na plataforma Hora do Código.

Turma	Total de códigos	% com Repetição	% com Função	Códigos extensos
Experimental	80	33,75%	17,5%	29
Controle	70	20,0%	5,71%	33

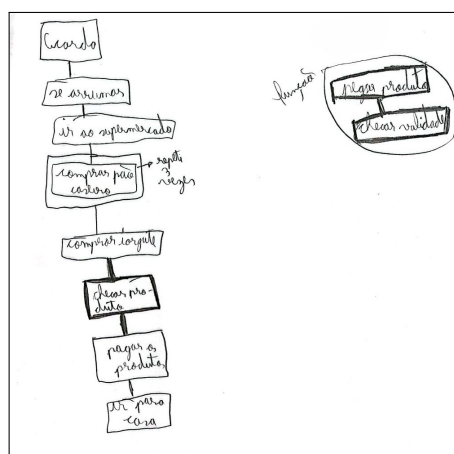
Apesar do uso de estruturas como repetição e função, o progresso geral foi limitado, já que apenas 6,67% dos estudantes de cada turma chegaram até o exercício dez de doze exercícios presentes no Hora do Código. Em ambos os grupos, observou-se um padrão relacionado à influência direta das instruções dos exercícios na escolha das estruturas utilizadas. A sequência didática do Hora do Código introduz conceitos de forma gradual, iniciando com algoritmos simples e avançando para o uso de repetição e função. No exercício três, onde há uma orientação explícita para o uso de repetição, 80% dos estudantes da turma experimental e 70% da turma de controle utilizaram essa estrutura corretamente. No entanto, no exercício cinco, que, embora também permitisse o uso de repetição, não apresentava uma instrução clara nesse sentido, os percentuais caíram para

40% no grupo experimental e 35% no grupo de controle. Ademais, os códigos que não utilizavam estruturas de repetição ou função, chamados neste estudo de códigos extensos, foram muito utilizados pelos estudantes para resolver os problemas, onde de 80 códigos do grupo experimental, 29 códigos eram extensos, e no grupo de controle, houve 33 códigos extensos de 70 no total, indicando que ambos os grupos resolveram os problemas com códigos sem utilizar as estruturas otimizadas.

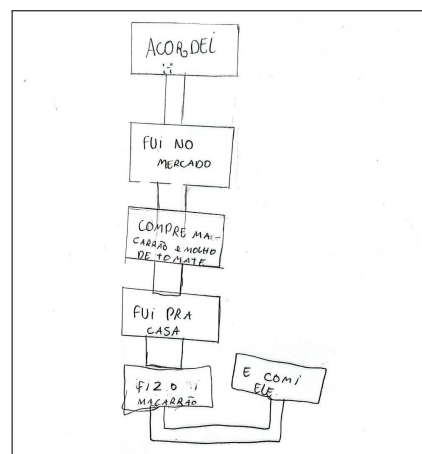
Isso pode indicar que os estudantes tendem a adotar estruturas de programação mais otimizadas apenas quando incentivados diretamente pelo enunciado da atividade, revelando que o entendimento desses conceitos ainda não foi compreendido como parte de um raciocínio autônomo, mas sim dependente de orientações explícitas. Esses dados reforçam a hipótese de que o simples uso de uma ferramenta interativa, sem auxílio pedagógico de um facilitador, pode levar os estudantes a executar instruções corretamente, mas sem necessariamente consolidar os conceitos-chave da programação.

5.3. Intervenção pedagógica e atividade desplugada

Após a atividade prática com o Hora do Código, foi realizada uma intervenção pedagógica apenas com o grupo experimental, que consistiu em uma aula expositiva, com linguagem acessível e exemplos contextualizados, relacionando os desafios resolvidos no Hora do Código com situações do cotidiano. O grupo controle não passou por essa mediação e seguiu diretamente para a atividade desplugada que se tratou da criação de um fluxograma representando a ida ao mercado. Como é possível visualizar na Figura 3(a), o grupo experimental utilizou as estruturas de repetição e função para compor seus fluxogramas, além de adicionar mais detalhes na sequência de passos, enquanto o grupo de controle não utilizou tanto essas estruturas e criou fluxogramas com comandos mais simples, como pode ser visto na Figura 3(b).



(a) Exemplo de fluxograma criado no grupo de experimento.



(b) Exemplo de fluxograma criado no grupo de controle.

Figura 3. Fluxogramas criados em ambos os grupos.

Os resultados dessa etapa revelaram diferenças entre os grupos. No grupo experimental, 70% dos estudantes utilizaram estruturas de repetição e 45% aplicaram funções em seus fluxogramas. No grupo de controle, os percentuais foram visivelmente menores: 43,48% utilizaram repetição e apenas 4,35% utilizaram funções.

Tabela 2. Uso de estruturas de programação nos fluxogramas.

