

Ferramentas digitais lúdicas no ensino de estruturas de repetição: um mapeamento sistemático da literatura

Playful digital tools in teaching repetition structures: a systematic mapping of the literature

Lukas Kardeck de Souza Vianna¹, Almir David Valente Santiago¹

¹Universidade Estadual do Sudoeste da Bahia

lukas.kardeck@gmail.com, almir.santiago@gmail.com

Resumo

Este artigo apresenta um mapeamento sistemático da literatura com o objetivo de identificar e analisar o uso de ferramentas digitais lúdicas no ensino do conceito de estruturas de repetição em programação. Foram investigadas quatro bases científicas internacionais, resultando na seleção de cinco estudos publicados entre 2013 e 2018. A análise revelou treze ferramentas com abordagens variadas, baseadas em jogos, simulações e ambientes visuais de blocos. Os resultados indicam que representações visuais, feedback imediato e progressão gradual das atividades contribuem para reduzir dificuldades comuns no aprendizado de loops, especialmente no rastreamento de variáveis e na compreensão do fluxo iterativo. Observou-se, contudo, que apenas uma das ferramentas analisadas foi projetada especificamente para o ensino de estruturas de repetição. As demais abordam o conceito de forma indireta, o que evidencia uma lacuna relevante na literatura. Com base nos padrões identificados, foram sintetizados requisitos essenciais para orientar o desenvolvimento de ferramentas mais eficazes, envolvendo visualização animada, suporte pedagógico, autonomia, interatividade e equilíbrio entre elementos lúdicos e rigor conceitual. O estudo reforça o potencial das ferramentas digitais lúdicas no ensino de programação, ao mesmo tempo em que destaca a necessidade de pesquisas com metodologias mais robustas.

Palavras-chave: ensino de programação; ferramentas lúdicas; estruturas de repetição; loops; tecnologias educacionais; mapeamento sistemático.

Abstract

This article presents a systematic mapping of the literature aimed at identifying and analyzing the use of playful digital tools in teaching the concept of repetition structures in programming. Four international scientific databases were examined, resulting in the selection of five studies published between 2013 and 2018. The analysis revealed thirteen tools employing varied approaches based on games, simulations, and visual block-based environments. The results indicate that visual representations, immediate feedback, and gradual activity progression help reduce common difficulties in learning loops, especially in variable tracing and understanding iterative flow. However, only one of the analyzed tools was specifically designed for teaching repetition structures; the others address the concept indirectly, highlighting a relevant gap in the literature. Based on the identified patterns, essential requirements were synthesized to guide the development of more effective tools, involving animated visualization, pedagogical support, autonomy, interactivity, and a balance between playful elements and conceptual precision. The study reinforces the potential of playful digital tools in programming education while also emphasizing the need for research with more robust methodologies.

Keywords: programming education; playful tools; repetition structures; loops; educational technology; systematic mapping.

Introdução

Diversos estudos sobre o ensino e a aprendizagem na área da Computação têm se debruçado sobre as dificuldades enfrentadas por alunos iniciantes em cursos de computação e engenharias, especialmente nas disciplinas introdutórias de programação de computadores [1-3]. Esses trabalhos destacam não apenas as barreiras cognitivas encontradas pelos estudantes, mas também os elevados índices de reprovação e desistência observados em instituições de ensino superior.

De forma mais específica, autores de [1] argumentam que uma metodologia de ensino baseada exclusivamente em sintaxe não é a mais eficaz para o processo de aprendizagem. Já os autores de [2] apontam que muitos estudantes não se sentem motivados devido à dificuldade de construir modelos mentais corretos sobre os conceitos ensinados. Complementando essa perspectiva, os autores de [3] defendem a necessidade de novas abordagens instrucionais que não se baseiam apenas em código textual.

No âmbito das dificuldades enfrentadas, o conceito de “estruturas de repetição” destaca-se como um dos primeiros grandes desafios para os alunos iniciantes em programação de computadores. Essa percepção é corroborada por pesquisa apresentada em [4], que, ao consultar mais de 300 docentes, apontou as estruturas de repetição como um dos tópicos mais difíceis de ensinar. Estudos mais recentes aprofundam essa constatação: [5] identificaram que iniciantes demonstram dificuldades em compreender o comportamento das variáveis em estruturas de repetição a cada iteração, enquanto [6] reforça que os “loops” continuam entre os conceitos mais difíceis de aprender devido à dificuldade na construção de modelos mentais e compreensão prática.

Diante desse cenário, este trabalho propôs pesquisar as estratégias descritas em estudos científicos que sugerem o uso de ferramentas digitais lúdicas para facilitar o ensino de programação, com foco no conceito de “estruturas de repetição”. Essa proposta dialoga com o que afirmam os autores de [3], ao destacarem que o desenvolvimento e o uso de tecnologias de software têm sido de grande interesse da comunidade acadêmica, especialmente aquelas que auxiliam o processo de ensino-aprendizagem por meio de ferramentas lúdicas que facilitam a visualização de conceitos complexos.

Estudos recentes, como o apresentado em [7], demonstram que o uso de ferramentas digitais interativas, como o GeoGebra, tem papel relevante no ensino de conteúdos abstratos. Segundo os autores, a aplicação do GeoGebra no ensino de cálculo diferencial possibilita uma melhor visualização dos conceitos, promove maior engajamento dos alunos e contribui significativamente para a superação de dificuldades conceituais e cognitivas enfrentadas por estudantes. Assim, considerando os desafios de aprendizagem e a difusão crescente das tecnologias de desenvolvimento de software, observa-se um campo fértil para contribuições significativas nesta pesquisa.

Esses aspectos ilustram a viabilidade do estudo e como o uso das tecnologias de desenvolvimento de software pode contribuir com inovações metodológicas de ensino não apenas na área da ciência da computação, mas também em diversos domínios do conhecimento. Dessa forma, os principais fatores que justificam a proposta deste projeto são: (i) as dificuldades identificadas no estudo dos conceitos de programação, em especial, das estruturas de repetição; e (ii) a percepção de que ferramentas digitais lúdicas podem proporcionar aos alunos uma melhor visualização e entendimento de conceitos complexos.

Considerando esse cenário e lacuna identificada na literatura, surge a seguinte questão que orientou a pesquisa: quais os principais requisitos para o desenvolvimento de uma ferramenta lúdica de ensino-aprendizagem para o conceito computacional de estruturas de repetição?

A partir desta questão, este estudo analisou as aplicações de ferramentas digitais lúdicas no ensino de estruturas de repetição em programação. Para alcançar esse propósito, foram estabelecidos os seguintes objetivos específicos:

1. Identificar ferramentas digitais lúdicas utilizadas no ensino de estruturas de repetição em programação;
2. Mapear as estratégias pedagógicas incorporadas nessas ferramentas;

3. Analisar como tais ferramentas auxiliam na compreensão das estruturas de repetição;
4. Discutir as potencialidades e limitações das abordagens encontradas.

