

Um Mapeamento Sistemático sobre o Ensino e Aprendizagem de Programação

Thiago Reis da Silva – IFMA/Campus São João dos Patos – thiago.reis@ifma.edu.br -
<https://orcid.org/0000-0003-4206-6801>

Italo Silva Barros – IFMA/Campus São João dos Patos – italo.barros@acad.ifma.edu.br -
<https://orcid.org/0000-0002-4295-8933>

Lívio Kayky Da Silva Sousa – IFMA/Campus São João dos Patos –
livio.silva@acad.ifma.edu.br - <https://orcid.org/0000-0002-8837-8942>

Apollo Lopes Damasceno Sá – IFMA/Campus São João dos Patos –
apolloed@acad.ifma.edu.br - <https://orcid.org/0000-0002-8749-5071>

Antonio Fhillipi Maciel Silva – UESPI/Campus Floriano – fhillipi.ti@frn.uespi.br -
<https://orcid.org/0000-0002-0410-0768>

Mike Christian Sousa Araujo – IFPI/Campus Angical – mike@ifpi.edu.br -
<https://orcid.org/0000-0001-7604-9159>

Eduardo Henrique da Silva Aranha – UFRN/PPgSC – eduardoaranha@dimap.ufrn.br -
<https://orcid.org/0000-0002-8446-638X>

Resumo. *Ensino de programação é uma tarefa difícil, devido à natureza complexa do assunto, dos estereótipos associados e das disciplinas de programação muitas vezes não conseguirem estimular adequadamente o aprendizado do aluno. Para ajudar o desenvolvimento de pesquisas nessa área, este estudo tem como objetivo mapear as publicações nacionais sobre o ensino e aprendizagem de programação. Foi realizado um mapeamento sistemático da literatura reunindo os resultados provenientes de 390 estudos publicados nos principais eventos e revistas nacionais da área. Observou-se uma grande aplicação de estudos na educação básica e a preferência no uso de ferramentas como o Scratch para auxiliar o processo de ensino e aprendizagem de programação. Além disso, percebe-se uma baixa empregabilidade de teorias de aprendizagem e de técnicas de coletas de dados nos estudos reportados, e carência de avaliações que não sejam do tipo estudo de caso. Também foi verificada uma carência de pesquisas voltadas para tecnologias voltadas a alunos com necessidades especiais, sendo assim uma lacuna que pode ser preenchida por novos pesquisadores que se interessem por esse tema.*

Palavras-chave: *Programação, Ensino, Aprendizagem, Mapeamento Sistemático da Literatura.*

A Systematic Mapping on Programming Teaching and Learning

Abstract. *Programming teaching is a difficult task, due to the complex nature of the subject, the associated stereotypes and programming disciplines often fail to adequately stimulate student learning. To help the development of research in this area, this study aims to map national publications on teaching and learning programming. A systematic mapping of the literature was carried out, gathering the results from 390 studies published in the main national events and magazines in the area. There was a great application of studies in basic education and the preference in the use of tools such as Scratch to assist the teaching and learning process of programming. In addition, there is a low employability of learning theories and data collection techniques in the reported studies, and a lack of evaluations that are not of the case study type. There was also a lack of research focused on technologies aimed at students with special needs, thus a gap that can be filled by new researchers who are interested in this topic.*

Keywords: *Programming, Teaching, Learning, Systematic Mapping of Literature.*

1. Introdução

Várias são as dificuldades enfrentadas pelos alunos durante o processo de aprendizagem de programação. Dentre elas, é possível citar o raciocínio lógico-matemático encontrado no uso de ponteiros, recursão, estruturas condicionais e de repetição. Conforme Silva, Medeiros e Aranha (2015), aprender programar tem sido destacado pelos iniciantes como uma tarefa difícil e complexa. Devido a essas dificuldades enfrentadas por alunos iniciantes em cursos das áreas tecnológicas, os resultados observados são altas taxas de retenção e de evasão (SOUZA, SILVA e BARBOSA, 2016).

