

UMA ANÁLISE DA APLICAÇÃO DE JOGOS PARA ENSINO DE PROGRAMAÇÃO BÁSICA A PARTIR DE UM MAPEAMENTO SISTEMÁTICO DA LITERATURA

 <https://doi.org/10.56238/arev7n1-176>

Data de submissão: 22/12/2024

Data de publicação: 22/01/2025

Alexsandro Monteiro Nascimento Callins

Graduando em Ciência da Computação
Universidade Federal do Amapá
E-mail: alexsandromnc@gmail.com

Davi Alberto Correa do Carmo

Graduando em Ciência da Computação
Universidade Federal do Amapá
E-mail: dcarmo200010.dd@gmail.com

Julio Cezar Costa Furtado

Doutor em Ciência da Computação
Universidade Federal do Amapá
E-mail: furtado@unifap.br

RESUMO

Este artigo realiza um mapeamento sistemático sobre o uso de jogos digitais no ensino de programação, analisando publicações de 2013 a 2024. O estudo busca identificar quais jogos são utilizados para ensinar conceitos de programação e em quais áreas esses jogos têm impacto educacional. Utilizando critérios rigorosos de inclusão e exclusão, a pesquisa selecionou artigos em bases como ACM Digital Library e IEEE Xplore. O mapeamento identificou 49 jogos voltados para o ensino de programação, com predominância de gêneros como indie, educativos e de simulação, disponíveis em plataformas como Windows, MacOS, Linux e web. As áreas de conhecimento cobertas pelos jogos incluem lógica de programação, manipulação de dados, sintaxe, funções, e debugging, promovendo o desenvolvimento dessas habilidades em um ambiente motivador. Os resultados sugerem que os jogos digitais são ferramentas eficazes para engajar estudantes e facilitar o aprendizado de conceitos abstratos em programação. Oferecendo prática interativa e feedback imediato, esses jogos complementam métodos de ensino tradicionais, tornando o aprendizado mais acessível e dinâmico. A pesquisa conclui que a integração de jogos no ensino de programação pode aumentar o interesse dos alunos e apoiar a assimilação de conteúdos complexos. Além disso, recomenda-se a continuidade de estudos sobre o impacto de jogos em habilidades de longo prazo, visando adaptar essas ferramentas a diferentes níveis educacionais e contextos de ensino.

Palavras-chave: Jogos Sérios. Programação. Educação. Ensino.

1 INTRODUÇÃO

Os jogos digitais são uma ferramenta poderosa no aprendizado lúdico e têm sido aplicados com êxito em diversas áreas do conhecimento (ZYDA, 2005). Com a crescente e dinâmica evolução tecnológica aplicada ao meio educativo, as formas de aprendizado tornaram-se mais acessíveis (METTLER; PINTO, 2015), e os jogos digitais demonstram-se cada vez mais eficazes e integrados ao cotidiano como uma alternativa de ensino. Quando empregados como instrumentos pedagógicos, esses jogos contribuem para a democratização do conhecimento, facilitando sua absorção por parte dos usuários (GARCÊZ; OLIVEIRA, 2022). Tudo isso contribui para que o mundo caminhe em uma direção onde educação e conhecimento se tornem menos privilégios de poucos (ZYDA, 2005).

O desenvolvimento tecnológico no meio digital colabora diretamente com a educação, uma vez que as novidades deste campo corroboram para avanços pedagógicos (SAMPSON et al., 2009). Essa colaboração torna os jogos digitais mais acessíveis, tanto em termos de custo quanto de plataformas, o que facilita o aprendizado de temas abstratos como programação (REIS; MEDEIROS; ARANHA, 2013).

O ensino de programação pode ser desafiador, dado que suas abordagens são, muitas vezes, distantes da realidade (DEITEL; DEITEL, 2009). No entanto, muitos meios têm sido desenvolvidos para minimizar essas barreiras presentes no ensino de programação. Com o avanço no meio tecnológico e educacional, tais barreiras tendem a ser reduzidas (GARCÊZ; OLIVEIRA, 2022).

Nesse cenário, o objetivo desta pesquisa não é apenas discutir o uso de jogos digitais de forma genérica. O foco é realizar um mapeamento sistemático (KITCHENHAM et al., 2009) para identificar quais jogos sérios existentes apoiam especificamente o processo de ensino e aprendizagem de lógica de programação. Esses jogos sérios, voltados para fins educativos, são desenvolvidos com a intenção de superar as dificuldades tradicionais do ensino de programação, oferecendo uma abordagem mais interativa e envolvente para os estudantes (METTLER; PINTO, 2015). A partir dessa análise, busca-se destacar os jogos mais eficazes nesse contexto, fornecendo um acervo relevante para professores e estudantes que desejam utilizar essas ferramentas no ensino de lógica de programação (REIS; MEDEIROS; ARANHA, 2013).

Nesse contexto, o objetivo desta pesquisa não é simplesmente discutir o uso de jogos digitais no aprendizado de forma genérica. O foco principal é realizar um mapeamento sistemático para identificar os jogos sérios existentes que apoiam especificamente o processo de ensino e aprendizagem de lógica de programação. Esses jogos sérios, voltados para finalidades educativas, são desenvolvidos com a intenção de superar as dificuldades tradicionais no ensino de programação, oferecendo uma abordagem mais interativa e engajante para estudantes. A partir dessa análise, são destacados os jogos

mais eficazes nesse contexto, proporcionando um acervo relevante para professores e estudantes que desejam utilizar essas ferramentas no ensino de lógica de programação.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção de fundamentação teórica são apresentados tópicos importantes para o melhor entendimento do trabalho.

2.1 INTRODUÇÃO AO ENSINO DE PROGRAMAÇÃO E JOGOS EDUCATIVOS

A programação tem se tornado, ao longo do tempo, uma área cada vez mais requisitada no mercado de trabalho. Segundo o Infojobs, apenas em 2023, está previsto um crescimento de 20% na busca por capacitação relacionado a área de programação. Dessa forma, para corresponder toda essa demanda, o mercado se mobiliza por meio de uma maior oferta — seja por empregos, cursos de capacitação ou estudos para melhorar o ensinamento e aprendizado na área.

Embora existam diversos meios de aprendizado em programação, muitos alunos enfrentam dificuldades para assimilar questões relacionadas a este assunto visto que a área da programação possui conceitos e aplicações que podem ser consideradas mais abstratas em comparação a outras disciplinas. Como alternativa para diversificar e complementar o processo de aprendizagem, destacam-se os jogos digitais educacionais. De acordo com Victal et al. (2015), os jogos digitais são ferramentas muito importantes para o aprendizado, o que os torna indispensáveis na educação.

2.2 JOGOS SÉRIOS

Um jogo pode ser definido como uma atividade recreativa e estimulante que envolve qualquer tipo de competição com um conjunto de regras, sendo estas regras executadas em um ambiente restrito, no qual existe envolvimento social em um espaço e tempo determinados (ZUCARELLI e COUTO, 2013), geralmente com um objetivo específico.