2 Fundamentação Teórica

Autores de [8] citam que as matérias introdutórias de programação, são responsáveis pelas altas taxas de desistência dos alunos, e [9] ratifica essa ideia, afirmando que as desistências ocorrem já nos primeiros semestres dos cursos. Estudos como [3] afirmam que a quantidade de alunos que desistem dos cursos de computação após as primeiras matérias de programação chega a 40%. Segundo [10], os alunos não conseguem escrever programas razoáveis após dois semestres com matérias de programação e que os livros didáticos abordam apenas processos de programação sem a devida contextualização com problemas do mundo real.

Quanto às principais causas para o problema, [2] afirmam que a grande quantidade de processos cognitivos envolvidos na tarefa da programação a torna complexa. Os autores de [11] relatam os altos níveis de dificuldade impostos pelos conceitos da programação introdutória. Já [12] afirmam que a abordagem do ensino de programação tem sido a mesma há mais de quatro décadas.

Segundo [9], as expectativas dos jovens iniciantes para com os cursos de computação estão fora da realidade, pois esses cursos requerem, por exemplo, sólidas bases de conhecimentos em matemática. Os autores afirmam que as matérias de programação são planejadas para que os discentes desenvolvam habilidades cognitivas que não foram desenvolvidas pelas grades curriculares dos cursos de ensino fundamental e médio. O estudo em [9] também apresenta um gráfico evolutivo contendo o percentual de matrículas dos cursos de computação na Universidade da Califórnia de Los Angeles (UCLA), ilustrado na Figura 1. Nela é possível perceber duas crescentes fases de aumento da procura por esses cursos. A primeira fase é registrada entre os anos de 1982 e 1983, e a segunda no início dos anos 2000. Os autores afirmam que a primeira foi motivada pelo advento dos microcomputadores, a partir do início dos anos 1980, e a segunda pelo surgimento da internet. Entretanto, independente das ondas motivadoras, é possível perceber uma forte tendência de queda que se mantém nos demais períodos.

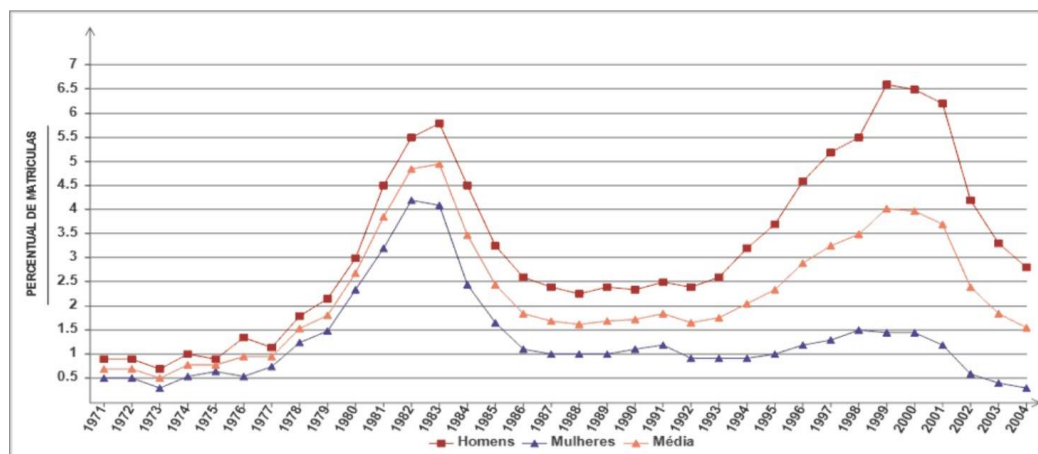


Figura 1: Evolução do percentual de matrículas nos cursos de computação na UCLA por Gênero (1971–2004)

Outro estudo relevante e que apresenta dados sobre a procura por cursos de computação é o relatório Taulbee Survey [13], uma das principais fontes de informações sobre dados estatísticos das áreas de Ciências e Engenharia da Computação na América do Norte. A Figura 2 mostra o gráfico evolutivo com os números de novos alunos graduandos das universidades norte-americanas e confirma a veracidade do aumento de matrículas mencionado em [9] no início dos anos 2000, mostrando a forte tendência por estes cursos na referida época.

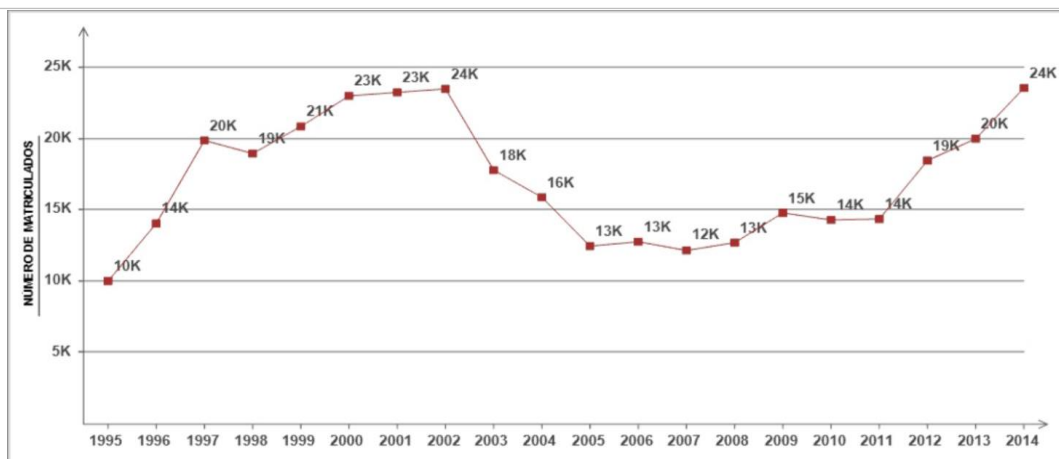


Figura 2: Evolução do número de novos alunos graduandos em cursos de computação nas universidades norte-americanas (1995–2014)

3 Metodologia

Esta pesquisa caracterizou-se como exploratória e bibliográfica, conforme a classificação proposta em [14]. No que diz respeito à pesquisa exploratória, sua aplicação é pertinente ao propósito deste trabalho, que buscou compreender o estado da arte sobre o uso de ferramentas digitais lúdicas voltadas ao ensino de estruturas de repetição em programação de computadores.

Para conduzir o estudo, adotou-se o mapeamento sistemático da literatura, seguindo as diretrizes propostas em [15]. De acordo com os autores, o mapeamento sistemático consiste em uma técnica estruturada que permite identificar padrões, tendências e lacuna em um campo de estudo, oferecendo uma visão abrangente das publicações existentes. Assim, ele se alinha tanto aos objetivos exploratórios quanto ao caráter bibliográfico da pesquisa, funcionando como uma metodologia que conecta ambos os aspectos.

O processo metodológico seguiu três etapas principais: (i) definição das diretrizes do mapeamento, incluindo formulação do objetivo, estrutura PICO e critério de inclusão e exclusão; (ii) construção e execução das estratégias de busca nas bases científicas selecionadas; e (iii) seleção, extração e síntese dos estudos elegíveis. Essas etapas permitiram uma análise das evidências disponíveis, possibilitando não apenas a identificação das ferramentas digitais lúdicas encontradas, mas também a compreensão de suas abordagens e contribuições no ensino de estruturas de repetição.

