

# Uma análise das dificuldades no ensino de programação na educação básica e os problemas de interação nas ferramentas computacionais

Thainá Eliza Flexa Souza<sup>1</sup>, João Pedro Vasconcelos de Lima<sup>1</sup>,  
Phelipe Luiz Dias Feio<sup>1</sup>, Marcelle Pereira Mota<sup>1</sup>

<sup>1</sup>Universidade Federal do Pará - UFPA  
Caixa Postal 479 – Belém – PA – Brasil

{thaina.souza, joao.vasconcelos, phelipe.feio}@icen.ufpa.br,  
mpmota@ufpa.br

**Abstract.** *Understanding technology has been widely covered in primary education. Many schools cover concepts such as computational thinking and programming in the classroom. This work aims to investigate the main difficulties primary school students face during activities involving programming and establish a relationship between these difficulties and interaction problems in the main tools. Through a review of the scientific literature, a general overview was obtained which identified "Repetition structures" and "Use of variables" as the most common difficulties, in addition to 50% of the difficulties encountered being related to interaction and interface problems.*

**Resumo.** *A compreensão da tecnologia vem sendo amplamente abordada na educação básica. Muitas escolas abordam conceitos como pensamento computacional e programação na sala de aula. Este trabalho objetiva investigar as principais dificuldades enfrentadas por estudantes da educação básica durante atividades envolvendo programação, bem como estabelecer uma relação entre tais dificuldades e problemas de interação nas principais ferramentas. Por meio da revisão da literatura científica, obteve-se um panorama que sinalizou "Estruturas de repetição" e o "Uso de variáveis" como as dificuldades mais presentes, além de 50% das dificuldades encontradas relacionarem-se a problemas de interação e interface.*

## 1. Introdução

O cotidiano tem sido permeado pelo crescimento da tecnologia, que se introduz cada vez mais precocemente na vida das pessoas, inclusive durante a infância. Em vista disso, é importante estimular o conhecimento de conceitos computacionais desde o começo da jornada escolar, ou seja, ao longo da educação básica [Kanaki and Kalogiannakis 2023]. Nesse contexto, os benefícios da introdução dessas iniciativas no ambiente escolar são diversos. Por exemplo, atividades que abordam ideias relacionadas à computação tornam as aulas mais dinâmicas, interessantes e inovadoras [Ponce et al. 2019, Fuste and Schmandt 2019, Jormanainen and Tukiainen 2021], além de servirem como estratégia metodológica adicional, incentivando alunos com dificuldades ou desinteresse em determinadas disciplinas [Castillo et al. 2019].

Além disso, é notório que a aquisição de novos conhecimentos frequentemente encontra obstáculos. Assim, na realização de atividades envolvendo Pensamento Computacional e Programação, os estudantes podem relatar dificuldades que limitaram seu desempenho no cumprimento das tarefas designadas. Desta forma, faz-se necessário investigar quais são essas dificuldades e se estão relacionadas com problemas de interação nas ferramentas utilizadas, a fim de mitigar esses problemas e melhorar o ensino-aprendizagem e a aplicabilidade das estratégias metodológicas. Para isso, utilizaram-se renomadas bases de dados científicas para análise de publicações no intervalo de 2019 a 2023. Os critérios de inclusão e exclusão foram definidos com base no público-alvo, objetivo e existência de dificuldades. Os resultados mostraram as principais dificuldades dos alunos da educação básica em tarefas envolvendo Pensamento Computacional e Programação, além dos *softwares* de programação mais utilizados e as dificuldades de aprendizagem encontradas.

Quanto à estrutura, o artigo está dividido respectivamente em: *Introdução*, *Trabalhos Relacionados*, *Mapeamento Sistemático da Literatura*, *Resultados*, *Discussão*, *Considerações Finais e Referências*. A *Introdução* apresenta os conceitos essenciais para compreender a pesquisa. Na parte dedicada aos *Trabalhos Relacionados*, são mencionados estudos científicos similares ao atual artigo, enfatizando os objetivos específicos de cada um e estabelecendo conexões com o contexto. A *Mapeamento Sistemático da Literatura* trata das *Questões de Pesquisa*, ou seja, as perguntas estabelecidas para pesquisa; *Critérios de Inclusão e Exclusão* instituem quais parâmetros selecionam os artigos que serão incluídos ou excluídos da análise conforme as *Questões de Pesquisa*; *Protocolo de Pesquisa* indica as bases de dados utilizadas, os critérios de busca e a temática da pesquisa. A subdivisão de *Resultados* engloba as respostas derivadas do objeto de estudo, apresentadas por meio de representações gráficas e percentuais. A partir desses resultados, foram construídos argumentos sobre a relevância dos achados e as situações em que se aplicam, sendo detalhados na seção de *Discussão*. Para concluir, a seção de *Considerações Finais* mostra uma visão geral do artigo, destacando considerações futuras e pontos-chave discutidos no texto.

## 2. Trabalhos Relacionados

Um Mapeamento Sistemático da Literatura (MSL) tem por objetivo analisar a tendência de determinado campo de estudo na comunidade científica, além de informar e identificar tópicos de pesquisas futuras ou lacunas existentes em tal setor. Nesse sentido, [Gardner et al. 2022] estabeleceram um estudo que tem por objetivo examinar iniciativas de aprendizagem não formais, estabelecendo uma visão panorâmica sobre qual tem sido o foco de pesquisas recentes sobre o assunto na educação básica e os impactos provenientes dessas iniciativas no contexto escolar.