Turma	Total de estudantes	% com Repetição	% com Função
Experimental	20	70,0%	45,0%
Controle	23	43,48%	4,35%

Além das diferenças quantitativas, observou-se um contraste na qualidade das soluções. No grupo experimental, os estudantes demonstraram compreensão mais profunda dos conceitos, aplicando funções para diminuir a quantidade de ações repetidas (como verificar validade ou pegar produtos) e até corrigindo seus próprios fluxogramas ao perceber incoerências, o que indica a construção de modelos mentais mais estruturados. Já no grupo controle, os participantes se limitaram a reproduzir padrões visuais semelhantes aos usados no Hora do Código, com foco em comandos explícitos como “andar para frente” ou “vire à direita”, o que reforça a ideia de que a ausência da mediação teórica pode restringir a transferência conceitual entre contextos.

Ademais, também foram coletados alguns *feedbacks* dos estudantes por turma, onde, no grupo experimental, os estudantes citaram relatos do tipo: *agora eu entendi, na função pode ter mais de uma ação e então eu posso só chamar a função e não preciso mais escrever essas ações*. No grupo experimental, sete estudantes relataram *feedbacks* semelhantes sobre terem entendido esses conceitos e utilizaram de forma correta nos fluxogramas. Já no grupo de controle, apenas dois estudantes deram *feedbacks* sobre terem entendido os conceitos de repetição e função, onde seus *feedbacks* foram, em grande maioria, para tirar dúvidas sobre o fluxo de seus comandos.

Os resultados reforçam a importância da presença de um facilitador no processo de aprendizagem de programação, pois a explicação conceitual de forma simples para estudantes novatos nos conteúdos facilita a abstração e criação de modelos mentais desses conteúdos. Caso os estudantes prossigam apenas seguindo instruções para criarem seus algoritmos, a abstração dos conteúdos pode não acontecer, dificultando o processo de aplicação desses conceitos em outros contextos ou em situações em que não haja instruções. Além disso, tendo em vista a implementação da BNCC Computação nas escolas, torna-se ainda mais importante a presença de um mediador com conhecimento na área, que saberá escolher as melhores sequências didáticas para tratar os conteúdos, seguindo um processo gradual que começa a partir dos anos iniciais da Educação Básica, com o exercício de habilidades essenciais do Pensamento Computacional e vai até os anos finais, evoluindo conforme é previsto na BNCC. Ou seja, é importante que as aplicações relacionadas à computação nas escolas não se limite ao uso de ferramentas tecnológicas.

5.4. Limitações da Aplicação

Este estudo possuiu algumas limitações durante a aplicação das etapas, sendo uma durante a execução da avaliação diagnóstica com o Bebras. A organização do desafio recomenda uma duração de 60 minutos para a realização da prova, entretanto o tempo disponível para as aulas na escola onde ocorreu o experimento era de apenas 50 minutos. Isso pode ter impactado no desempenho de alguns estudantes, principalmente nos itens mais complexos, que exigiam mais tempo de leitura e raciocínio. Ainda assim, buscou-se minimizar esse viés garantindo que os estudantes realizassem o teste em condições iguais.

Outra limitação observada ocorreu durante a etapa de resolução dos desafios do Hora do Código. Ambas as turmas tinham em torno de 20 pessoas, porém a sala de informática da escola possuía apenas 14 computadores disponíveis para uso. Dessa forma, foi necessário que alguns estudantes realizassem as atividades em duplas ou trios, utilizando a funcionalidade de *pair programming* da própria plataforma, que permite o registro de participantes compartilhando a mesma solução. Embora essa abordagem também tenha valor pedagógico, ela pode ter interferido na autonomia e na tomada de decisão individual, uma vez que nem todos os estudantes interagiram diretamente com o ambiente de programação o tempo todo.

6. Discussão

Esse contraste reforça a hipótese de que a prática em programação em blocos, embora útil, não garante sozinha a abstração dos conteúdos, tornando o papel do mediador fundamental para a consolidação do aprendizado. Quando combinadas a explicações teóricas e a exemplos práticos já vivenciados, essas atividades favoreceram a transferência do conhecimento para outros contextos, possibilitando a construção de modelos mentais mais consistentes e o desenvolvimento de maior autonomia. Os resultados sugerem que, apesar de eficazes para introduzir programação de forma lúdica, tais ferramentas têm seu impacto pedagógico ampliado quando acompanhadas de momentos de reflexão que estimulam a compreensão conceitual.

Os dados qualitativos reforçam essa hipótese. Os estudantes que participaram da intervenção mostraram avanços ao utilizar funções compostas, organizar ações repetitivas em estruturas mais eficientes e representar situações cotidianas por meio de sequências lógicas. Já no grupo controle, os participantes criaram mais diagramas com repetições manuais e comandos simples, próximos das instruções fornecidas pela própria plataforma, indicando uma possível falta de autonomia, apenas focando em seguir os exemplos trabalhados no Hora do Código.