4 Condução do Mapeamento Sistemático

O mapeamento sistemático foi conduzido com o objetivo de explorar os estudos mais recentes que abordam a definição, aplicação e avaliação de ferramentas digitais lúdicas no apoio ao processo de ensino-aprendizagem do conceito de estruturas de repetição em programação de computadores. Esse método permitiu identificar padrões, classificar contribuições e compreender a extensão das pesquisas desenvolvidas na área, fornecendo uma visão abrangente sobre o estado da arte.

Para orientar o processo de investigação, definiu-se o objetivo do mapeamento e adotou-se a estrutura PICO (População, Intervenção, Comparação e Resultados), que auxilia na formulação das questões e na delimitação da estratégia de busca. A população compreende trabalhos científicos que discutem ferramentas digitais aplicadas ao ensino de estruturas de repetição; a intervenção refere-se ao uso dessas

ferramentas em atividades de aprendizagem envolvendo estruturas de repetição; a comparação considera diferentes abordagens, estratégias didáticas e tipos de ferramentas; e os resultados esperados incluem evidências sobre características, funcionalidades e impactos no processo de ensino-aprendizagem.

Com base nesse objetivo e nessa estrutura, foram desenvolvidas as seguintes questões que norteiam a investigação:

1. Quais as principais dificuldades identificadas no processo de ensino-aprendizagem do conceito computacional de “estruturas de repetição”?
2. Quais as ferramentas digitais lúdicas utilizadas para apoiar o processo de ensino-aprendizagem de “estruturas de repetição”?
3. Quais as principais características dessas ferramentas que as tornam eficientes e como elas foram construídas?

Para o processo de seleção dos estudos, foram definidos os critérios de inclusão e exclusão apresentados no Quadro 1. Esses critérios foram aplicados em cada uma das etapas do processo de seleção, sendo os de inclusão referentes aos critérios específicos do tema da pesquisa, e os de exclusão referentes a outros critérios técnicos.

INCLUSÃO	EXCLUSÃO
CI-1: Apresenta no título e/ou no <i>abstract</i> que se trata de ensino de repetição ou ensino de programação, utilizando estruturas de repetição.	CE-1: Estudos que foram publicados anteriormente ao ano de 2013.
CI-2: Apresenta na introdução e/ou conclusão que o artigo tem como foco o processo de ensino-aprendizagem de estruturas de repetição com uso de ferramentas de software.	CE-2: Estudos que não se caracterizam como artigos científicos e/ou não possuem DOI.
	CE-3: Estudos não disponíveis ou de difícil acesso

Quadro 1: Critérios de inclusão e exclusão

Inicialmente, esta pesquisa considerou seis bases científicas: ACM Digital Library, IEEE Xplore, ScienceDirect, SpringerLink, Scopus e Compendex. No entanto, após uma análise das restrições de acesso, as bases Scopus e Compendex foram desconsideradas, pois atualmente exigem vínculo institucional para a busca e acesso aos artigos. Assim, foram selecionadas as seguintes quatro bases finais e, visando garantir consistência nos resultados, definidas as configurações de busca para cada uma:

- **ACM Digital Library**: em modo “Advanced Search”, com a busca de palavras configuradas para “Anywhere” e o período “All Dates”, no conjunto “The ACM Full-Text Collection”. Foram considerados até os 20 primeiros artigos exibidos na página de resultados.
- **IEEE Xplore**: em modo “Advanced Search”, com a busca definida para “Full Text & Metadata”. Foram analisados até os 25 primeiros artigos exibidos.
- **ScienceDirect**: em modo “Advanced Search”. Após a realização da busca, o “Subject Areas” foi filtrado para “Computer Science”. Foram considerados na análise até os 25 primeiros artigos exibidos.
- **Springer Link**: com os filtros configurados como “content type = article” e “discipline = computer science”, foram analisados até os 20 primeiros artigos exibidos na página de resultados.

De acordo com as questões de pesquisa anteriormente definidas, foram identificados dois subconjuntos de palavras-chave e seus sinônimos, relacionados aos termos centrais da pesquisa: ‘Ensino em Programação’ e ‘Estruturas de repetição’. Esses conjuntos de termos foram definidos para delimitar o alcance da pesquisa e garantir que os principais artigos sobre o tema fossem incluídos. Os subconjuntos definidos são os seguintes:

- "teaching programming" – "teach programming" – "teaching to program" – "teach to program" – "teaching coding" – "teach coding" – "teaching to code" – "teach to code" – "learning programming" – "learn programming" – "learning to program" – "learn to program" – "learning coding" – "learn coding" – "learning to code" – "learn to code";
- "loop structure" – "looping structure" – "for loop" – "while loop" – "repetition structure" – "control flow" – "control structure" – "iteration structure" – "programming loop" – "control repetition" – "looping construct" – "for each loop"

Considerando que algumas bases científicas abrangem uma ampla gama de domínios do conhecimento (como engenharia, matemática, biologia e medicina) e os termos dos subconjuntos destacados acima podem empregar distintas conotações em cada domínio, percebeu-se a importância de realizar uma análise qualitativa cuidadosa para cada palavra-chave e seus sinônimos. Essa etapa buscou delimitar quais termos são mais apropriados para cada uma das bases, de tal forma que os resultados da busca estivessem alinhados tanto ao contexto de ensino de programação quanto ao de estruturas de repetição, evitando que artigos de temas indesejados fossem incluídos na análise.

O Quadro 2 apresenta o resultado da avaliação referente ao termo “*Ensino em programação*”, listando as palavras-chave e seus sinônimos utilizados em cada base científica, o número de artigos retornados por cada termo e a categorização da relevância para a busca. O Quadro 3 mostra os mesmos elementos para o termo “*Estrutura de repetição*”. Em ambos os casos, as categorias de relevância foram definidas conforme descrito a seguir:

- **G** (*good*) – aparentemente a maior parte dos artigos abordam sobre o tema (‘Ensino em programação’ ou ‘Estrutura de repetição’);
- **R** (*regular*) – grande quantidade de artigos que não tratam do tema;
- **B** (*bad*) – a maior parte dos artigos são indesejados.