Também, a pesquisa de [Brunsek et al. 2020] desenvolveram um mapeamento sistemático que examina a importância do desenvolvimento profissional (DP) na melhoria da prática em programas de educação e acolhimento na primeira infância (EAPI), abordando a dificuldade em determinar onde investir recursos limitados, dada a variedade de programas de DP disponíveis.

Outrossim, [Liang et al. 2021] realizaram um levantamento bibliográfico a respeito de literaturas científicas que utilizam Interfaces de Usuários Tangíveis (IUT) (co-

nhecidas como *Interfaces Tangíveis* ou *Interfaces Físicas*) para apoiar a aprendizagem criativa de crianças, mostrando também as vantagens de utilização das IUT e em quais contextos essa estratégia metodológica é mais aplicada. Igualmente, [Martins et al. 2023] produziu um mapeamento sistemático sobre o ensino do pensamento computacional (PC) para pré-escolares, destacando o aumento do interesse dos pesquisadores nesse tema nos últimos anos.

No entanto, nota-se que os estudos mencionados analisaram quesitos específicos, a exemplo da investigação de abordagens educacionais informais na educação básica [Gardner et al. 2022]; como está o emprego de Interfaces de Usuário Tangíveis na aprendizagem de crianças [Liang et al. 2021]; a melhoria de programas de educação na primeira infância por meio do desenvolvimento profissional (DP) [Brunsek et al. 2020] e a incidência do ensino de pensamento computacional em pré-escolares (na faixa etária de 4 a 5 anos) e quais atividades vêm sendo implementadas [Martins et al. 2023], mas não citam a dificuldade na aplicação dessas estratégias, tampouco mencionam se tais dificuldades estão relacionados à interação do usuário, por exemplo.

Assim, o presente estudo identificou as dificuldades frequentemente encontradas em tarefas de programação, examinar as ferramentas predominantes em tais ações e investigar a possível relação com problemas de interação. Para atingir esse propósito, foram exploradas evidências científicas nas renomadas bases de dados *ACM Digital Library*, *Science Direct* e *SBC OpenLib*.

### 3. Mapeamento Sistemático da Literatura

Buscou-se investigar a existência de dificuldades relacionadas ao ensino de programação na educação básica, além de saber quais são essas dificuldades e se estão relacionadas a problemas de interação, tendo foco em produções cujo público-alvo sejam estudantes da educação básica.

Neste estudo, a investigação foi baseada na estrutura definida em [Petersen et al. 2015] e delineada em três fases: *planejamento*, composto pela estruturação, protocolos de pesquisa e avaliação dos protocolos, além da definição de objetivos; a fase de *execução*, com a identificação de pesquisas orientadoras do tema central, extração e coleta de dados para a produção de conhecimento; e a fase de apresentação dos *resultados*, que descreve os resultados obtidos e a avaliação das pesquisas selecionadas. É relevante enfatizar que os autores deste artigo conduziram todas as fases descritas, conforme detalhado, com o propósito de avaliar cada publicação à luz dos critérios informados nas seções subsequentes.

#### 3.1. Questões de Pesquisa

O questionamento principal desta pesquisa é: *Quais são as dificuldades relacionadas às práticas de programação na educação básica, conforme a literatura científica?* Além disso, formularam-se outras perguntas:

- Das principais dificuldades de aprendizagem, quais as ferramentas de programação utilizadas?
- Essas dificuldades de aprendizagem estão relacionadas com problemas de interação da ferramenta utilizada?
- No período de 2019 a 2023, quantos estudos científicos foram gerados dentro desses critérios de inclusão?

### 3.2. Critérios de Inclusão e Exclusão

Os Critérios de Inclusão (CI) foram definidos para selecionar estudos que eram consonantes com os objetivos desta MSL. Em oposição a estes, foram definidos Critérios de Exclusão (CE) para filtrar publicações que não atenderam a um ou mais critérios de inclusão. São eles:

- CI01 - Público-alvo deve ser estudante da educação básica
- CI02 - Atividade principal envolve programação
- CI03 - Artigo citar dificuldades dos alunos durante atividades de programação
- CE01 - Público-alvo pertencer ao nível superior, professores, pais ou outros públicos
- CE02 - Objetivo não promove o ensino da programação
- CE03 - Artigo duplicado ou fora do eixo de pesquisa, anais de eventos

### 3.3. Protocolo de Pesquisa

A definição das *strings* de pesquisa no presente Protocolo foi determinada pelas bibliotecas digitais *ACM Digital Library*, *ScienceDirect* e *SBC OpenLib*, conforme exibido na Tabela 1. As duas primeiras são coletâneas internacionais e relevantes para a comunidade científica, bem como a *SBC OpenLib* contém estudos científicos brasileiros que mostram um panorama nacional acerca do eixo temático em questão.

Vale ressaltar que apesar da sintaxe exclusiva de cada biblioteca, as *strings* diferem em relação aos termos de busca devido à limitação de encontrar artigos relacionados ao tema principal.

**Tabela 1. Descrição das Strings de busca e base de dados pesquisadas**

Bases de busca	Strings de busca
ACM Digital Library	(education AND school) AND ("kid difficulty*" OR children) AND (computational thinking OR programming) AND Publication Date: (01/01/2019 TO 12/31/2023)
ScienceDirect	school AND (kid OR children) AND (computational thinking OR programming) AND difficult AND Publication Years: 2019-2023
SBC OpenLib	(educação OR escola* AND ("criança*" OR "estudante*" OR "pensamento computacional" OR programação AND ("dificuldade*")))

O protocolo de pesquisa apresentado contou com a participação dos 4 (quatro) autores do artigo, que foram responsáveis por avaliar cada publicação com base na análise dos resumos e na leitura completa dos trabalhos encontrados nas bases de dados. Eles também foram responsáveis pela seleção e/ou exclusão dos trabalhos, seguindo os critérios de inclusão e exclusão estabelecidos. Dos 228 (duzentos e vinte e oito) estudos inicialmente identificados, 99 (noventa e nove) foram selecionados para a coleta de dados, pois atendiam a pelo menos um dos critérios de inclusão.