Por outro lado, um serious game é propriamente um jogo, mas projetado com um propósito educacional específico. Ele não é focado para entretenimento, mas em ensinar determinado conteúdo ou treinar habilidades que podem ser aplicadas no mundo real. Serious games visam integrar entretenimento e aprendizado, possibilitando uma abordagem mais séria para a conquista de objetivos educacionais, através do uso de conflitos e desafios. (METTLER e PINTO, 2015; ZYDA, 2005).

Para melhor entendimento de jogos e serious games, os jogos são criados com o objetivo principal de proporcionar diversão, enquanto os serious games visam ensinar habilidades específicas de maneira eficaz. No contexto educacional, jogos têm sido usados como ferramentas para estimular

alunos e facilitar a compreensão de conceitos complexos. Já os serious games, no contexto da programação, podem simular ambientes práticos de codificação, auxiliando na aplicação de conceitos teóricos.

3 O MAPEAMENTO SISTEMÁTICO DA LITERATURA

Um Mapeamento Sistemático da Literatura (MSL) é uma investigação aprofundada de um fenômeno de interesse que fornece resultados específicos e minuciosos através da avaliação do conteúdo e da qualidade do material pesquisado (Kitchenham et al., 2007, p.), esse tipo de mapeamento enfatiza a importância de uma análise criteriosa da literatura para a obtenção de insights valiosos. Com o intuito de obter uma visão abrangente sobre o uso de jogos no ensino de programação, optou-se pela realização de um mapeamento. O intervalo temporal dos estudos selecionados foi delimitado de março de 2013 a março de 2024. Essa escolha foi fundamentada na premissa de que um período de 10 anos oferece uma perspectiva atualizada e abrangente para o contexto de pesquisa, além de cobrir um período relevante de estudos relacionados.

3.1 PLANEJAMENTO

Segundo Grant, Maria & Booth, Andrew. (2009), fatores como a definição clara dos objetivos do mapeamento e a alocação eficaz de responsabilidades são essenciais para a eficácia do processo. O planejamento do Mapeamento Sistemático da Literatura (MSL) foi realizado com o propósito de identificar os estágios fundamentais para a realização eficaz da pesquisa, análise e sumarização dos artigos, bem como os elementos contemplados em cada etapa. O planejamento e execução desse processo visaram à elaboração de um Protocolo de Mapeamento Sistemático da Literatura (PMSL).

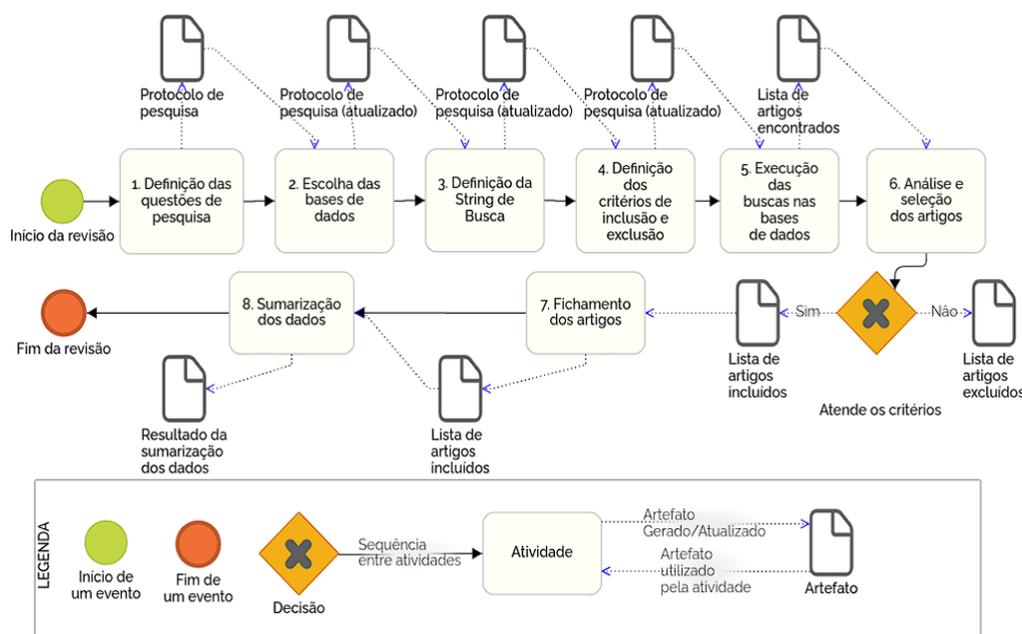
Os procedimentos do PRSL foram definidos através de reuniões entre o pesquisador e o professor. Os procedimentos podem ser visualizados na Figura 1, em suas respectivas ordens de execução, assim como o relacionamento dos artefatos com suas atividades no decorrer do processo.

3.1.1 Questões da pesquisa

De acordo com Petticrew, M., Roberts, H., às questões de pesquisa definem o que se pretende descobrir e os objetivos a serem alcançados, sendo essenciais para garantir que o mapeamento seja focado e relevante para os objetivos da pesquisa. Esse processo é realizado por meio da formulação de uma pergunta principal (QP) e perguntas secundárias (QS). A QP orienta a pesquisa de maneira holística, considerando que o estado da arte no uso de jogos no ensino de programação envolve diversos aspectos. Dessa forma, é necessário adotar escopos que delimitem aspectos específicos e

possibilitem uma análise mais minuciosa. As QS, por sua vez, especificam quais aspectos devem ser abordados para que a questão principal possa ser respondida. As questões formuladas para este estudo são:

Figura 1 - Etapas do mapeamento sistemático



HEFLO

Fonte: Elaborado pelos autores (2024)

- QP: Quais são os jogos existentes que auxiliam no ensino da disciplina de Programação?
- A QP tem como objetivo identificar os jogos digitais disponíveis que podem contribuir para o ensino dos conceitos iniciais de programação, fornecendo uma visão geral das opções acessíveis;
- QS1: Quais plataformas mais utilizadas?
- A QS1 busca identificar as plataformas mais frequentemente utilizadas para a execução dos jogos, com o intuito de determinar onde eles podem ser acessados;
- QS2: Quais são os tipos de jogos comumente usados?
- Esta questão visa especificar quais tipos de jogos foram encontrados nos trabalhos analisados, bem como quais são Open Source, o que pode ser relevante para a acessibilidade e customização dos recursos educacionais;
- QS3: Qual o idioma do jogo?
- A QS3 tem como objetivo verificar os idiomas nos quais os jogos estão disponíveis, considerando a questão da acessibilidade;
- QS4: Qual tipo de conhecimento o jogo visa treinar?

- Por meio da QS4, busca-se entender quais conhecimentos os jogos visam desenvolver como objetivo primário, permitindo a avaliação dos benefícios educacionais oferecidos por cada jogo.

3.1.2 Definição das bases de dados

A seleção das bases de dados deve ser baseada no escopo do mapeamento e nos objetivos da pesquisa, considerando bibliotecas digitais, repositórios de artigos científicos, periódicos eletrônicos, entre outros. As bases escolhidas para a busca foram ACM Digital Library e IEEE Xplore, devido à confiabilidade e grande quantidade de trabalhos publicados na área de programação, além de serem referências para a elaboração de base curricular em cursos de computação. Entre os critérios utilizados para a definição da base de dados, podemos elencar:

- Disponibilidade de acesso por meio da internet;
- A disponibilidade destas bases para consulta/download gratuito dos trabalhos para estudantes da UNIFAP.
- Esta pesquisa executou buscas automatizadas nas bases ACM Digital Library e IEEE Xplore por meio do uso da string de busca estabelecida no processo.