Sinônimos	ACM		IEEE		ScienceDirect		SpringerLink	
	Quantidade	Qualidade	Quantidade	Qualidade	Quantidade	Qualidade	Quantidade	Qualidade
“teaching programming”	1947	G	151589	G	584	G	4790	G
“teach programming”	1158	G	151589	G	186	G	2113	G
“teaching to program”	7	G	151364	G	394	R	6636	R
“teach to program”	2	G	151364	G	206	G	2867	R
“teaching coding”	133	G	85640	B	11	B	2943	G
“teach coding”	94	R	85640	B	23	B	1376	R
“teaching to code”	2	B	85607	B	21	B	4743	B
“teach to code”	1	B	85607	B	16	B	2074	B
“learning programming”	1877	G	633474	G	1482	R	26501	G
“learn programming”	1232	R	633474	G	207	B	15212	G
“learning to program”	1797	G	631718	G	1133	R	26234	R
“learn to program”	852	G	631718	G	487	R	16008	B
“learning coding”	104	R	522627	B	208	B	15354	B

“learn coding”	140	R	522627	B	102	B	9781	R
“learning to code”	370	G	522103	B	608	B	34384	B
“learn to code”	242	R	522103	B	417	B	21005	B

Quadro 2: Matriz avaliativa das palavras-chave sobre ‘Ensino em programação’

Sinônimos	ACM		IEEE		ScienceDirect		SpringerLink	
	Quantidade	Qualidade	Quantidade	Qualidade	Quantidade	Qualidade	Quantidade	Qualidade
“loop structure”	1639	B	12233	B	1712	B	413	B
“looping structure”	158	B	216	R	1712	R	19	R
“for loop”	9395	G	21305	B	118851	R	2315	B
“while loop”	6943	G	11312	G	118851	R	1993	B
“repetition structure”	50	B	232	B	588	B	21	B
“control flow”	25937	B	48801	B	9683	B	3210	B
“control structure”	4776	B	62214	B	8284	R	1258	R
“iteration structure”	62	B	410	R	463	B	28	B
“programming loop”	35	R	116	R	645	R	7	R
“control repetition”	15	R	49	B	50	B	2	B
“looping construct”	196	B	190	R	695	R	19	B
“for each loop”	1405	G	3778	B	118851	B	252	B
“loop structure”	1639	B	12233	B	1712	B	413	B
“looping structure”	158	B	216	R	1712	R	19	R
“for loop”	9395	G	21305	B	118851	R	2315	B
“while loop”	6943	G	11312	G	118851	R	1993	B

Quadro 3: Matriz avaliativa das palavras-chave sobre ‘Estrutura de repetição’

A partir desse processo, foi possível construir uma string de busca otimizada para cada uma das bases científicas. Foram aplicados os operadores lógicos AND e OR, sendo o AND (de caráter exclusivo) utilizado para combinar os principais termos da pesquisa, e o OR (de caráter inclusivo) utilizado para separar os sinônimos de cada termo. O procedimento consistiu em unir as palavras-chave avaliadas como **G** por meio do conectivo lógico OR para considerar trabalhos relevantes acerca dos temas centrais desta pesquisa. No caso em que nenhum termo foi classificado como **G**, foram considerados aqueles classificados como **R**. As *strings* de busca resultantes deste procedimento, específicas para cada base, foram as seguintes:

- **ACM Digital Library:** ("teaching programming" OR "teach programming" OR "teaching to program" OR "teach to program" OR "teaching coding" OR "learning programming" OR "learning to program" OR "learn to program" OR "learning to code") AND ("for loop" OR "while loop" OR "for each loop");

- **IEEE Xplore:** ("teaching programming" OR "teach programming" OR "teaching to program" OR "teach to program" OR "learning programming" OR "learn programming" OR "learning to program" OR "learn to program") AND ("while loop");
- **ScienceDirect:** ("teaching programming" OR "teach programming" OR "teach to program") AND ("looping structure" OR "for loop" OR "while loop" OR "control structure" OR "programming loop" OR "looping construct");
- **SpringerLink:** ("teaching programming" OR "teach programming" OR "teaching coding" OR "learning programming" OR "learn programming") AND ("looping structure" OR "control structure" OR "programming loop").

Durante o processo de mapeamento sistemático, foi utilizada a ferramenta digital *Rayyan* [16] para a catalogação e organização durante a triagem inicial dos estudos selecionados nas bases científicas. O Quadro 4 ilustra o resumo do processo.

ETAPA	ATIVIDADE
Aplicação das Strings nas fontes de busca	Catalogação dos resultados na ferramenta Rayyan
Processo de seleção inicial	Leitura do título, abstract e palavras-chave.
	Aplicação dos critérios CI-1, CE-1 e CE-2.
Processo de seleção refinada	Leitura da introdução e conclusão
	Aplicação do critério CI-2 e CE-3
Revisão aprofundada	Leitura completa dos artigos.
	Sumarização e conclusão

Quadro 4: Etapas do processo do mapeamento

O processo de seleção inicial consistiu na aplicação das strings em cada uma de suas respectivas fontes de buscas. As informações básicas dos estudos recuperados (título, abstract, autor, data de publicação, dentre outros) foram exportadas das bases de dados e, então, importadas para a ferramenta Rayyan. O Quadro 5 apresenta a quantidade de estudos recuperados em cada fonte após a aplicação das strings.

Fonte de busca	Resultados obtidos
ACM	423
IEEE	144
ScienceDirect	126
SpringerLink	12
Total	705

Quadro 5: Resultados da aplicação das strings nas fontes de busca

As informações essenciais dos artigos que foram importadas para o Rayyan já contemplavam o critério de exclusão CE-1, pois, ao aplicar o filtro de data, foi possível (em todas as bases utilizadas) garantir que fossem retornados apenas os estudos mais recentes. Dentro da ferramenta, cada estudo passou por uma análise seguindo o critério de inclusão CI-1, que se resume à leitura do título e do abstract, e o critério de exclusão CE-2, que trata dos estudos que não possuem DOI e nem são artigos científicos.

Feita a leitura, a cada estudo foi atribuído um rótulo indicando que já havia sido avaliado e, em seguida, registrava-se a decisão de inclusão ou exclusão. Para um registro detalhado, o Quadro 6 evidencia o

número de artigos que foram incluídos e excluídos (conforme os critérios CI-1 e CE-2), por base científica.

Fonte de busca	Artigos recuperados	Artigos incluídos	Artigos excluídos
ACM	423	11	412
IEEE	144	9	135
ScienceDirect	126	3	123
SpringerLink	12	1	11
Total	705	24	681

Quadro 6: Total de artigos incluídos e excluídos por base científica

Os 24 artigos incluídos foram selecionados e catalogados para o processo de seleção refinada. A primeira atividade desta nova etapa consistiu na busca e obtenção das versões completas dos artigos, que foram então submetidas à ferramenta Rayyan, a fim de organizar os documentos e facilitar sua triagem. Durante esse processo, foi aplicado o critério de exclusão CE-3, responsável por eliminar os estudos indisponíveis ou de difícil acesso, inviabilizando a leitura das seções necessárias para análise.

Uma vez catalogados todos os artigos, foi possível prosseguir com a análise individual dos estudos, seguindo o critério de inclusão CI-2. Esse critério consistiu na leitura da introdução e conclusão, com o objetivo de verificar se o foco central da pesquisa está relacionado ao processo de ensino-aprendizagem de estruturas de repetição em programação de computadores, desde que articulado ao uso de ferramentas de software.

5 Resultados do Mapeamento

Como resultado, cinco estudos foram selecionados por atenderem explicitamente aos critérios estabelecidos, sendo encaminhados para a etapa de revisão aprofundada, na qual foram examinados com maior nível de detalhamento. O objetivo consistiu em realizar a leitura integral dos artigos e analisar as principais características, abordagens metodológicas e contribuições relevantes para responder às questões de pesquisa do trabalho. O Quadro 7 apresenta os estudos considerados nesta etapa de revisão, organizados com informações essenciais para contextualização: título do artigo, base de publicação, referência e o público-alvo abordado em cada proposta.