Ademais, é importante destacar que a escolha do intervalo dos anos de 2019 a 2023 oferece atualizações significativas sobre o tópico em questão, evidenciar o progresso

do assunto em estudo e proporcionar uma visão recente que revele as tendências atuais e a situação atual da pesquisa.

## 4. Resultados

### 4.1. Quais são as dificuldades relacionadas às práticas de programação na educação básica, conforme a literatura científica?

Vale destacar que dentre as dificuldades enfrentadas pelos alunos da educação básica, as mais expressivas foram *Uso de Variáveis*, *Estruturas de Repetição* e *Comandos de entrada/saída*, respectivamente. O *Uso de variáveis* está diretamente relacionado às dificuldades envolvendo a articulação de como inicializar, referenciar e atualizar uma variável, inclusive em booleanos [Luo et al. 2022, Genesio et al. 2023]. Já as dificuldades relacionadas à *Estrutura de repetição* referem-se ao manuseio de *loops* de aninhamento [Choi 2023, Bentz and Standl 2023].

Outras dificuldades também foram citadas em menor prevalência. É o caso da *Otimização/Decomposição*, que decorre de análises algorítmicas, de otimização, decomposição ou depuração na programação [Bryndová and Mališů 2020]. O *Conhecimento da língua inglesa* evidencia o desafio na utilização de ferramentas computacionais devido à limitação no conhecimento do idioma inglês presente nessas [Choi 2023]. A *Sintaxe* é caracterizada por dificuldades vinculadas a sintaxes específicas de determinadas linguagens de programação, como o *Python*, e problemas na digitação [Qian et al. 2019].

Os *Símbolos de pontuação* evidenciam a dificuldade de manipular a pontuação correta no código [Qian et al. 2019]. A *Instabilidade do software ou internet* envolve adversidades de conexão de internet ou ambientes de aplicação online fora do ar [Galdo et al. 2022, Genesio et al. 2023]. Os *Comandos condicionais* tratam de dificuldades envolvendo estruturas condicionais "se... então..." [Pinheiro et al. 2019]. Já os *Comandos de entrada/saída* referem-se às dificuldades envolvendo o *input* do *Python* [Qian et al. 2019], por exemplo. A *Ortografia* [Qian et al. 2019] pauta-se na dificuldade em escrever as palavras corretamente, e está diretamente ligada às outras dificuldades, como a *Sintaxe* e *Comandos de entrada/saída*. É importante evidenciar que existiram artigos que abordaram mais de uma dificuldade enfrentada, observadas na Figura 1.

### 4.2. Das principais dificuldades de aprendizagem, quais as ferramentas de programação utilizadas?

Em relação à Pergunta 2 do trabalho (*Das principais dificuldades de aprendizagem, quais as ferramentas de programação utilizadas?*) uma diversidade de *softwares*, com destaque para a ferramenta *Scratch*, *Pencil Code* e *Hour of Code*, ambas baseadas em linguagens de programação visuais. Todavia, outras ferramentas e métodos de aplicação também estiveram presentes (como a computação desplugada), e em alguns artigos foram utilizadas mais de uma ferramenta em sua definição (Figura 2).

### 4.3. Essas dificuldades de aprendizagem estão relacionadas com problemas de interação da ferramenta utilizada?

A partir das dificuldades expostas, observou-se que cerca de 50% dessas estão relacionadas a problemas de interação nas ferramentas utilizadas (Pergunta 3). Nesse sentido,