Estudos têm sido realizados por pesquisadores para tentar superar as dificuldades encontradas pelos alunos. Dentre as propostas, encontram-se, por exemplo, o uso de jogos, robótica, ferramentas e metodologias. Os resultados desses estudos indicam que os alunos se sentem mais empenhados e motivados por essa nova experiência de aprendizagem, produzindo níveis mais altos de engajamento do estudante, melhorias em relação ao material de aprendizagem e maiores taxas de sucesso de aprendizagem comparado aos métodos tradicionais de ensino (SILVA *et. al.*, 2015).

Sendo assim, deve-se promover e fundamentar novas pesquisas que insiram de forma efetiva o ensino de programação nos diferentes níveis de ensino. Para isso, é importante o conhecimento dos trabalhos existentes e das experiências reportadas, as quais apresentam as características e desafios encontrados no cenário do ensino nacional. Por essa razão, este trabalho investiga a literatura brasileira sobre a utilização de abordagens – jogos, ferramentas, robóticas e outras – no ensino e aprendizagem de programação. Para isto, este trabalho foi organizado como um Mapeamento Sistemático da Literatura (MSL), analisando 12 congressos e 5 revistas relevantes nas áreas de Computação e Informática na Educação.

Para apresentar o trabalho realizado, este artigo segue organizado em seis seções além desta introdutória. A Seção 2 apresenta trabalhos correlatos e a Seção 3 a metodologia utilizada. Logo após, a Seção 4 apresenta os resultados do mapeamento realizado, sendo as ameaças à validade do estudo apresentadas na Seção 5. Por fim, na Seção 6 são feitas as considerações finais desse trabalho.

2. Estudos Correlatos

Silva *et. al.*, (2015) desenvolveram uma Revisão Sistemática da Literatura (RSL) sobre às abordagens para o ensino e aprendizagem de programação publicados entre 2009 a 2013. Os resultados mostraram uma tendência no desenvolvimento de ferramentas para o ensino de algoritmos em cursos superiores. Em Holanda, Freire e Continho (2019) é apresentado uma RSL referente às estratégias para o ensino e aprendizagem de programação para iniciantes no ensino superior, publicados nos anos de 2014 a 2018. Da mesma forma, Blatt, Becker e Ferreira (2017) através de um MSL investigou as metodologias e ferramentas de apoio para o ensino de programação.

Esta pesquisa difere-se, das citadas anteriormente, ao selecionar e analisar trabalhos que apresentam ferramentas, jogos, metodologias, técnicas e robótica relacionadas ao ensino de programação em todos os níveis de ensino, publicados entre 2014 a 2020, além de analisar uma grande base de dados (eventos e revistas) nacionais. Este estudo também se diferencia dos outros por eles não abordarem o uso de tecnologias assistivas.

3. Mapeamento Sistemático

Este Mapeamento Sistemático da Literatura (MSL) está estruturado com base nas diretrizes originais propostas por Petersen, Vakkalanka e Kuzniarz (2015). O objetivo é investigar a literatura brasileira observando as principais características da pesquisa nacional acerca do ensino e aprendizagem de programação. O período considerado por essa investigação consiste nos anos de 2014 a 2020.

3.1. Questões de Pesquisa

Para atingir o objetivo definido, esse mapeamento responde as seguintes Questões de Pesquisa (QP):

- **QP1:** O que está sendo proposto (metodologia, ferramenta, jogo, robótica, software, aplicativo e outras) para o ensino de programação, e visando qual público-alvo?
- **QP2:** Que tipos de estudos estão sendo realizados por pesquisadores que investigam o processo de ensino e aprendizagem de programação?
- **QP3:** Quais as técnicas de coleta e análise de dados utilizadas nos estudos?
- **QP4:** Quais benefícios e limitações estão sendo relatados pelo uso das abordagens propostas?
- **QP5:** Os estudos desenvolvidos suportam quais tipos de Tecnologia Assistiva?