REFERÊNCIA	TÍTULO	BASE DE PUBLICAÇÃO	PÚBLICO-ALVO
Gomes, Falcão e Tedesco (2018) [17]	Exploring an approach based on digital games for teaching programming concepts to young children	ScienceDirect	Educação Infantil
Mladenović, Boljat e Žanko (2017) [18]	Comparing loops misconceptions in block-based and text-based programming languages at the K-12 level	SpringerLink	Ensino médio
Frison (2015) [19]	A Teaching Assistant for Algorithm Construction	ACM	Ensino Superior
Touretzky et al. (2013) [20]	Accelerating K-12 Computational Thinking Using Scaffolding, Staging, and Abstraction	ACM	Ensino fundamental

Zhang et al. (2013) [21]	Visualizing Loops Using a Game-Like Instructional Module	IEEE	Ensino Superior
-----------------------------	--	------	-----------------

Quadro 7: Artigos selecionados para revisão aprofundada

Os cinco artigos selecionados para esta etapa apresentam uma distribuição temporal entre os anos de 2013 e 2018. Dois estudos datam de 2013, enquanto os demais foram publicados em 2015, 2017 e 2018, mostrando que há um interesse consistente, embora pontual, na investigação de ferramentas digitais lúdicas no ensino de estruturas de repetição. Vale ressaltar que, conforme salientado no CE-1 (Quadro 1), os estudos anteriores ao ano de 2013 não foram considerados neste mapeamento.

No que diz respeito às bases científicas de publicação, percebe-se que em todas as que foram investigadas ao longo deste mapeamento, houve ao menos a presença de um artigo que explora a temática central desta pesquisa. Foram encontrados dois estudos na ACM Digital Library, um na IEEE Xplore, um na SpringerLink e outro na ScienceDirect. Essa distribuição entre as fontes de busca corrobora o alcance da estratégia de busca adotada e contribui para a relevância dos estudos analisados, uma vez que essas bases reúnem parte significativa da produção científica qualificada sobre o ensino em computação e tecnologias educacionais.

Quanto ao público-alvo dos estudos analisados, observa-se uma distribuição equilibrada entre diferentes níveis de ensino. Dois dos artigos concentram-se no ensino superior, enquanto os demais abordam contextos da educação básica. Dentre estes, um estudo foi aplicado na educação infantil, um no ensino fundamental e outro no ensino médio. Essa variedade de faixas etárias indica que as ferramentas digitais lúdicas têm sido exploradas como recurso pedagógico tanto nos estágios iniciais quanto em fases mais avançadas da formação em programação, com adaptações conforme o perfil dos estudantes e os objetivos educacionais.

A partir da análise dos artigos supracitados, foi possível identificar um conjunto de 13 ferramentas digitais com propostas lúdicas, utilizadas como estratégias de apoio pedagógico ao ensino em programação, incluindo as estruturas de repetição. O Quadro 8 apresenta um resumo dessas ferramentas, indicando, para cada uma delas, o nome, o tipo de linguagem utilizada, a abordagem lúdica (como jogos, simulações ou desafios) e se o foco da ferramenta é exclusivo no conceito de repetição.

Para fins de análise comparativa, as linguagens adotadas pelas ferramentas foram classificadas em três categorias principais: Visual, Textual e Híbrida. A seguir, a descrição de cada categoria:

- A categoria “Visual” reúne ferramentas que utilizam uma interface visual onde o usuário manipula elementos gráficos, como blocos, ícones ou comandos visuais, por meio de ações como arrastar e soltar ou clicar em botões.
- A categoria “Textual” refere-se a ferramentas que utilizam linguagens de programação tradicionais, em que o código é escrito por meio de texto.
- A categoria “Híbrida” engloba ferramentas que combinam a manipulação visual com elementos de código textual ou pseudocódigo.

FERRAMENTA	TIPO DE LINGUAGEM	ABORDAGEM LÚDICA	FOCO EXCLUSIVO EM REPETIÇÃO	REFERÊNCIA
AlgoTouch	Híbrida	Simulador de algoritmos interativo	Não	Frison (2015) [19]
Microsoft Kodu	Visual	Criação de jogos	Não	Touretzky et al. (2013) [20]
Alice	Híbrida	Simulador 3D com animações	Não	Touretzky et al. (2013) [20]

Scratch	Visual	Plataforma educacional baseado em blocos	Não	Mladenović, Boljat e Žanko (2017) [18]
Logo	Textual	Simulador gráfico	Não	Mladenović, Boljat e Žanko (2017) [18]
Módulo Turtle do Python	Textual	Simulador gráfico	Não	Mladenović, Boljat e Žanko (2017) [18]
Code Baymax	Visual	Jogo com resolução de desafios	Não	Gomes, Falcão e Tedesco(2018) [17]
Kodable	Visual	Jogo com resolução de desafios	Não	Gomes, Falcão e Tedesco(2018) [17]
Code Studio	Visual	Plataforma educacional baseado em blocos	Não	Gomes, Falcão e Tedesco(2018) [17]
LEGO Bits and Bricks	Visual	Jogo com resolução de desafios	Não	Gomes, Falcão e Tedesco(2018) [17]
Lightbot	Visual	Jogo com resolução de desafios	Não	Gomes, Falcão e Tedesco(2018) [17]
The Foos	Visual	Jogo com resolução de desafios	Não	Gomes, Falcão e Tedesco(2018) [17]
Iterative Dungeon	Híbrida	Jogo com resolução de desafios	Sim	Zhang et al. (2013) [21]

Quadro 8: Ferramentas encontradas na literatura

Nota-se uma diversidade significativa nas abordagens lúdicas empregadas, oferecendo diferentes formas de engajar os alunos. Algumas utilizam jogos de aventura, outras simuladores ou atividades interativas, o que evidencia o potencial didático dessas ferramentas de enriquecer a aprendizagem e facilitar a compreensão dos conceitos de programação, diante das variadas formas de aplicação.

A maioria das ferramentas identificadas se apoia em linguagens visuais, com a manipulação de blocos ou ícones, o que está alinhado ao público-alvo predominante dos estudos, geralmente em estágios iniciais de aprendizagem em computação. É interessante observar ainda que apenas uma das ferramentas teve como foco exclusivo o ensino de estruturas de repetição, enquanto as demais abordam o conceito dentro de um contexto mais amplo de fundamentos da programação.

Essas observações iniciais forneceram uma visão geral do conjunto de ferramentas identificadas e como os estudos as propuseram. Na próxima seção, é apresentada a avaliação de cada uma das ferramentas selecionadas, bem como a comparação entre elas.

6 Discussão

Com base nos dados extraídos nos resultados do mapeamento, foi conduzida uma análise comparativa das ferramentas digitais lúdicas encontradas, considerando quatro critérios centrais:

1. contexto de aplicação;
2. resultados observados;
3. limitações identificadas;
4. foco no ensino de estruturas de repetição.

Esses critérios foram definidos com base nas características observadas durante o mapeamento, permitindo compreender como as ferramentas são utilizadas e como se comportam em diferentes situações de ensino-aprendizagem.