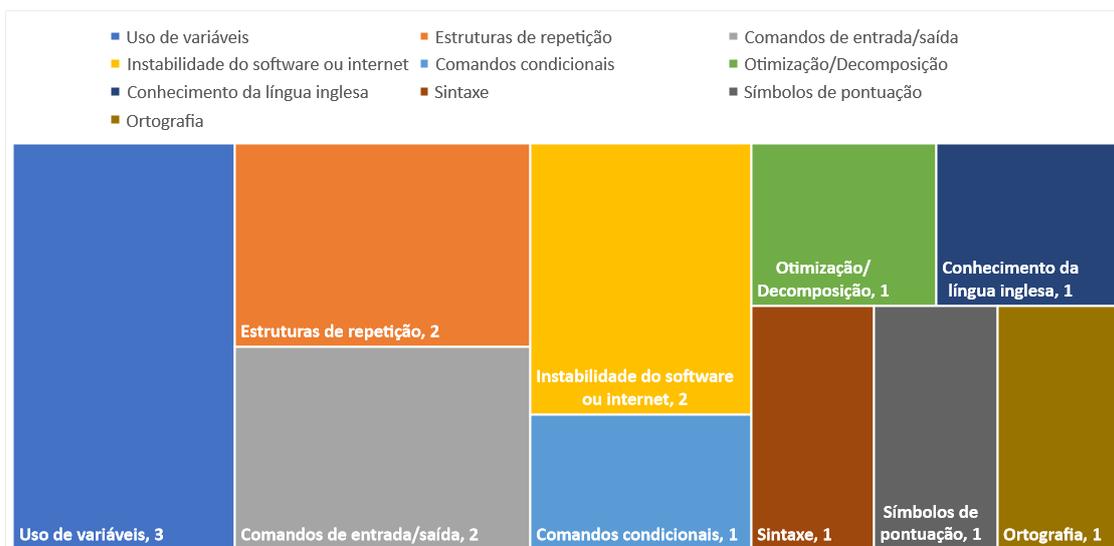


Figura 1. Principais dificuldades abordadas

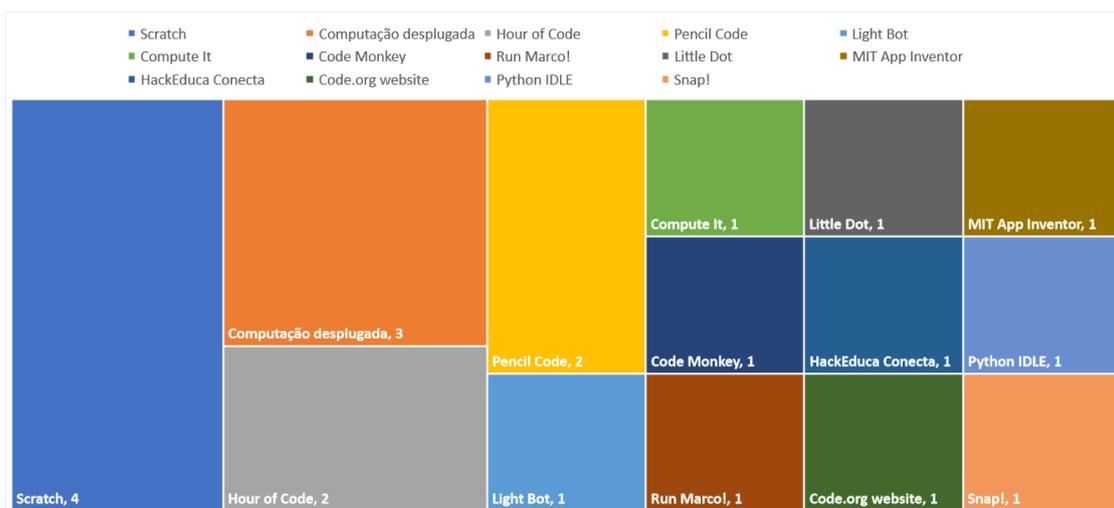


Figura 2. Principais ferramentas de programação utilizadas

identificaram-se tais problemas de usabilidade, buscou-se realizar uma inspeção nas interfaces a fim de identificar e classificar tais problemas citados e outros pertinentes.

Desta forma, uma dificuldade de aprendizagem relevante e relacionada à interação são as *Estruturas de repetição* [Choi 2023, Bentz and Standl 2023], em que os blocos visuais apresentam uma estrutura muito compacta, conforme visualizado na Figura 3. Essa disposição dos elementos dificulta a rápida interpretação pelo usuário, especialmente quando se trata de crianças que dependem de uma compreensão imediata do conteúdo visual para assimilar determinados conceitos.

Uma das dificuldades de aprendizagem também relacionadas à interação e citadas no uso de ferramentas para o ensino de programação foi a predominância do *Conhecimento da língua inglesa* na maioria das aplicações (*Scratch*, *Pencil Code*, *Light Bot*, *Hour of Code*, *MIT App Inventor*) [Branco et al. 2021], conforme as Figuras 4 e 5.

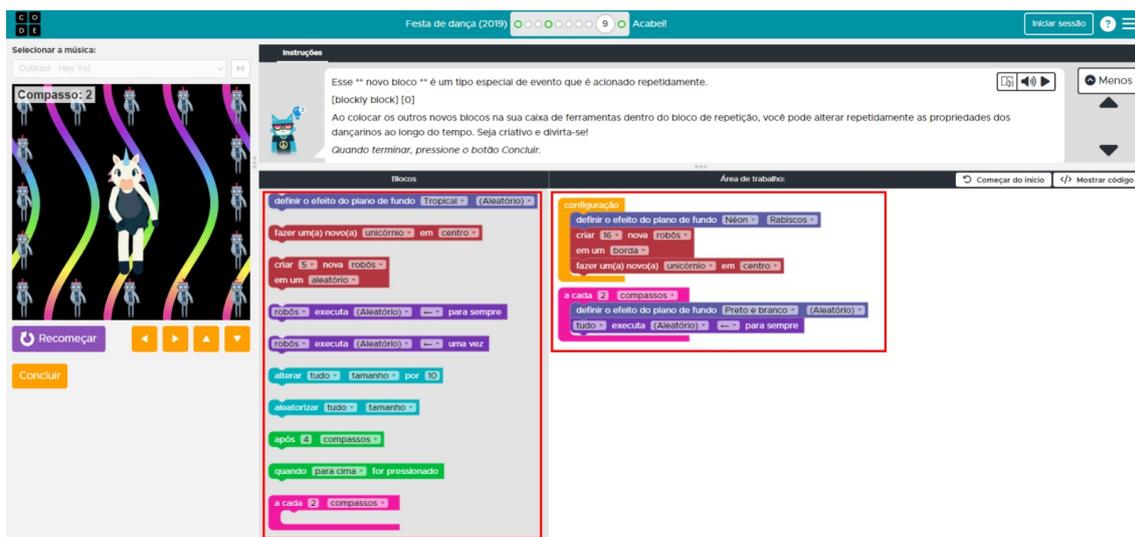


Figura 3. Code.org website - Presença de muitos blocos aninhados para entendimento de estrutura de repetição, respectivamente.