3.1.3 String de busca

Neste estudo, foi empregado o método PIO - Population (Participantes), Intervention (Tipo de Intervenção) e Outcomes (Resultados), para a formulação da string de busca, utilizando palavras-chave relevantes para o contexto da pesquisa. O Quadro 1 apresenta a aplicação desse método, alinhado com os objetivos deste estudo.

Quadro 1 - Método PIO

Population (Participantes)	Acadêmicos de Computação.
Intervention (Intervenção)	Uso de jogos para apoiar o ensino de Programação.
Outcomes (Resultados)	<ol style="list-style-type: none"> 1. Áreas e conhecimentos de programação contemplados; 2. Metodologias de ensino; 3. Plano de aula; 4. Jogos;

Fonte: Elaborado pelos autores (2024)

A criação da String de busca precisa é essencial para o abrangimento de estudos pertinentes à pesquisa. Baseado na questão principal e nas questões secundárias estabelecidas previamente, foi realizada uma combinação de palavras-chave para determinar uma String de busca que seja pertinente.

Para a validação da string foram estabelecidos os seguintes critérios: retornar entre 140 e 305 resultados obtidos com as bases utilizadas; e, foram determinadas palavras chaves que deveriam conter nos trabalhos retornados, caso ainda fosse retornado muitas palavras-chave diferentes, era essencial recalibrar a string de busca.

A string de busca é a combinação de termos(palavras-chave) e operadores lógicos “OR”, “AND” e “NOT”, como mostrado no quadro 2 e 3, que são utilizados para recuperar estudos relevantes nas bases de dados selecionadas. No contexto do método de pesquisa delineado por Malcher et al. (2015), a utilização do operador 'OR' é imperativa para associar palavras-chave e sinônimos, enquanto o emprego do operador 'AND' é apropriado quando a intenção é associar um conjunto específico de palavras-chave a todos os elementos investigados. A string gerada pode ser vista no quadro abaixo.

Quadro 2 - String de busca

(("Learn to Code" OR "Programming Fundamentals" OR "Programming Fundamental" OR "Programming Basic" OR "Programming Basics" OR NOT("Object Oriented")) AND (teaching OR learning OR training OR education OR tutorial) AND (Game* OR Serious Game* OR Simulator*))

Fonte: Elaborado pelos autores. (2024)

Dentre os trabalhos pesquisados, foram incluídas publicações de conferências, revistas, artigos de jornal e periódicos científicos, abrangendo o período de março de 2013 até março de 2024, totalizando um intervalo de 11 anos.

3.1.4 Critérios de inclusão e exclusão

A seleção dos estudos foi guiada pelos critérios de inclusão e exclusão, os quais estabelecem os parâmetros claros para a definição dos estudos a serem considerados. A aplicação rigorosa desses critérios é fundamental para garantir a relevância e a consistência na escolha dos estudos para a revisão (HIGGINS, J.; GREEN, S, 2008).

Após a realização da busca automatizada nas bases de dados, iniciou-se a fase de seleção dos trabalhos. O processo de triagem (pré-seleção) foi dividido em duas fases distintas. Na primeira fase, foram analisados os títulos, resumos e palavras-chave. Na segunda fase, realizou-se uma revisão detalhada dos trabalhos, para avaliar se estes atendiam aos critérios estabelecidos para exclusão e inclusão e exclusão, como delineado no Quadro 3.

Quadro 3 - Critérios de seleção dos estudos primários

ID	Critério de Inclusão (CI) ou Critério de Exclusão (CE)
CE.1	Artigos que não estejam disponíveis livremente para consulta ou download (em versão completa) nas fontes de pesquisa ou por meio de busca manual (para artigos que não sejam fornecidos na íntegra) realizada nas ferramentas de busca Google (http://www.google.com.br/) e/ou Google Scholar (http://scholar.google.com.br/);
CE.2	Artigos que claramente não atendam às questões de pesquisa.
CE.3	Artigos repetidos (em mais de uma fonte de busca) tiveram apenas sua primeira ocorrência considerada;
CE.4	Estudos enquadrados como resumos, keynote speeches, cursos, tutoriais, workshops e afins;
CE.5	Artigos que não mencionam as palavras-chave da pesquisa no título, resumo ou nas palavras-chave do artigo, salvo trabalhos que abordem melhoria do processo de software nos quais seja observada a possibilidade do Controle Estatístico de Processos ser tratado ao longo do trabalho;
CE.6	Excluir se o estudo não estiver inserido no contexto;
CE.7	Excluir se o estudo não estiver apresentado em uma das linguagens aceitas (Inglês e Português).

Fonte: Elaborado pelos autores (2024)

4 RESULTADOS DO MAPEAMENTO DA LITERATURA

Nesta seção, são apresentados os resultados obtidos a partir da execução do planejamento, com o objetivo de responder às questões principal (QP) e secundárias (QS) previamente definidas.

4.1 RESULTADOS DOS ESTUDOS PRIMÁRIOS

As strings de busca aplicadas nas bases de dados ACM Digital Library e IEEE Xplore resultaram em um total de 755 trabalhos. A base ACM Digital Library retornou 290 artigos (38,41%), apresentando o menor número de resultados, enquanto a base IEEE Xplore apresentou um total de 465 artigos (61,58%).

A tabela 1 exibe o total de artigos classificados de acordo com os critérios de pré-seleção, Critérios de Exclusão (CE) e Critérios de Inclusão (CI). Cada linha representa o número total de estudos obtidos de cada indexador. A fase de pré-seleção abrangeu o maior número de artigos (561) dentre as pesquisas eliminadas, seguida pelos CE.2 e CE.1, com 165 e 17 exclusões, respectivamente. Os dados na coluna CI.1 indicam os trabalhos que atenderam o critério de inclusão estabelecido.

Ainda na tabela 1, observa-se que o motor de busca IEEE Xplore teve a maior quantidade de artigos excluídos (453), sendo seguidos pelo CE.2, que representou o principal critério de eliminação de artigos com 158 exclusões. Por outro lado, o ACM Digital Library excluiu 285 artigos durante a pré-seleção, sendo o CE.2 o critério com maior taxa de eliminação de trabalhos, totalizando 7 exclusões.