Ao analisar esses critérios, buscou-se compreender como as ferramentas foram utilizadas em diferentes ambientes, quais benefícios e dificuldades foram relatados pelos autores, quais limitações aparecem com maior frequência e até que ponto elas tratam diretamente o conceito de estruturas de repetição. A discussão apresentada a seguir segue essa organização, aprofundando cada critério e destacando os padrões encontrados.

O primeiro critério analisado é o contexto de aplicação, que diz respeito ao ambiente em que cada ferramenta foi empregada, ao perfil dos estudantes participantes e às condições de uso (com ou sem mediação docente, em atividades formais ou informais, em atividades extensas ou oficinas rápidas). Esse critério foi importante, pois permitiu compreender em que medida o ambiente educacional influenciou a eficácia da ferramenta e como ela se adaptou aos diferentes níveis de experiência dos alunos.

Nos estudos revisados, observou-se que a maioria das ferramentas foi aplicada em turmas de iniciantes, incluindo estudantes do ensino médio e dos primeiros semestres do ensino superior, o que está alinhado ao propósito de apoiar a aprendizagem de conceitos fundamentais. Outro aspecto relevante é que quase todas as intervenções ocorreram com acompanhamento docente, sugerindo que as ferramentas ainda dependem dessa mediação, especialmente quando envolvem interpretação de resultados ou tomada de decisões durante a resolução das atividades.

O Quadro 9 apresenta a sistematização das informações referentes ao primeiro critério analisado, e que resume o contexto de aplicação das ferramentas digitais lúdicas analisadas nos estudos selecionados. São destacados aspectos como o público-alvo, o ambiente educacional, a duração das intervenções, a modalidade de ensino e a presença de mediação docente.

FERRAMENTA	PÚBLICO-ALVO	CONTEXTO EDUCACIONAL	DURAÇÃO E MODALIDADE	MEDIAÇÃO DOCENTE
Iterative Dungeon	Ensino Superior	Curso formal	1 hora - Presencial	Sim
Scratch/ Logo / Python (Turtle)	Alunos de 5º e 6º anos	Aulas regulares	3 semanas - Presencial	Sim
AlgoTouch	Professores/Alunos	Seminários	Curta - Presencial	Não aplicado
Kodu / Alice	10 a 17 anos	Acampamento intensivo	5 dias - Presencial	Sim
Code Baymax e os demais	5 a 7 anos	Aulas extracurriculares	9 semanas - Presencial	Sim

Quadro 9: Contexto de aplicação das ferramentas

O segundo critério corresponde aos resultados observados, que envolve a análise dos efeitos reportados pelas pesquisas, incluindo impacto na aprendizagem, motivação, clareza conceitual e engajamento dos estudantes. Esse critério abrange especificamente aquilo que os estudos demonstram como consequência do uso de cada ferramenta, seja por meio de avaliações quantitativas ou por observações qualitativas.

Os resultados descritos nos estudos apontam melhorias na compreensão de conceitos básicos de programação, especialmente quando as ferramentas utilizam visualização animada ou representação gráfica do fluxo de execução. Estudantes relataram maior facilidade em entender como o programa progride e como as variáveis mudam ao longo das atividades.

Contudo, as avaliações quantitativas observadas nos estudos ainda são escassas, e grande parte das conclusões provém de percepções dos próprios alunos ou de análises exploratórias dos autores. Alguns estudos indicaram aumento de motivação e engajamento, reforçado pelos elementos lúdicos presentes nas ferramentas, como desafios, recompensas e narrativas. O Quadro 10 sintetiza o segundo critério e evidencia o papel das ferramentas no processo de aprendizagem.

FERRAMENTA	TIPO DE EVIDÊNCIA	DESTAQUES
Iterative Dungeon	Quantitativa e Qualitativa	Aumento de 10% na média de desempenho (pré-teste/pós-teste); engajamento elevado; compreensão aprimorada dos loops com visualização passo a passo
Scratch	Quantitativa	Maior desempenho entre alunos de baixa habilidade; compreensão facilitada por blocos visuais
Logo e Python (Turtle)	Quantitativa	Desempenho inferior aos blocos visuais; dificuldades com sintaxe textual, indentação e abstração
Alice	Qualitativa	Ajuda na transição entre laços implícitos e explícitos; comandos visuais reforçam entendimento da estrutura dos loops
Kodu	Qualitativa	Favorece compreensão da lógica de repetição por meio da mecânica do jogo; foco em laços implícitos e alto engajamento
AlgoTouch	Qualitativa	Avaliada como útil por docentes; destaque para interatividade e visualização de algoritmos com laços
Code Baymax	Qualitativa	Comandos visuais simples facilitaram o entendimento inicial do conceito de repetição
Lightbot, Code Studio e Kodable	Qualitativa	Estudantes reconheceram a necessidade de usar repetição, mas apresentaram grande dificuldade na execução correta dos comandos de loop, mesmo com mediação docente
Lego Bits and Bricks e The Foos	Qualitativa	Estudantes não identificaram espontaneamente a presença de repetição nas atividades; aplicação do conceito só ocorreu com intervenção do professor

Quadro 10: Resumo dos resultados observados

O terceiro critério analisado trata das limitações identificadas, que inclui tanto limitações metodológicas dos estudos quanto restrições próprias das ferramentas. Esse critério buscou evidenciar fragilidades que impactam a validade dos resultados e a adoção prática das soluções analisadas. Entre as limitações mais recorrentes destacaram-se o tamanho reduzido das amostras, a curta duração das intervenções e a falta de comparações com grupos de controle, o que dificultou a generalização dos resultados.

Do ponto de vista técnico, algumas ferramentas apresentaram limitações de usabilidade, pouca documentação ou dependência de plataformas específicas, restringindo seu uso em diferentes contextos. Além disso, muitos estudos reconhecem a necessidade de acompanhamento docente frequente, o que pode ser um obstáculo para sua adoção em cenários que buscam uma maior autonomia por parte do estudante.

O Quadro 11 sintetiza o terceiro critério, referente às limitações identificadas, e reúne esses aspectos de uma forma que pode-se visualizar padrões que se repetem em diferentes pesquisas.

TIPO DE LIMITAÇÃO	IMPACTO NO ESTUDO	QUANTIDADE DE ESTUDOS
Ausência ou limitação de avaliações quantitativas sistemáticas	Dificulta mensurar objetivamente os ganhos de aprendizagem e comparar resultados entre estudos	3
Curta duração das intervenções	Limita a avaliação quanto a retenção ou transferência de conhecimento	2
Ausência de grupos de controle e validação metodológica robusta	Reduz a validade interna dos estudos e a confiabilidade das conclusões	5
Uso simultâneo de múltiplas ferramentas e conceitos	Dificulta isolar o impacto de cada ferramenta no aprendizado de repetição	1
Limitações técnicas das tecnologias utilizadas	Restringem a complexidade das atividades, afetam acessibilidade e podem reduzir a motivação	1
Falta de descrição ou análise da mediação docente	Dificulta compreender a influência do professor no processo de aprendizagem	3

Quadro 11: Principais limitações e seus impactos nos estudos

O quarto critério avaliado é o foco no ensino de estruturas de repetição, que analisa se a ferramenta foi projetada especificamente para ensinar loops ou se aborda o tema apenas como parte de conteúdos mais amplos. Esse critério é central para esta pesquisa, uma vez que o objetivo do estudo é compreender de que forma ferramentas lúdicas auxiliam na aprendizagem do ensino de estruturas de repetição.