Figura 4. Code.org website - Mensagem de erro disponível apenas em inglês.

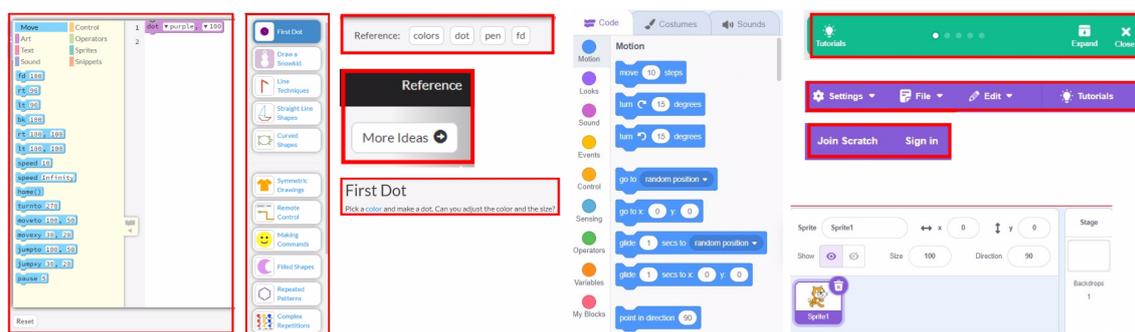
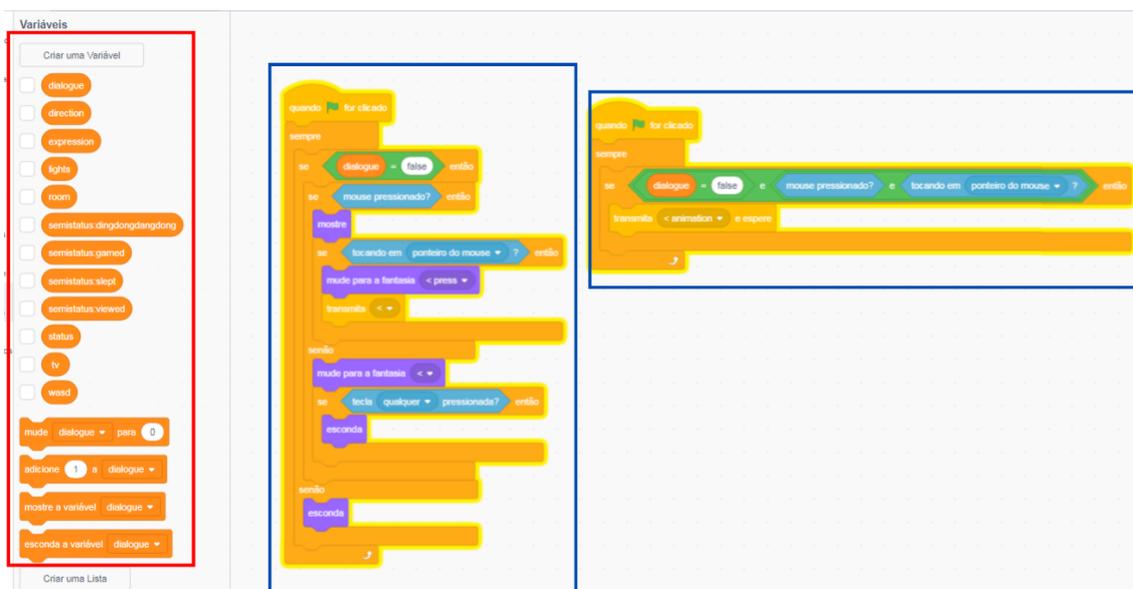


Figura 5. Exemplo de predominância da língua inglesa no Pencil Code e Scratch

Quanto à questão da dificuldade do *Uso de variáveis* e *Comandos de entrada/saída* no Scratch [Luo et al. 2022, Genesisio et al. 2023], nota-se a excesso de blocos que, por um lado, podem ser benéficos para reproduzir uma ação detalhada, mas também podem confundir o usuário (Figura 6), pois em determinados blocos é necessário a seleção de um número notável de elementos.



**Figura 6. Scratch - Em vermelho, sinaliza-se o problema de uso de variáveis; em azul, para entrada e saída de dados.**

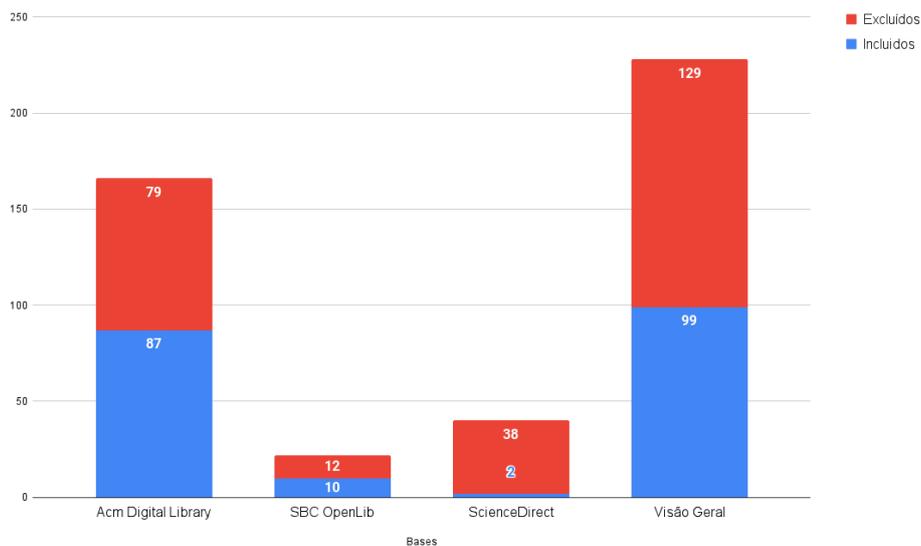
Fatores como *Sintaxe*, *Símbolos de pontuação e Ortografia* [Qian et al. 2019] também foram decorrentes de problemas de interação, visto que existem barreiras consideráveis no processo de digitação em alguns casos específicos. Por exemplo, é notória a necessidade de haver uma atenção dedicada aos usuários de línguas não-ocidentais, os quais possuem símbolos parecidos com os dos alfabetos ocidentais, mas com usos diferentes (ponto e vírgula)[Qian et al. 2019], o que pode causar dúvidas durante a manipulação da ferramenta, prejudicando diretamente o ato de programar.