Tabela 1 - Estudos retornados com base nos critérios de inclusão e exclusão

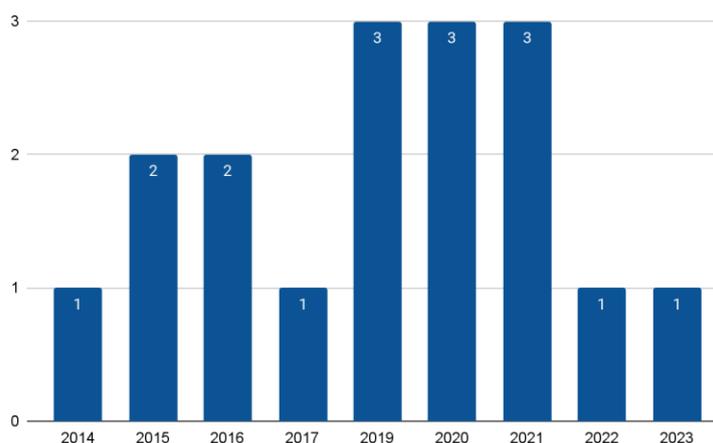
Estudos Primários					
Fonte	Trabalhos retornados	Pré-seleção	CE.2	CE.1	CI.1
IEEE Xplore	465	289	158	6	12
ACM Digital Library	290	272	7	6	5
Total	755	561	165	12	17

Fonte: Elaborado pelos autores (2024)

O resultado da seleção dos estudos resultou em 17 artigos publicados entre os períodos de 2014 até 2023. A Figura 2 ilustra a quantidade de artigos encontrados por ano. Os anos com o maior número de publicações foram 2019, 2020 e 2021, com 3 artigos cada. Os anos 2015 e 2016 tiveram 2 artigos cada. Em contrapartida, os anos 2014, 2017, 2022 e 2023 apresentaram a menor taxa de publicações, com apenas 1 artigo em cada um desses anos.

Em se tratando da distribuição dos tipos de publicação, os resultados demonstram que a maioria das contribuições se concentrou em conferências, com 13 artigos (76,5%). Além disso, houve publicações em revistas, com 2 artigos (11,8%), artigo de jornal e workshop, ambos com 1 artigo (5,9%).

Gráfica 1 - Publicações por ano



Fonte: Elaboração própria (2024)

A Tabela 2 classifica os eventos de participação de acordo com o tipo de publicação mencionado anteriormente. Notavelmente, no caso de conferências, a “IEEE Frontiers in Education Conference (FIE)”, se destacou com a maior quantidade de trabalhos, totalizando 2 publicações. Isso difere dos demais eventos de outros tipos de publicações, onde houve uma igualdade numérica de 1

publicação em cada um. Essa distribuição pode ser observada na Tabela 2, que apresenta os dados detalhados.

Tabela 2 - Locais de publicação.

Conferência	Artigos
IEEE International Conference on Internet of Things and Intelligence Systems (IoT&IS)	1
IEEE Frontiers in Education Conference (FIE)	2
IEEE 20th International Conference on Advanced Learning Technologies (ICALT)	1
IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T)	1
International Conference on Intelligent Systems and Advanced Computing Sciences (ISACS)	1
ISDOC '14: Proceedings of the International Conference on Information Systems and Design of Communication	1
NordiCHI '16: Proceedings of the 9th Nordic Conference on Human-Computer Interaction	1
IDC '16: Proceedings of the The 15th International Conference on Interaction Design and Children	1
IDC '21: Interaction Design and Children	1
SoutheastCon	1
IEEE International Conference on Advanced Learning Technologies (ICALT)	1
XLIII Conferência Latino-Americana de Computação (CLEI)	1
Revistas	Artigos
IEEE Access	1
IEEE Revista Iberoamericana de Tecnologias del Aprendizaje	1
Artigo de jornal	Artigos
ACM Transactions on Computing Education	1
Workshop	Artigos
IEEE Blocks and Beyond Workshop (B&B)	1

Fonte: Elaborado pelos autores (2024)

Além disso, os detalhes sobre as universidades, autores e aos países com maior número de artigos publicados sobre o tema podem ser encontrados na Tabela 3. Na contagem, todas as aparições dos elementos foram catalogadas, independentemente de serem autores ou coautores dos trabalhos. No que diz respeito às universidades, todas as instituições dos trabalhos selecionados contribuíram igualmente, com cada uma apresentando 1 publicação relacionada ao tema de pesquisa. Quando se

trata dos países que mais contribuíram, os Estados Unidos lideram com 4 pesquisas publicadas, seguidos pela Alemanha, com 3 artigos publicados. No que diz respeito aos autores que mais apareceram nos trabalhos selecionados, nenhum deles esteve envolvido em mais de 1 trabalho.

Os dados apresentados na Tabela 3 são valiosos para identificar as universidades, autores e países que mais se destacaram no campo abordado neste trabalho. Essas informações podem servir como fonte de orientação e colaboração para o desenvolvimento de pesquisas relacionadas.

4.1.1 Qp: quais são os jogos existentes que auxiliam no ensino da disciplina de programação?

Esta questão visa identificar e catalogar os jogos digitais disponíveis que podem contribuir para o ensino dos conceitos iniciais de programação, proporcionando uma visão abrangente das opções acessíveis para educadores e alunos.

A pesquisa resultou na identificação de um total de 49 jogos que se destacam por suas contribuições ao ensino de programação. Esses jogos foram selecionados com base em critérios rigorosos, que garantiram a relevância e a qualidade dos estudos analisados. A lista completa dos jogos identificados, juntamente com as áreas de conhecimento abordadas, pode ser consultada no Apêndice A deste trabalho. Essa informação é valiosa não apenas para educadores que buscam integrar jogos em suas práticas pedagógicas, mas também para pesquisadores que desejam explorar mais a fundo o potencial dos jogos no ensino de programação.

4.1.2 Qs1: quais plataformas mais utilizadas?

Levando em consideração o contexto desta pesquisa, é essencial considerar as plataformas em que esses jogos estão disponíveis. A análise dos resultados obtidos revela uma variedade de opções, com Windows sendo a plataforma mais amplamente utilizada, com um total de 26 (27,7%) jogos disponíveis. Além disso, o sistema operacional MacOS oferece suporte a 20 jogos e o SteamOS + Linux com 14 jogos, ambos voltados para dispositivos desktop, e a plataforma web dando suporte para 12 jogos. A Figura 3 ilustra tanto as plataformas citadas acima como outras que não dão suporte a tantos jogos dos 49 selecionados para essa pesquisa.

4.1.3 Qs2: quais são os tipos de jogos comumente usados?

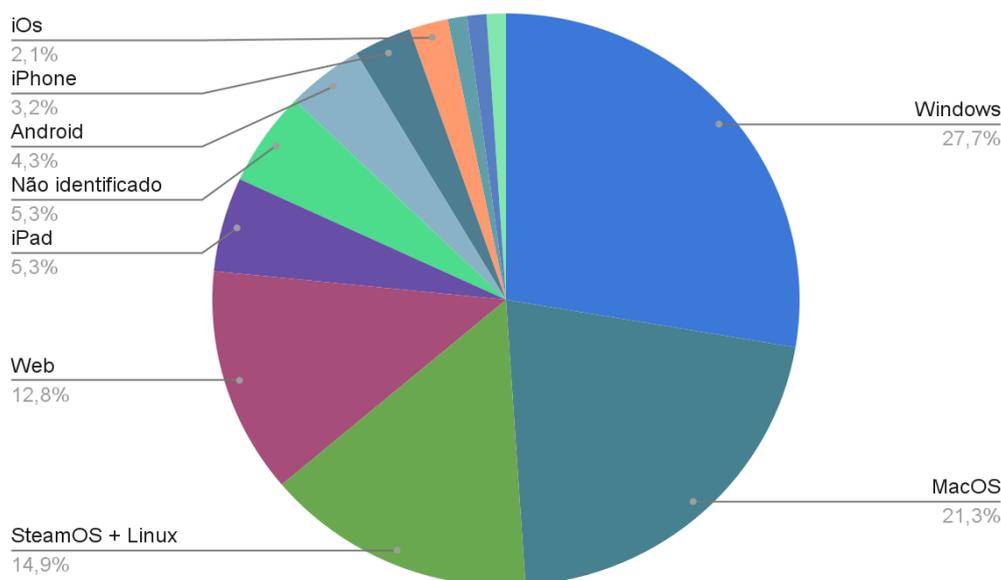
A figura 4 revela uma distribuição variada dos tipos de jogos mais frequentemente utilizados dentre os trabalhos selecionados. Os jogos “Indie” lideram a preferência, contabilizando 25 instâncias, refletindo possivelmente a crescente popularidade e diversidade de desenvolvedores independentes na indústria de jogos. Em segundo lugar, os jogos educativos de destacam, totalizando 16 jogos que