Os resultados revelaram que apenas um dos estudos [21] descreve uma ferramenta criada explicitamente para tratar estruturas de repetição (*Iterative Dungeon*), enquanto os demais abordam o tema de forma indireta. Em muitas ferramentas, o conceito de estruturas de repetição é explorado apenas conforme surgem durante as atividades, sem mecanismos específicos para sua visualização ou análise. Ainda assim, elementos como animações, simulações passo a passo e representações visuais contribuem para a compreensão do fluxo repetitivo, mesmo quando esse não é o foco central.

O Quadro 12 sintetiza o quarto critério, referente ao foco no ensino de estruturas de repetição, e apresenta como cada estudo aborda a temática, evidenciando a lacuna existente na literatura.

FERRAMENTA	ABORDAGEM SOBRE REPETIÇÃO
Iterative Dungeon	Níveis progressivos com aplicação direta de ‘for’, ‘nested for’ e ‘while’; visualização animada e prática em Java
Scratch	Blocos visuais claros de repetição; favorece iniciantes
Logo e Python (Turtle)	Permite laços, mas exige maior abstração
Alice	Comandos explícitos (“Loop n times”, “While loop”); mapeia inicialização, condição e corpo
Kodu	Explora loops implícitos e explícitos; diferencia comportamento contínuo e controlado por estados
Code Baymax	Repetição visual pré-definida; facilita compreensão inicial
Lightbot	Usa recursividade como repetição; alto nível de abstração
Code Studio	Blocos de loop explícitos; aplicação inicial guiada

The Foos, Kodable, Lego Bits and Bricks	Repetição implícita/indireta; sem instrução formal de loops, presente como parte do enredo ou de sequências pré-definidas
AlgoTouch	Repetição como parte da visualização de algoritmos

Quadro 12: Forma de abordagem do conceito de repetição pelas ferramentas analisadas

Por fim, ao integrar os resultados dos quatro critérios analisados, foi possível identificar um conjunto de padrões que ajudam a caracterizar o que torna algumas das ferramentas estudadas mais eficazes do que outras. Essa integração permitiu observar elementos que aparecem repetidamente nas soluções analisadas e que, juntos, apontam caminhos concretos para o desenvolvimento de novas tecnologias educacionais voltadas ao ensino de estruturas de repetição.

A partir dessa síntese, foi possível reunir um conjunto estruturado de requisitos essenciais que uma ferramenta digital lúdica deveria possuir para apoiar de forma consistente a aprendizagem de estruturas de repetição. Esses requisitos foram organizados e apresentados no Quadro 13, que reúne as características identificadas como mais relevantes e recorrentes nos estudos analisados. A discussão detalhada desses requisitos, bem como sua relação com as questões de pesquisa, é retomada na seção de conclusões.

CATEGORIA	REQUISITO	JUSTIFICATIVA
Conteúdo e Progressão Didática	Cobrir diferentes tipos de loops (for, while, nested) de forma progressiva	A abordagem gradual do Iterative Dungeon favoreceu a compreensão; em outras ferramentas, a ausência de progressão limitou o aprofundamento conceitual.
Visualização e Interatividade	Visualização animada da execução dos loops	Ferramentas como Scratch mostraram que recursos visuais ajudam na compreensão; a ausência desse recurso dificultou o entendimento em ambientes mais abstratos.
Suporte Pedagógico	Possibilidade de uso com ou sem mediação docente	Muitas ferramentas falharam sem intervenção ativa (ex.: Code Studio); autonomia é importante para ampliar contextos de aplicação.
Feedback	Fornecer retorno imediato e claro sobre erros e acertos	Poucas ferramentas ofereceram feedback instrucional, mas quando presente, aumentou a aprendizagem ativa e o engajamento.
Motivação e Engajamento	Equilibrar elementos lúdicos e narrativos com clareza conceitual	Narrativas envolventes (como em The Foos) engajaram, mas às vezes desviaram o foco do conceito de repetição.
Flexibilidade de Aplicação	Permitir ajustes de dificuldade e abordagem conforme idade e nível de experiência do estudante	Ferramentas como Lightbot e Code Studio apresentaram desafios que estavam acima da capacidade do público-alvo, resultando em baixa autonomia dos alunos.

Quadro 13: Requisitos para uma ferramenta digital lúdica ideal voltada ao ensino de estruturas de repetição

7 Conclusões

Este estudo teve como objetivo investigar como ferramentas digitais lúdicas têm sido utilizadas para apoiar o ensino do conceito de estruturas de repetição em programação de computadores. A partir do mapeamento sistemático realizado e da análise aprofundada dos estudos selecionados, foi possível

identificar tendências relevantes na literatura, compreender desafios persistentes no ensino de loops e sintetizar elementos essenciais para o desenvolvimento de ferramentas mais eficazes.

Em relação à primeira questão do mapeamento: “Quais as principais dificuldades identificadas no processo de ensino-aprendizagem de estruturas de repetição”, os resultados revelam que estudantes iniciantes enfrentam obstáculos significativos para compreender o funcionamento de laços iterativos. Dificuldades relacionadas ao fluxo de execução, ao rastreamento de variáveis e à construção de modelos mentais adequados foram identificadas em diferentes contextos educacionais e níveis de ensino. Estudos como [17] e [18] mostram que essas dificuldades se intensificam quando o conteúdo é apresentado de forma abstrata ou sem apoio visual, reforçando a importância de recursos que tornem o processo mais concreto e observável.

Quanto à segunda questão: “Quais ferramentas digitais lúdicas têm sido utilizadas para apoiar esse processo”, identificaram-se treze ferramentas distribuídas entre os cinco estudos selecionados, incluindo ambientes mais conhecidos, como Scratch, Alice, Kodu e Code Baymax, além de propostas mais específicas, como Iterative Dungeon. Apesar da variedade, apenas a ferramenta Iterative Dungeon foi projetada com foco direto no ensino de estruturas de repetição, enquanto as demais tratam o tema de maneira complementar em atividades mais amplas de lógica ou programação introdutória. Essa lacuna evidencia a necessidade de tecnologias que abordem o conceito de forma mais direcionada e sistemática.

A terceira questão “Quais características tornam essas ferramentas eficientes e como foram construídas” foi respondida a partir da avaliação feita na seção anterior. Os estudos mostram que ferramentas que oferecem visualização passo a passo, recursos interativos, feedback imediato e progressão gradual das atividades tendem a favorecer o entendimento conceitual e o engajamento dos alunos. Narrativas visuais (como em Alice e Kodu), blocos estruturados (como em Scratch) e simulações detalhadas do fluxo de execução (como em Iterative Dungeon) demonstraram contribuir para a redução da carga cognitiva e para a compreensão mais clara do comportamento dos loops.