Todavia, é importante salientar que nem todas as dificuldades de aprendizagem se relacionaram a problemas de interação. Por exemplo, em alguns casos, a estratégia metodológica utilizada foi a *computação desplugada* (ensino computacional sem uso de computadores) [Bryndová and Mališů 2020, Pinheiro et al. 2019], logo não houve um *software* para inferir problemas de interação.

#### **4.4. No período de 2019 a 2023, quantos estudos científicos foram gerados dentro desses critérios de inclusão?**

Obteve-se como resposta da Pergunta 4 o número total de 228 (duzentos e vinte e oito) artigos primários, sendo 99 (noventa e nove) incluídos e 129 (cento e vinte e nove) excluídos, conforme a Figura 7.

Os 99 (noventa e nove) trabalhos científicos incluídos na pesquisa representam 43,42% dos estudos selecionados, ou seja, em relação ao total desde o início da pesquisa. Os critérios CI01/CI02 (atividades de programação voltadas para alunos da educação básica) são os mais abordados, correspondendo a 82,47%. Em contrapartida, apenas 3,09% representa o critério de inclusão CI02 (atividade principal envolve programação). Assim, compreende-se a incidência de pesquisas envolvendo o ensino de programação na educação básica, mas sem citar alguma dificuldade de aprendizagem ao longo dessa atividade.



**Figura 7. Visão Geral da extração de dados da pesquisa.**

Por outro lado, o CE03 (*Público-alvo pertencer ao nível superior, aos professores, pais ou outros públicos*) apresentou maioria de estudos científicos durante as buscas nas bases de dados selecionadas, seguido do CE01, que indica trabalhos cujo objetivo não promove o ensino de programação (Figura 8).



**Figura 8. Extração de dados da pesquisa - Apenas critérios de exclusão**

Assim, obteve-se apenas 9 (nove) estudos que abrangeram os três critérios de inclusão, ou seja, relataram dificuldades de aprendizagem durante as atividades envolvendo programação por parte do público-alvo em questão, listados na Tabela 2

## 5. Discussão

O questionamento principal da pesquisa *Quais são as dificuldades relacionadas às práticas de programação na educação básica, conforme a literatura científica?* revelou

**Tabela 2. Listagem completa dos estudos que obtiveram todos os critérios de inclusão estipulados**

Estudos	Títulos	Local de publicação
[Branco et al. 2021]	Programming for Children and Teenagers in Brazil: A 5-year Experience of an Outreach Project	SIGCSE 21: 52nd ACM Technical Symposium on Computer Science Education
[Choi 2023]	The Influence of Code.org on Computational Thinking and Learning Attitude in Block-Based Programming Education	ICEEL 22: 2022 6th International Conference on Education and E-Learning
[Qian et al. 2019]	Using Data to Understand Difficulties of Learning to Program: A Study with Chinese Middle School Students	CompEd '19: ACM Conference on Global Computing Education
[Luo et al. 2022]	Elementary Students' Understanding of Variables in Computational Thinking-Integrated Instruction: A Mixed Methods Study	SIGCSE 2022: 3rd ACM Technical Symposium on Computer Science Education
[Bryndová and Mališů 2020]	Assessing the Current Level of the Computational Thinking Within the Primary and Lower Secondary School Students using Educational Robotics Tasks	ICEMT 2020: The 4th International Conference on Education and Multimedia Technology
[Galdo et al. 2022]	Pair Programming in a Pandemic: Understanding Middle School Students' Remote Collaboration Experiences	SIGCSE 2022: 53rd ACM Technical Symposium on Computer Science Education
[Bentz and Standl 2023]	Novice Programmers Conceptions of Loops in K-12 Education in Consideration of Interest and Ability	WiPSCE 23: 18th WiPSCE Conference on Primary and Secondary Computing Education Research
[Genesio et al. 2023]	Aprendendo Lógica de Programação Desenvolvendo Jogos Digitais: Um Relato de Experiência	CBIE - Workshop de Informática na Escola (WIE)
[Pinheiro et al. 2019]	Programa(ção): Atividades lúdicas para ensino de programação em escolas públicas	WEI - Workshop sobre Educação em Computação

7 (sete) problemas principais: *Uso de variáveis, Instabilidade do software ou internet, Conhecimento da língua inglesa, Ortografia, Estruturas de repetição, Comandos condicionais, Sintaxe, Comandos de entrada/saída, Otimização/Decomposição e Símbolos de pontuação.*

A investigação e os resultados da pesquisa foram cruciais para acompanhar as principais dificuldades envolvendo programação, tendo como público-alvo os estudantes da educação básica. Entretanto, foi revelado que existe uma escassez de estudos científicos que mencionem dificuldades desse público, pois apenas 9 (nove) trabalhos foram selecionados. Além disso, a partir das dificuldades expostas, observou-se que cerca de metade desses trabalhos apresentam dificuldades relacionadas às questões de interação das ferramentas. A fim de mitigar tais barreiras, buscou-se propostas de soluções viáveis para cada ferramenta utilizada de acordo com cada problema relatado.