abordam esse gênero. Além disso, observa-se uma presença notável de jogos de simulação (10), puzzle (14) e aventura (7), indicando uma ampla variedade de gêneros entre os jogos selecionados.

A categorização "Não identificado" sugere a presença de jogos cujo gênero não foi claramente especificado. Este panorama diversificado evidencia a complexidade e a riqueza do cenário de jogos atual, atendendo a uma ampla gama de interesses e propósitos. Os demais tipos jogos que não apareceram com muita frequência podem ser vistos na figura abaixo juntamente com os demais citados acima.

O número de artigos citados que são classificados como sendo Open Source foram 4 jogos (Blockly Games, Pencil Code, Robô Instructus() e Stone Story RPG) que foram utilizados nos trabalhos retornados, enquanto 32 jogos não puderam ser classificados como sendo Open Source.

Gráfico 2 - Plataformas mais utilizadas para os jogos.

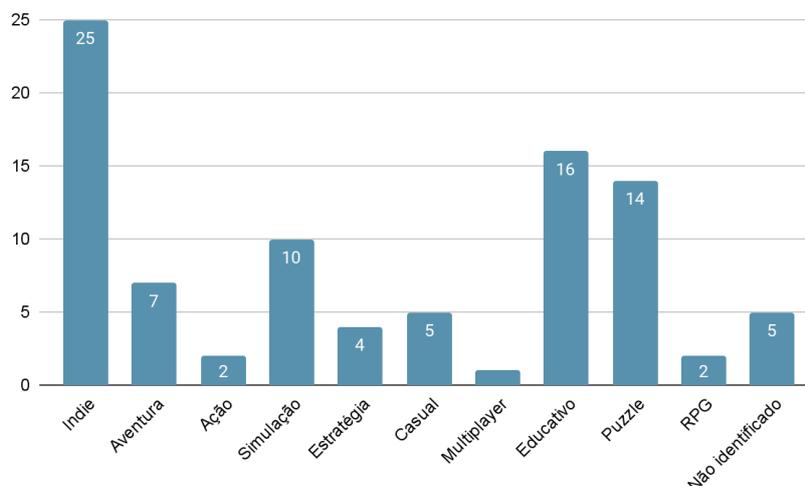


Fonte: Elaborado pelos autores (2024)

4.1.4 Qs3: qual o idioma do jogo?

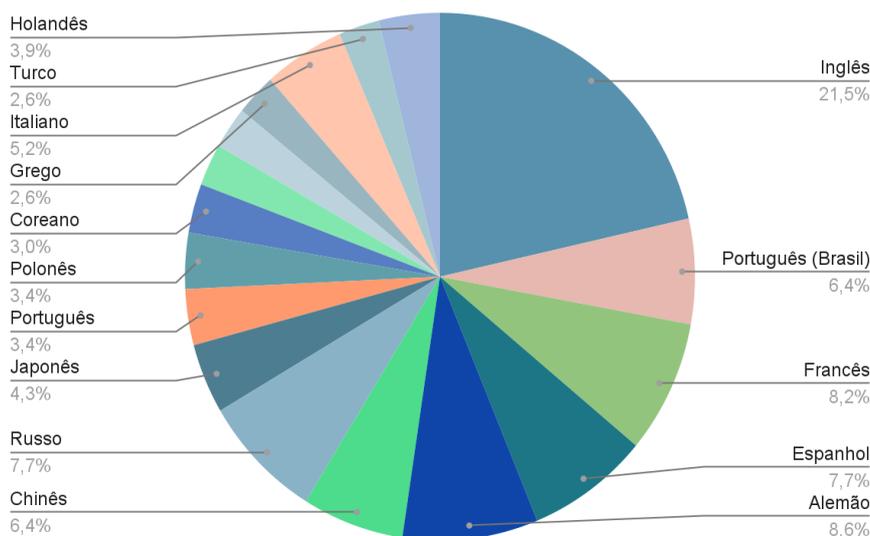
A figura 5 apresenta os idiomas disponíveis para os 49 jogos educacionais selecionados. Ela inclui os 13 idiomas mais frequentemente disponibilizados, desde aqueles presentes em todos os jogos, como o inglês (com 49 aparições, 21,5%), até os que aparecem em pelo menos 6 jogos. Os idiomas mais comuns incluem inglês, português (Brasil), francês, espanhol, alemão, chinês e russo

Gráfico 3 - Tipos de jogos.



Fonte: Elaborado pelos autores (2024)

Gráfico 4 - Idioma dos jogos.



Fonte: Elaborado pelos autores (2024)

4.1.5 Qs4: qual tipo de conhecimento o jogo visa treinar?

Com base nos trabalhos analisados foram levados em consideração nove tópicos que são essenciais no aprendizado de programação. Os tópicos se destringem da seguinte forma: Lógica de programação, desenvolve habilidades de pensamento lógico e resolução de problemas; Sintaxe de linguagens de programação, integra conceitos de sintaxe em desafios e tarefas interativas, os alunos podem aprender de forma significativa, aplicando esses conhecimentos em um contexto prático e realista; Manipulação de dados, os jogadores podem aprender a armazenar, modificar e recuperar informações, desenvolvendo habilidades cruciais para a programação; Funções e procedimentos, permite que os usuários compreendam a importância da modularidade e organização do código.

Ao criar e chamar funções específicas para tarefas distintas no jogo, os aprendizes podem ganhar uma compreensão prática da reutilização de código e da eficiência na resolução de problemas; Tratamento de erros e debugging, ao enfrentar erros e bugs no contexto do jogo, os jogadores aprendem a identificar, diagnosticar e corrigir problemas no código, desenvolvendo habilidades essenciais para a resolução de problemas na programação; Projetos práticos, desenvolver projetos práticos dentro do jogo permite a consolidação de conhecimentos e construção de um portfólio de experiências tangíveis; Adaptação a novos desafios, incentiva a flexibilidade e a resiliência, qualidades cruciais para programadores que frequentemente enfrentam novos problemas e tecnologias; Feedback imediato, os jogadores podem ajustar rapidamente sua abordagem, compreender melhor os conceitos e reforçar o aprendizado de forma prática; Estrutura de repetição e condição, ajuda os jogadores a desenvolverem a lógica necessária para criar programas mais complexos e eficientes.