As questões QP1 e QP4 são importantes para auxiliar pesquisadores a propor novas tecnologias e a comparar com as existentes. Já as questões QP2 e QP3 podem identificar oportunidades de melhoria no processo de pesquisa nacional. Por fim, a QP5 foi inserida por acreditar-se que a educação deve ser inclusiva, acessível a todos, e que muito ainda tem que ser feito para se atingir esse objetivo.

3.2. Processo de busca e seleção de artigos

O processo de pesquisa escolhido foi a busca manual e exaustiva dos anais das conferências e revistas mais relevantes na área. Desta forma, as seguintes conferências foram analisadas: *Computer and Beach* (COTB); *Congresso Internacional de Informática Educativa* (TISE); *Encontro Nacional de Computação dos Institutos Federais* (ENCompIF); *Seminário Jogos Eletrônicos, Educação e Comunicação* (SJEEC); *Simpósio Brasileiro de Informática na Educação* (SBIE); *Simpósio Brasileiro de Jogos e Entretenimento Digital* (SBGAMES); *Simpósio Brasileiro de Sistemas de Informação* (SBSI); *Simpósio Brasileiro sobre Fatores Humanos em Sistemas* (IHC); *Workshop do Congresso Brasileiro de Informática na Educação* (WCBIE); *Workshop de Desafios da Computação Aplicada à Educação* (DesafIE); *Workshop de Educação em Computação* (WEI) e *Workshop de Informática na Escola* (WIE).

Além disso, este estudo considera os artigos publicados nas seguintes revistas: *Revista Informática na Educação: Teoria e Prática* (RIETP); *Revista Brasileiro de Informática na Educação* (RBIE); *Revista Brasileira de Computação Aplicada* (RBCA); *Revista de Informática Teórica e Aplicada* (RITA) e *Revista Novas Tecnologias na Educação* (RENOTE). Os eventos e revistas considerados neste trabalho são importantes bases nacionais de pesquisa na área de informática educacional e incluem tanto trabalhos teóricos como práticos, além de terem sido usados em outros MSLs, tais como os de Blatt, Becker e Ferreira (2017) e de Souza, Batista e Barbosa (2016).

Uma abordagem baseada em três etapas foi utilizada no processo de busca, em um total de 586 analisados. Na primeira etapa, todos os estudos primários recuperados foram avaliados a fim de se identificar aqueles relevantes para se responder às questões

de pesquisa. Após a leitura dos títulos, resumos e palavras chaves, este conjunto inicial foi reduzido para 425. Durante esta triagem, foram aplicados Critérios de Inclusão (CI) e Critérios de Exclusão (CE) para cada estudo analisado (ver Tabela 1).

Tabela 1 – Critérios de Inclusão e Exclusão.

Critérios de Inclusão	Critérios de Exclusão
CI1: Publicações que relatam o uso de abordagens para o ensino e aprendizagem de programação; CI2: No caso de evoluções de um mesmo trabalho, incluir apenas o artigo mais recente; CI3: Publicado entre 2014 a 2020.	CE1: Publicações cujo foco principal não é o uso de abordagens para o ensino e aprendizagem de programação; CE2: Documentos publicados na forma de resumos ou apresentações, ou de estudos secundários (ou seja, revisões sistemáticas da literatura e mapeamentos sistemáticos); CE3: Trabalhos cujo texto completo não esteja disponível gratuitamente na Web; e CE4: Artigos duplicados.