Além das contribuições, também se observaram limitações metodológicas e tecnológicas recorrentes nos estudos analisados, como amostras pequenas, intervenções de curta duração, ausência de grupos de controle, predominância de análises qualitativas e dependência de mediação docente. Tais fragilidades restringem a generalização dos resultados e indicam a necessidade de avaliações mais rigorosas e aprofundadas em pesquisas futuras.

No que diz respeito à questão central desta pesquisa: “Quais os principais requisitos para o desenvolvimento de uma ferramenta lúdica de ensino-aprendizagem para o conceito computacional de estruturas de repetição”, a integração dos achados permitiu identificar um conjunto sólido de requisitos essenciais. Esses requisitos, apresentados no Quadro 13, envolvem aspectos relacionados à organização do conteúdo e à progressão didática, à forma como a visualização e a interatividade são estruturadas, ao tipo de suporte pedagógico oferecido, à qualidade do feedback fornecido ao estudante, à capacidade de promover motivação e engajamento e à flexibilidade de aplicação em diferentes contextos e níveis de experiência.

Em síntese, as evidências indicam que uma ferramenta realmente efetiva deve integrar esses elementos de maneira equilibrada, permitindo que o estudante compreenda gradualmente o funcionamento das estruturas de repetição, explore seu comportamento de forma ativa e tenha acesso a recursos que tornem o processo iterativo mais evidente.

Diante dos resultados obtidos, conclui-se que ferramentas digitais lúdicas possuem grande potencial para apoiar o ensino de estruturas de repetição, sobretudo quando aliadas a práticas pedagógicas bem planejadas e quando incorporam mecanismos que tornem visíveis os processos internos da execução iterativa. Entretanto, há ainda um espaço amplo para a criação de soluções voltadas exclusivamente para esse conceito, bem como para pesquisas com maior rigor metodológico.

Espera-se que este estudo contribua para a reflexão crítica sobre o uso dessas ferramentas no ensino de programação e inspire novas investigações, recursos educacionais e práticas docentes mais alinhadas às necessidades reais dos estudantes iniciantes.

Referências

- [1] R. Covington e L. Benegas, A cognitive based approach for teaching programming to computer science and engineering students, em ASEE Annual Conference, Portland, pp. –, ASEE, Washington, DC, 2005. DOI: 10.18260/1-2--14910.
- [2] V. G. Renumol, D. Janakiram e S. Jayaprakash, Identification of cognitive processes of effective and ineffective students during computer programming, ACM Trans. Comput. Educ., 10(3) (2010) 1–21. DOI: 10.1145/1821996.1821998.
- [3] M. Eagle e T. Barnes, Wu's castle: teaching arrays and loops in a game, em ACM Annual Conference on Innovation and Technology in Computer Science Education, 13., Madrid, pp. 245–249, ACM, New York, 2008. DOI: 10.1145/1384271.1384337.
- [4] N. B. Dale, Most difficult topics in CS1: results of an online survey of educators, Inroads – The SIGCSE Bulletin, 38(2) (2006) 49–53. DOI: 10.1145/1138403.1138432.
- [5] R. Kadar, N. A. Wahab, J. Othman, M. Shamsuddin e S. B. Mahlan, A study of difficulties in teaching and learning programming: a systematic literature review, Int. J. Acad. Res. Prog. Educ. Dev., 10(3) (2021) 591–605. DOI: 10.6007/IJARPED/v10-i3/11100.
- [6] C. S. Cheah, Factors contributing to the difficulties in teaching and learning of computer programming: a literature review, Contemp. Educ. Technol., 12(2) (2020) 1–18. DOI: 10.30935/cedtech/8247.
- [7] R. Ziatdinov e J. R. Valles Jr., Synthesis of modeling, visualization, and programming in GeoGebra as an effective approach for teaching and learning STEM topics, Mathematics, 10(3) (2022) 1–17. DOI: 10.3390/math10030398.
- [8] A. P. Ambrósio, F. M. Costa, L. Almeida, A. Franco e J. Macedo, Identifying cognitive abilities to improve CS1 outcome, em Frontiers in Education Conference (FIE), Rapid City, SD, pp. F3G-1–F3G-7, IEEE, 2011.
- [9] C. C. Hernandez, L. Silva, R. A. Segura, J. Schimiguel, M. F. Ledon, L. N. Bezerra, I. F. Silveira, Teaching programming principles through a game engine, Sciences – New York, (2010) 1–8.
- [10] M. E. Caspersen e J. Bennedsen, Instructional design of a programming course, em International Computing Education Research (ICER '07), Atlanta, GA, pp. 111–122, ACM, 2007.
- [11] S. Alaoutinen e K. Smolander, Student self-assessment in a programming course using Bloom's revised taxonomy, em 15th ACM Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE '10), Ankara, Turkey, pp. 155–159, ACM, 2010.
- [12] S. Cooper, Y. J. Nam e L. Si, Initial results of using an intelligent tutoring system with Alice, em 17th ACM Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE '12), Haifa, Israel, pp. 138–143, ACM, 2012.
- [13] Computing Resource Association, Taulbee Survey. Disponível em: <http://cra.org/resources/taulbee-survey>. Acesso em: 12 ago. 2024.

-
- [14] A. C. Gil, Como elaborar projetos de pesquisa, Atlas, São Paulo, 2002.
- [15] K. Petersen, S. Vakkalanka e L. Kuzniarz, Guidelines for conducting systematic mapping studies in software engineering: an update, *Inf. Softw. Technol.*, 64 (2015) 1–18. DOI: 10.1016/j.infsof.2015.03.007.
- [16] M. Ouzzani, H. Hammady, Z. Fedorowicz e A. Elmagarmid, Rayyan — a web and mobile app for systematic reviews, *Systematic Reviews*, 5(210) (2016). DOI: 10.1186/s13643-016-0384-4.
- [17] T. C. S. Gomes, T. P. Falcão e P. C. A. R. Tedesco, Exploring an approach based on digital games for teaching programming concepts to young children, *Int. J. Child-Comput. Interact.*, 16 (2018) 77–84. DOI: 10.1016/j.ijcci.2017.12.005.
- [18] M. Mladenović, I. Boljat e Ž. Žanko, Comparing loops misconceptions in block-based and text-based programming languages at the K-12 level, *Educ. Inf. Technol.*, 23(4) (2017) 1483–1500. DOI: 10.1007/s10639-017-9673-3.
- [19] P. Frison, A teaching assistant for algorithm construction, em *ACM Conference on Innovation and Technology in Computer Science Education*, Vilnius, pp. 9–14, ACM, New York, 2015. DOI: 10.1145/2729094.2742588.
- [20] D. S. Touretzky, D. Marghitu, S. Ludi, D. Bernstein e L. Ni, Accelerating K-12 computational thinking using scaffolding, staging, and abstraction, em *ACM Technical Symposium on Computer Science Education*, 44., Denver, pp. 609–614, ACM, New York, 2013. DOI: 10.1145/2445196.2445374.
- [21] J. Zhang, M. Atay, E. R. Caldwell e E. J. Jones, Visualizing loops using a game-like instructional module, em *IEEE International Conference on Advanced Learning Technologies*, 13., Pequim, pp. 448–450, IEEE, 2013. DOI: 10.1109/ICALT.2013.137.