Abaixo, na tabela 3, pode-se conferir a quantidade de jogos encontrados relacionados a cada conhecimento mencionado anteriormente.

Tabela 3 - Quantidade de jogos encontrados por conhecimento

Quantidade de jogos encontrados por Conhecimentos		
Lógica de programação 49 jogos encontrados	Sintaxe de Programação 23 jogos encontrados	Manipulação de Dados 19 jogos encontrados
Funções e Procedimentos 24 jogos encontrados	Tratamento de Erros e Debugging 17 jogos encontrados	Projetos Práticos 31 jogos encontrados
Adaptação a Novos Desafios 39 jogos encontrados	Feedback Imediato 28 jogos encontrados	Estruturas de repetição e condição 29 jogos encontrados

Fonte: Elaboração própria (2024)

A lista completa dos jogos identificados, juntamente com as áreas de conhecimento abordadas, pode ser consultada no Apêndice A deste trabalho.

5 ANÁLISE SOBRE OS PRINCIPAIS JOGOS

Esta análise, explorou alguns dos principais jogos e ferramentas educacionais voltados para o aprendizado de programação. Esses jogos, além de fornecerem uma experiência interativa, também abordam conceitos fundamentais como lógica, estruturas de repetição, funções, e tratamento de erros, ajudando os jogadores a desenvolverem suas habilidades de forma prática e divertida.

CodinGame é uma plataforma que oferece desafios práticos em várias linguagens de programação. Segundo a documentação oficial, “os desafios de CodinGame abrangem uma ampla gama de conceitos de programação, permitindo que os usuários pratiquem desde a lógica básica até

estruturas de dados avançadas" (CodinGame, 2024). Seu maior diferencial é o feedback imediato, que permite aos usuários corrigirem seus erros instantaneamente e aprimorarem suas habilidades em lógica e sintaxe de programação, além de trabalhar com estruturas de repetição, condições e manipulação de dados em tempo real. A plataforma também aborda funções e algoritmos avançados, o que a torna adequada tanto para iniciantes quanto para programadores experientes. A adaptação a novos desafios é facilitada por uma ampla gama de exercícios que variam de fácil a difícil.

Desenvolvido pela Microsoft Research, Code Hunt ensina programação por meio de quebra-cabeças que exigem a correção de código pré-escrito. Conforme documentado oficialmente, "Code Hunt oferece uma abordagem prática para ensinar programação por meio de desafios interativos que reforçam a lógica, o uso de estruturas de repetição e o tratamento de erros" (Microsoft Research, 2024). Um dos pontos fortes do jogo é a adaptação a novos desafios e o feedback imediato, permitindo aos jogadores testarem suas soluções em tempo real e aprenderem com seus erros. O jogo foca em lógica de programação, estruturas de repetição e o tratamento de erros por meio de debugging intuitivo. O ambiente é ideal para a prática da sintaxe de programação, com um enfoque em loops e condições.

Wu's Castle ensina lógica de programação em um cenário de aventura. Conforme descrito na documentação, "Wu's Castle proporciona um ambiente imersivo no qual a lógica de programação e a manipulação de dados são introduzidas de forma gradual, por meio de uma narrativa divertida e desafiadora" (Wu's Castle, 2024). O jogo se concentra em manipulação de dados e conceitos de fluxo de controle através de estruturas de repetição e condicionais. Seu grande diferencial é a narrativa envolvente, que torna o aprendizado divertido e acessível, ao mesmo tempo que incentiva o uso de funções e procedimentos para resolver os desafios apresentados.

Code World utiliza blocos de código para ensinar programação de forma visual. A documentação afirma que "Code World permite que iniciantes visualizem o fluxo de controle e compreendam o comportamento do código de forma clara e intuitiva" (Code World, 2024). Através de blocos lógicos, os usuários aprendem fluxo de controle, sequência de comandos, estruturas de repetição e condições. A plataforma é excelente para iniciantes, pois oferece uma compreensão clara e intuitiva do comportamento do código e permite a adaptação a novos desafios à medida que os alunos progridem.

Em Coding Planets os jogadores controlam robôs em diferentes planetas, resolvendo problemas que requerem lógica e o uso de estruturas de repetição e condições. Segundo a documentação oficial, "Coding Planets se concentra no desenvolvimento do raciocínio lógico por meio da aplicação prática de loops, condições e lógica de programação em ambientes desafiadores"

(Coding Planets, 2024). A plataforma ajuda a reforçar o raciocínio lógico e incentiva o uso de funções para soluções mais complexas. Adaptação a novos desafios é central na mecânica do jogo, já que cada planeta apresenta novos problemas e níveis de dificuldade.

Focado em funções e procedimentos, o Robô Instructus desafia os jogadores a encontrarem soluções criativas para os problemas apresentados. De acordo com a documentação, "Robô Instructus incentiva a criatividade na resolução de problemas ao mesmo tempo que desafia os usuários a otimizarem suas soluções para uma eficiência máxima" (Robô Instructus, 2024). A plataforma é ideal para aqueles que desejam aprender a otimização de código e a organização de projetos de forma eficiente. O jogo oferece feedback contínuo sobre as soluções propostas, incentivando o processo de tentativa e erro.

Em Debugger 3.16: Hack'n'Run, os jogadores atuam como depuradores, aprendendo sobre debugging e tratamento de erros em tempo real. A documentação destaca que "Debugger 3.16 coloca o jogador na posição de um depurador, onde ele deve identificar e corrigir erros de programação em tempo real, aprendendo o valor da otimização e do teste constante" (Debugger 3.16, 2024). O jogo é projetado para ensinar a importância de testar, corrigir e otimizar código de forma prática. É uma excelente ferramenta para aqueles que desejam melhorar suas habilidades em encontrar e corrigir erros de programação, com foco em lógica, loops e condições.

Karel é uma ferramenta educacional clássica que introduz os fundamentos da programação, como sintaxe, loops, condicionais e tratamento de erros, de forma simples e controlada. De acordo com a documentação, "Karel the Robot foi projetado para ajudar os alunos a entender os conceitos básicos de programação por meio da execução de tarefas simples e controladas em um ambiente de fácil utilização" (Karel, 2024). Os jogadores programam um robô para realizar tarefas específicas, facilitando o aprendizado de lógica e controle de fluxo. A simplicidade da ferramenta é um ponto forte para iniciantes que estão começando a entender como programar.

Utilizando programação visual, Glitch Space ensina conceitos de lógica funcional e fluxo de dados por meio de quebra-cabeças. Segundo a documentação oficial, "Glitch Space utiliza um sistema de programação visual para ajudar os jogadores a entender como manipular dados e resolver problemas usando lógica funcional" (Glitch Space, 2024). O jogo oferece uma experiência de aprendizado tangível, com uma interface intuitiva que facilita a compreensão de como os dados podem ser manipulados para resolver problemas. O foco está na lógica de programação e no uso de blocos de código para transformar dados de maneira eficiente.

Screeps: World é uma plataforma multiplayer que permite aos jogadores programarem colônias em JavaScript, o que exige a adaptação constante a um ambiente dinâmico e competitivo.