Frente à estrutura compacta das *Estruturas de repetição* [Choi 2023, Bentz and Standl 2023] (Figura 3), uma alternativa que permite que o sistema seja viável tanto para os usuários experientes quanto inexperientes é a implementação de recursos (como um botão, por exemplo) para auxiliar na manipulação do aninhamento, além de pequenos tutoriais que auxiliem o estudante a discernir os diferentes tipos de estruturas de repetições contidas na Figura 3. Isso pode ser feito pela criação de um assistente virtual, visando facilitar a comunicação entre o sistema e o usuário, que, ao ser questionado sobre determinada função, explique seu funcionamento e aplique isso em determinado exemplo, demonstrando tal funcionamento. Desta forma, é crucial existirem

mensagens de erro claras. Porém, em algumas ferramentas, a mensagem de erro está na língua inglesa, mesmo após configurar outro idioma, prejudicando a identificação do problema com exatidão por parte do usuário (Figura 4).

Por outro lado, o *Conhecimento da língua inglesa* nas ferramentas (*Scratch*, *Pencil Code*, *Light Bot*, *Hour of Code*, *MIT App Inventor*) [Branco et al. 2021], conforme a Figura 5, pode ser diminuído a partir da adequação do *software* e suas funções à linguagem conhecida pelo participante (termos familiares aos usuários), em vez de empregar linguagem técnica, esta na maioria expressa em inglês.

Quanto à dificuldade do *Uso de variáveis* e *Comandos de entrada/saída* no *Scratch* [Luo et al. 2022, Genesio et al. 2023], a quantidade de blocos que podem confundir o usuário (Figura 6) pode ser amenizada. Para isso, recomenda-se criar um modo simplificado para a criação de projetos no *Scratch*, a fim de facilitar a aprendizagem de alunos iniciantes e proporcionar maior familiaridade com a ferramenta.

Na perspectiva de *Sintaxe*, *Símbolos de pontuação e Ortografia*, [Qian et al. 2019] é notória a necessidade de haver uma atenção dedicada aos usuários de línguas não-ocidentais [Qian et al. 2019]. Portanto, é relevante a adoção de alerta que sinalize ao usuário que determinado símbolo ou sintaxe possa estar errado e demonstrem possíveis alternativas para consertar o código, uma possibilidade que não existe na ferramenta *Python IDLE*. Com isso, será possível ter uma interação mais positiva com o programa, tendo em vista que custará menos tempo na resolução do problema no código.

Por fim, entende-se que as implicações supracitadas, à primeira vista, podem não demonstrar um problema de grande magnitude. No entanto, ao analisá-las do ponto de vista dos estudantes do ensino básico, isso poderá afetar diretamente seu desempenho durante o processo de aprendizagem, uma vez que não tendo uma experiência muito satisfatória durante o primeiro contato com a plataforma, o usuário poderá ter uma maior tendência a não se interessar facilmente durante seu primeiro contato. Isso poderá resultar em um maior desinteresse em se aprofundar nos estudos sobre programação. Logo, as ferramentas alisadas demonstram ser promissoras no ensino de programação, mas ainda se fazem necessários refinamentos para que a experiência do usuário seja mais positiva.

## 6. Considerações Finais

Pelo estudo apresentado, os resultados provenientes na literatura sobre dificuldades relacionadas ao aprendizado de programação em escolares pertencentes à educação básica foram ínfimos (9 estudos de 228 totais). Ou seja, evidencia uma baixa existência de trabalhos científicos que relatem as dificuldades de estudantes sobre tal assunto.

Além disso, mostrou-se que a programação baseada em blocos (em destaque o *Scratch*) é uma alternativa que possui grande potencial para a introdução de noções relacionadas à Programação para crianças e adolescentes no contexto escolar, visto que o fácil manuseio e o *design* colorido são meios que facilitam a disseminação do conhecimento associado à ludicidade. Desta forma, destacou-se a relevância da investigação acerca de dificuldades envolvendo conceitos de computação. Isso contribui na base teórica e prática para avanços futuros no ramo da educação em computação e também no encorajamento dos professores no ensino de conteúdos relacionados à informática e à tecnologia, capacitando-os [Martins et al. 2023].

O trabalho apresentou limitações em relação ao acesso a determinadas ferramentas mencionadas na literatura científica, necessárias para realizar testes e demonstrar evidências no presente estudo. Houve desafios no acesso e uso dessas ferramentas devido ao custo elevado, frequentemente em dólares, além das dificuldades associadas ao seu manuseio, muitas vezes complexo e de difícil compreensão.

Assim, os trabalhos futuros desta pesquisa norteiam o desenvolvimento de estudos que investiguem as dificuldades de aprendizagem nos próximos anos, dada a escassez de pesquisas relacionadas ao levantamento e acompanhamento das dificuldades enfrentadas por alunos na educação básica. Além disso, é crucial tornar o processo de ensino-aprendizagem cada vez mais motivador para o público-alvo, visando a redução das dificuldades durante as práticas de programação na educação básica. Para alcançar esse objetivo, é necessário o desenvolvimento de produções científicas que abordem formas de implementar estratégias metodológicas eficazes, flexíveis e inovadoras, visando oferecer atividades significativas que atendam às diversas formas de aprendizado dos alunos.