7. Considerações Finais

Este trabalho investigou, através de um estudo de caso, se a mediação teórica contribui para que estudantes do 5º ano do Ensino Fundamental compreendam e apliquem conceitos de programação em contextos diferentes daqueles praticados em plataformas como o Hora do Código. A metodologia envolveu a aplicação de uma avaliação diagnóstica, um teste prático com desafios de programação em blocos e uma intervenção pedagógica com apenas um dos grupos. Os resultados indicaram que, embora ambos os grupos conseguissem resolver problemas seguindo instruções explícitas, apenas o grupo que recebeu mediação foi capaz de aplicar estruturas como repetição e função em novos contextos, demonstrando maior compreensão conceitual.

Como trabalhos futuros, pretende-se realizar análises estatísticas dos dados quantitativos coletados para cruzamento de dados entre os resultados do Hora do Código e dos fluxogramas criados durante a atividade.

Referências

Albuquerque, D., Bremgartner, V., Lima, H., and Salgado, N. (2016). Uma experiência do uso do hardware livre arduino no ensino de programação de computadores. In *Workshop de Informática na Escola (WIE)*, pages 51–60. SBC.

- Chiazzese, G., Allegra, M., and Ottaviano, S. (2019). Unplugged coding activities in primary school: A pilot study. In *Proceedings of the 11th International Conference on Computer Supported Education*, volume 1, pages 79–86.
- Creswell, J. W. and Creswell, J. D. (2017). *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications.
- da Silva, A. D., Oliveira, C., Ilnicki, K., Madeira, C., and Campos, A. (2019). Pensamento computacional como auxílio para estimular a noção espacial das crianças do ensino fundamental. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 8, page 387.
- da Silva Pinto, S. C. C. and Mattos, M. S. (2019). A programação de jogos como um instrumento motivador da aprendizagem. *Revista Espaço Pedagógico*, 26(2):370–394.
- da Silva Pontes, P. R. and Victor, V. F. (2022). Robótica educacional: uma abordagem prática no ensino de lógica de programação. *Revista Sítio Novo*, 6(1):57–71.
- Fincher, S. A. and Robins, A. V. (2019). *The Cambridge handbook of computing education research*. Cambridge University Press.
- ISTE and CSTA (2011). Operational definition of computational thinking for k–12 education. <https://csta.acm.org/Curriculum/sub/CurrFiles/472.11CTDefinition.pdf>. Acesso em: jun. 2025.
- Junior, A. d. O. C. and Rivera, J. A. (2024). Bncc computação: O que os acadêmicos de licenciatura precisam saber sobre o pensamento computacional? In *Workshop sobre Educação em Computação (WEI)*, pages 878–891. SBC.
- Kalelioglu, F., Gülbahar, Y., and Kukul, V. (2016). A review of computational thinking studies in education field. *Malaysian Online Journal of Educational Technology*, 4(3):1–12.
- Martins, L. (2016). Ensinando lógica de programação aplicada a robótica para alunos do ensino fundamental. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 27, page 31.
- MEC (2018). *Base Nacional Comum Curricular: Componente de Computação*. Ministério da Educação, Brasília, DF. Acesso em: 9 dez. 2024.
- Moreno-León, J., Robles, G., and Román-González, M. (2018). The effect of block-based programming on computational thinking: An intervention study. In *2018 IEEE Global Engineering Education Conference (EDUCON)*, pages 1364–1372. IEEE.
- Motta, L. C. P. (2020). Metodologias ativas: Aplicando pensamento computacional para o processo de ensino e aprendizagem. *REVISTA ACADÊMICA ALCIDES MAYA*, 2(2).
- Narciso, R., Fernandes, A. B., Burin, G. R. E., de Medeiros, G. A. C. M., da Silva, L. I., Gomes, L. C. M., da Silva, M. V. M., da Cruz, R. C. d. V., Lopes, R. C., and da Silva Bezerra, V. M. (2024). Importância da programação na educação fundamental: preparando alunos para o futuro digital. *Revista Ibero-Americana de Humanidades, Ciências e Educação*, 10(3):268–282.
- Quequi, G. B. (2021). Programação na resolução de problemas envolvendo polígonos regulares por meio do scratch: uma experiência no ensino fundamental.

- Sáez-López, J. M., Buceta Otero, R., and De Lara García-Cervigón, S. (2021). Introducing robotics and block programming in elementary education. *RIED. Revista Iberoamericana de Educación a Distancia*, 24(1):95–113.
- Scheffel, E. J. and Motta, C. L. (2022). Desenvolvimento das competências de computação dispostas na bncc a partir da aprendizagem baseada em problemas com alunos do ensino fundamental. In *Simpósio Brasileiro de Informática na Educação (SBIE)*, pages 85–94. SBC.
- Vigotsky, L. S. et al. (1987). Pensamento e linguagem.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35.
- Wing, J. M. (2011). Research notebook: Computational thinking—what and why? In *The Link*, volume 6, pages 1–5. Carnegie Mellon University.