Segundo a documentação oficial, "Screeps é um jogo de programação em tempo real, onde os jogadores controlam colônias inteiramente por meio de scripts escritos em JavaScript, promovendo a resolução de problemas complexos em um ambiente multiplayer" (Screeps, 2024). O jogo incentiva a otimização contínua do código e o uso de funções e procedimentos para controle eficiente de recursos. A complexidade do jogo desafia os jogadores a resolverem problemas de maneira criativa e a lidarem com projetos práticos em um ambiente de tempo real.

Apesar de os jogos analisados acima terem seus pontos fortes ressaltados, vale lembrar que eles não se limitam a esses conhecimentos específicos. Estes são tratados como principais pois abrangem simultaneamente os nove indicadores selecionados após um mapeamento sistemático da literatura, que contribuem de forma significativa para o ensino de programação básica.

6 CONCLUSÕES

Este estudo evidenciou a eficácia e o potencial dos jogos digitais como ferramentas pedagógicas específicas para o ensino de programação. A partir de uma revisão sistemática da literatura, identificou-se um conjunto de 49 jogos distintos, categorizados por gênero, plataforma e tipo de conhecimento abordado, revelando-se amplamente adaptados para o ensino de conceitos-chave em programação, como lógica, manipulação de dados, sintaxe, funções e tratamento de erros. A diversidade desses jogos, aliada a suas características interativas e práticas, permite uma abordagem instrucional que não apenas complementa métodos tradicionais, mas também propicia um ambiente de aprendizado dinâmico e altamente engajador.

Apesar dos resultados positivos, o presente estudo apresenta algumas limitações que devem ser consideradas ao interpretar seus resultados. Primeiramente, a revisão sistemática foi realizada exclusivamente nas bases ACM Digital Library e IEEE Xplorer, escolhidas por sua relevância na área de computação. No entanto, essa escolha restringe o escopo dos estudos analisados, excluindo trabalhos que poderiam ser encontrados em outras bases acadêmicas. Além disso, o foco desta revisão esteve direcionado ao uso de jogos para o ensino de conceitos de programação básica, como lógica, manipulação de dados e sintaxe, limitando a aplicabilidade dos resultados em níveis intermediários ou avançados de programação. Conceitos mais complexos, como programação orientada a objetos, estruturas de dados e algoritmos avançados, exigem abordagens específicas e, possivelmente, jogos diferenciados. Portanto, estudos futuros poderiam explorar jogos que abordam esses conteúdos para avaliar sua eficácia no ensino de habilidades mais sofisticadas em programação. Por fim, há a questão da influência temporal. O período coberto pela revisão, embora abrangente, pode refletir uma tendência recente ou pontual no desenvolvimento de jogos educacionais. Assim, é possível que

mudanças rápidas no mercado de jogos e nas práticas pedagógicas afetem a relevância dos achados ao longo do tempo, exigindo revisões periódicas para manter o estudo atualizado com as inovações tecnológicas e educacionais emergentes.

Quanto ao impacto do trabalho, do ponto de vista acadêmico, este estudo contribui substancialmente para o campo da Educação em Computação, oferecendo uma análise criteriosa e abrangente do uso de jogos no ensino de programação. Ao mapear e sistematizar os jogos disponíveis, esta pesquisa provê uma base sólida para que educadores e instituições incorporem, com embasamento teórico e prático, recursos de jogos em seus currículos. Esse levantamento detalhado permite que futuros pesquisadores e docentes adotem uma visão mais informada sobre quais jogos são mais eficazes em cada área da programação, otimizando, assim, as práticas de ensino e potencializando o desenvolvimento de habilidades críticas.

Além disso, o trabalho contribui para a expansão das fronteiras de pesquisa na área de jogos sérios, reforçando a necessidade de investigações mais profundas sobre a eficácia dos jogos digitais e sua aplicabilidade no contexto acadêmico. Os resultados desta revisão sistemática também indicam que o uso de jogos no ensino pode ser uma estratégia eficaz para aumentar a retenção e o desempenho dos alunos em disciplinas com elevado nível de abstração, como programação.

Socialmente, a utilização de jogos no ensino de programação representa uma abordagem democratizante, ao facilitar o acesso ao conhecimento técnico de maneira inclusiva e atraente. Jogos digitais, pela sua acessibilidade em plataformas amplamente utilizadas (como Windows, MacOS e Linux) e por sua versatilidade de aplicação, possuem o potencial de reduzir barreiras de entrada para o aprendizado em tecnologia, particularmente em comunidades com acesso limitado a recursos educacionais formais. Ao tornar o ensino de programação mais interativo e motivador, os jogos contribuem para uma inclusão digital mais efetiva, possibilitando que indivíduos de diferentes origens e contextos socioeconômicos desenvolvam habilidades valiosas para o mercado de trabalho. Assim, a implementação de jogos educacionais em programação pode ser vista como uma ferramenta de transformação social, com potencial para impactar positivamente a formação de profissionais capacitados e reduzir desigualdades no acesso ao ensino tecnológico.

Este estudo abre caminho para múltiplas vertentes de pesquisas futuras. Recomenda-se que novas investigações explorem a eficácia dos jogos digitais para o desenvolvimento de habilidades de programação a longo prazo, com um enfoque em estudos longitudinais que permitam avaliar o progresso dos alunos e a retenção do conhecimento adquirido através desses jogos.

Além disso, seria relevante investigar o potencial de personalização dos jogos para diferentes estilos de aprendizado e perfis de alunos, a fim de otimizar a adaptação do ensino de programação

para uma audiência heterogênea. Outro eixo promissor para futuras pesquisas envolve a análise de impactos psicológicos e motivacionais do uso de jogos no ensino, compreendendo de que forma esses recursos podem influenciar a autoestima, o interesse contínuo e a percepção de eficácia dos estudantes na área de computação.

Essas direções sugeridas visam fortalecer a compreensão sobre o papel dos jogos digitais como facilitadores no aprendizado de programação e consolidar práticas pedagógicas inovadoras que sejam adaptáveis e eficientes para os diversos contextos educacionais. A continuidade de investigações nesse campo é essencial para garantir que as novas gerações tenham acesso a um aprendizado em tecnologia que seja tanto tecnicamente robusto quanto socialmente inclusivo