## Referências

- Bentz, A. and Standl, B. (2023). Novice programmers conceptions of loops in k-12 education in consideration of interest and ability. In *Proceedings of the 18th WiPSCE Conference on Primary and Secondary Computing Education Research*, WiPSCE '23, New York, NY, USA. Association for Computing Machinery.
- Branco, A., Dutra, C., Zumpichiatti, D., Campos, F. A., SantClair, G., Mello, J., Moreira, J. a. V., Godinho, J., Marotti, J., and Gomide, J. (2021). Programming for children and teenagers in brazil: A 5-year experience of an outreach project. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, SIGCSE '21, page 411–417, New York, NY, USA. Association for Computing Machinery.
- Brunsek, A., Perlman, M., McMullen, E., Falenchuk, O., Fletcher, B., Nocita, G., Kamkar, N., and Shah, P. S. (2020). A meta-analysis and systematic review of the associations between professional development of early childhood educators and children's outcomes. *Early Childhood Research Quarterly*, 53:217–248.
- Bryndová, L. and Mališů, P. (2020). Assessing the current level of the computational thinking within the primary and lower secondary school students using educational robotics tasks. In *Proceedings of the 4th International Conference on Education and Multimedia Technology*, ICEMT '20, page 239–243, New York, NY, USA. Association for Computing Machinery.
- Castillo, R. E., Cheng, C. J., Agustin, J. S., and Aragon, M. C. R. (2019). Development of an educational mobile game applications for grade 5 for knowledge channel inc. In *Proceedings of the 2nd International Conference on Information Science and Systems*, ICISS '19, page 99–104, New York, NY, USA. Association for Computing Machinery.
- Choi, W. C. (2023). The influence of code.org on computational thinking and learning attitude in block-based programming education. In *Proceedings of the 2022 6th International Conference on Education and E-Learning*, ICEEL '22, page 235–241, New York, NY, USA. Association for Computing Machinery.
- Fuste, A. and Schmandt, C. (2019). Hypercubes: A playful introduction to computational thinking in augmented reality. In *Extended Abstracts of the Annual Symposium on*

- Computer-Human Interaction in Play Companion Extended Abstracts*, CHI PLAY '19 Extended Abstracts, page 379–387, New York, NY, USA. Association for Computing Machinery.
- Galdo, A. C., Celepkolu, M., Lytle, N., and Boyer, K. E. (2022). Pair programming in a pandemic: Understanding middle school students' remote collaboration experiences. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education - Volume 1*, SIGCSE 2022, page 335–341, New York, NY, USA. Association for Computing Machinery.
- Gardner, T., Leonard, H. C., Waite, J., and Sentance, S. (2022). What do we know about computing education for k-12 in non-formal settings? a systematic literature review of recent research. In *Proceedings of the 2022 ACM Conference on International Computing Education Research - Volume 1*, ICER '22, page 264–281, New York, NY, USA. Association for Computing Machinery.
- Genesio, N., Menezes, M., Almeida, J., Boaventura, A., and Valle, P. (2023). Aprendendo lógica de programação desenvolvendo jogos digitais: Um relato de experiência. In *Anais do XXIX Workshop de Informática na Escola*, pages 375–386, Porto Alegre, RS, Brasil. SBC.
- Jormanainen, I. and Tukiainen, M. (2021). Attractive educational robotics motivates younger students to learn programming and computational thinking. In *Eighth International Conference on Technological Ecosystems for Enhancing Multiculturality*, TEEM'20, page 54–60, New York, NY, USA. Association for Computing Machinery.
- Kanaki, K. and Kalogiannakis, M. (2023). Algorithmic thinking in early childhood. In *Proceedings of the 6th International Conference on Digital Technology in Education*, ICDTE '22, page 66–71, New York, NY, USA. Association for Computing Machinery.
- Liang, M., Li, Y., Weber, T., and Hussmann, H. (2021). Tangible interaction for children's creative learning: A review. In *Proceedings of the 13th Conference on Creativity and Cognition*, CC '21, New York, NY, USA. Association for Computing Machinery.
- Luo, F., Yan, W., Liu, R., and Israel, M. (2022). Elementary students' understanding of variables in computational thinking-integrated instruction: A mixed methods study. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education - Volume 1*, SIGCSE 2022, page 523–529, New York, NY, USA. Association for Computing Machinery.
- Martins, E. C., da Silva, L. G. Z., and de Almeida Neris, V. P. (2023). Systematic mapping of computational thinking in preschool children. *International Journal of Child-Computer Interaction*, 36:100566.
- Petersen, K., Vakkalanka, S., and Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and software technology*, 64:1–18.
- Pinheiro, J., Godinho, J., Guedes, Y., Cardoso, G., Zumpichiatti, D., and Gomide, J. (2019). Programa(ção): Atividades lúdicas para ensino de programação em escolas públicas. In *Anais do XXVII Workshop sobre Educação em Computação*, pages 91–100, Porto Alegre, RS, Brasil. SBC.

Ponce, P., Molina, A., Mata, O., and Baltazar, G. (2019). Lego® ev3 platform for stem education in elementary school. In *Proceedings of the 2019 8th International Conference on Educational and Information Technology, ICEIT 2019*, page 177–184, New York, NY, USA. Association for Computing Machinery.

Qian, Y., Yan, P., and Zhou, M. (2019). Using data to understand difficulties of learning to program: A study with chinese middle school students. In *Proceedings of the ACM Conference on Global Computing Education, CompEd '19*, page 185–191, New York, NY, USA. Association for Computing Machinery.