REFERÊNCIAS

- CODE WORLD (2024). Code World: Visual programming environment. Disponível em: <https://www.codeworld.com/>. Acesso em: 09 nov. 2024.
- CODINGAME. (2024). CodinGame: Programming challenges. Disponível em: <https://www.codingame.com/>. Acesso em: 09 nov. 2024.
- CODING PLANETS. (2024). Coding Planets: Logic-based problem-solving. Disponível em: <https://www.codingplanets.com/>. Acesso em: 09 nov. 2024.
- DEITEL, P.; DEITEL, H. Como programar em C. 3. ed. São Paulo: Pearson Prentice Hall, 2009.
- DEBUGGER 3.16. (2024). Debugger 3.16: Real-time debugging game. Disponível em: <https://www.debugger316.com/>. Acesso em: 09 nov. 2024.
- GARCÊZ, A. V.; OLIVEIRA, J. M. L. The benefits of using digital games to teach programming to children. *Research, Society and Development*, v. 11, n. 17, 2022.
- GLITCH SPACE. (2024). Glitch Space: Visual programming puzzles. Disponível em: <https://www.glitchspace.com/>. Acesso em: 09 nov. 2024.
- GRANT, M.; BOOTH, A. A typology of reviews: An analysis of 14 review types and associated methodologies. *Health Information and Libraries Journal*, v. 26, p. 91-108, 2009.
- HIGGINS, J.; GREEN, S. *Cochrane Handbook for Systematic Reviews of Interventions*. London, Wiley-Blackwell, 2008.
- KAREL. (2024). Karel the Robot: Educational programming tool. Disponível em: <https://www.karelrobot.com/>. Acesso em: 09 nov. 2024.
- KITCHENHAM, B.; BRERETON, P.; BUDGEN, D.; TURNER, M.; BAILEY, J.; LINKMAN, S. Systematic literature reviews in software engineering: A systematic literature review. *Information and Software Technology*, v. 51, p. 7-15, 2009.
- MALCHER, P.; FERREIRA, D.; OLIVEIRA, S.; VASCONCELOS, A. Um mapeamento sistemático sobre abordagens de apoio à rastreabilidade de requisitos no contexto de projetos de software. *Revista de Sistemas de Informação da FSMA*, n. 16, p. 3-15, 2015.
- METTLER, T.; PINTO, R. Serious games as a means for scientific knowledge transfer: A case from engineering management education. *IEEE Transactions on Engineering Management*, v. 62, n. 2, p. 256-265, 2015.
- MICROSOFT RESEARCH. (2024). Code Hunt: Interactive coding game. Disponível em: <https://www.microsoft.com/en-us/research/project/code-hunt/>. Acesso em: 09 nov. 2024.
- PETTICREW, M.; ROBERTS, H. *Systematic Reviews in the Social Sciences: A Practical Guide*. Blackwell Publishing, 2006.

REIS, T.; MEDEIROS, J.; ARANHA, E. Ensino de programação utilizando jogos digitais: uma revisão sistemática da literatura. *Revista Novas Tecnologias na Educação*, v. 11, 2013.

ROBÔ INSTRUCTUS. (2024). Robô Instructus: Creative coding challenges. Disponível em: <https://www.roboinstructus.com/>. Acesso em: 09 nov. 2024.

SAMPSON, M.; MCGOWAN, J.; COGO, E.; GRIMSHAW, J.; MOHER, D.; LEFEBVRE, C. An evidence-based practice guideline for the peer review of electronic search strategies. *Journal of Clinical Epidemiology*, v. 62, n. 9, p. 944-52, 2009.

SCREEPS. (2024). Screeps: World multiplayer programming game. Disponível em: <https://www.screeps.com/>. Acesso em: 09 nov. 2024.

WU'S CASTLE. (2024). Wu's Castle: Programming logic game. Disponível em: <https://www.wuscastle.com/>. Acesso em: 09 nov. 2024.

ZYDA, M. From visual simulation to virtual reality to games. *Computer*, v. 38, n. 9, p. 25-32, 2005.

ZUCARELLI, I.; COUTO, L. *Jogo de tabuleiro em incentivo à alimentação infantil*. Universidade São Judas Tadeu, São Paulo, 2013.

APÊNDICE A

Tabela 1 - Jogos Identificados

Jogos	Conhecimentos								
	LP	SP	MD	FP	TED	PP	AND	FI	ERC
CodinGame	S	S	S	S	S	S	S	S	S
Wu's Castle	S	S	S	S	S	S	S	S	S
Algorithms: Explained and Anim	S	N	N	N	N	N	N	N	S
Coding Planets	S	S	S	S	S	S	S	S	S
Human Resource Machine	S	N	N	S	S	S	S	S	S
7 Billion Humans	S	N	N	PARC	PARC	S	S	S	S
MIT App Inventor	S	S	S	S	N	S	S	S	N
Blockly Games	S	S	S	N	N	S	S	PARC	S
Code.org	S	S	N	N	N	S	S	N	S
Gameblox	S	S	S	S	S	S	PARC	N	S
Pencil Code	S	S	S	S	N	S	S	N	S
Microsoft Make Code	S	S	PARC	N	N	S	N	PARC	S
Code Spell	S	S	S	S	PARC	S	PARC	S	N
Baba is You	S	N	N	N	N	N	N	N	N
SpaceChem	S	N	N	N	N	N	S	S	N
Cyber Sentinel	S	N	N	N	N	N	S	S	N
Neon Noodles	S	N	N	N	N	N	S	S	N
Autonauts	S	N	N	N	N	PARC	S	S	S
Robô Instructus()	S	S	S	S	S	S	S	S	S
Marco Run	S	N	N	N	N	N	S	S	S
Karel (IDE)	S	S	S	S	S	S	S	S	S
Daisy the Dinosaur	S	N	N	N	PARC	S	S	S	S
Cargo-Bot	S	N	N	PARC	N	PARC	S	S	S
Code Hunt	S	S	S	S	S	S	S	S	S
Code Combat	S	S	N	S	S	S	S	S	N
Move The Turtle	S	N	N	N	N	S	S	S	S
Robozzle	S	N	N	N	N	PARC	S	N	N
HMS - Haathi Mera Saath	S	N	N	N	N	N	N	N	N
The Odyssey of Phoenix	S	S	N	S	N	N	S	N	N
LightBot	S	N	N	N	N	PARC	S	S	S
Minecraft Education Edition (M:EE)	S	N	N	S	N	S	S	S	S
Code Craft	S	N	S	S	N	S	PARC	N	S
I hate this game	S	N	N	N	N	N	N	N	N
Markov Alg.	S	N	N	N	N	N	N	N	N
The Endless Mission	S	PARC	S	S	N	S	PARC	N	PARC

The Magic Circle	S	N	N	N	N	N	N	N	N
Code World	S	S	S	S	S	S	S	S	S
Debugger 3.16: Hack'n'Run	S	S	S	S	S	S	S	S	S
Glitch space	S	S	S	S	S	S	S	S	S
Algo Bot	S	N	N	PARC	N	PARC	S	N	N
Algotica Iterations	S	S	N	N	N	N	S	N	N
Jump Step Step	S	N	N	N	N	N	S	N	N
Bot Land (F2P)	S	S	N	N	S	S	S	S	S
Cool Headed	S	N	N	N	N	N	N	N	N
Glyphs Apprentice	S	N	N	N	N	N	S	N	N
Screeps: World	S	S	S	S	S	S	S	S	S
TIS-100	S	S	S	S	N	N	S	N	S
Robot Training	Não Encontrado								
MagiPlay	S	N	N	N	N	N	N	N	N

Fonte: Elaboração própria (2024)

Tabela 2 - Legenda das Abreviações da tabela 4 dos Assuntos nos Jogos

Conhecimentos		
LP - Lógica de programação	SP - Sintaxe de Programação	MD - Manipulação de Dados
FP - Funções e Procedimentos	TED - Tratamento de Erros e Debugging	PP - Projetos Práticos
AND - Adaptação a Novos Desafios	FI - Feedback Imediato	ERC - Estruturas de repetição e condição

Fonte: Elaboração própria (2024)