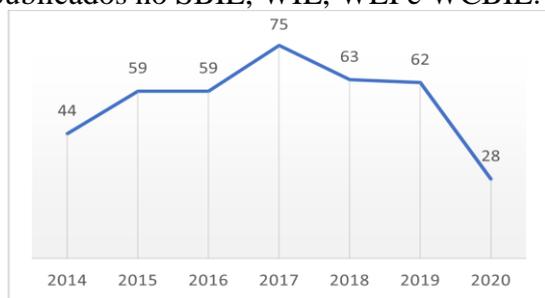
Em uma segunda etapa, procedeu-se à leitura dos resumos e conclusão, e novamente à aplicação dos CI e CE nos 425 artigos candidatos. O resultado foi um subconjunto de 390 estudos primários. Na terceira e última etapa, os 390 estudos que compõe a seleção final foram lidos na íntegra. Os dados de identificação dos artigos incluídos neste MSL podem ser visualizados em: < encurtador.com.br/coIK9 >.

3.3. Processo de extração de dados

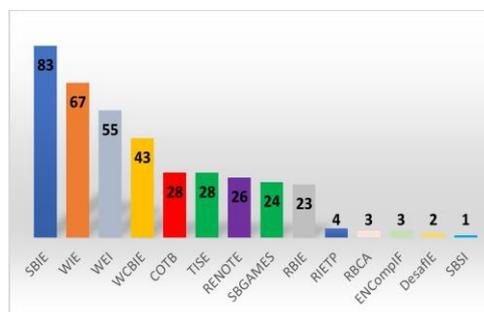
Durante a leitura de cada artigo, foram extraídos os seguintes dados: objetivo do estudo; tipo do estudo (ex., experimento controlado, estudo de caso, entre outros); abordagens utilizadas; forma de coleta e análise dos dados do estudo; benefícios, conclusões e dificuldades encontradas.

4. Resultados e Discussões

Como resultado da busca baseadas em títulos, resumos e palavras-chaves, de um total de 586 artigos, foram selecionados 390 estudos. Analisando a Figura 1 (a), pode-se observar um número significativo e consistente de publicações ao longo dos anos, mostrando que a comunidade acadêmica está ativa nesta área. Apenas o ano de 2020 foi atípico, provavelmente devido ao fechamento das escolas durante a pandemia do coronavírus. Além disso, através da Figura 1 (b), verifica-se uma maior concentração de trabalhos publicados no SBIE, WIE, WEI e WCBIE.



(a) Evolução das publicações.



(b) Quantidades de estudos por eventos/revistas.

Figura 1: (a) Evolução das publicações e (b) quantidade de estudos por eventos/revistas.

A seguir, cada uma das questões de pesquisa deste MSL é respondida de acordo com as informações extraídas dos estudos primários. Um resumo dessas informações pode ser visto em: <encurtador.com.br/hlGM2>.

QP1: O que está sendo proposto (metodologia, ferramenta, jogo, robótica, software, aplicativo e outras) para o ensino de programação, e visando qual público-alvo?

Esta questão tem como objetivo apresentar as principais abordagens presentes nos estudos. Sendo assim, foram definidas categorias que pudessem melhor organizá-las, tomando-se como base a proposta de Silva *et al.* (2015). Desta forma, as estratégias encontradas são apresentadas na Tabela 2, sendo classificada em 5 tópicos: Ferramentas de Software, Jogos, Metodologias, Robótica e Outras abordagens.

Tabela 2 – Visão Geral das Abordagens.

Abordagens	Descrição	QTD	%
Ferramentas de Software	Uso de Scratch (36); Ferramenta para auxiliar o ensino de programação (28); Ambiente de ensino de programação (17); Juiz Online (18); Ferramenta de personalização e acompanhamento da resolução de exercícios (9); App Inventor (9); Ferramenta de programação para deficientes visuais e surdos (6); Ferramenta gamificada para ensino de programação (6); Tutor virtual inteligente (6); <i>Framework</i> gamificado (5); Softwares educacionais (3); Outras ferramentas (16).	159	41%
Jogos	Jogos digitais (54); Jogos não digitais (6); Jogos sérios digitais (16); Jogos sérios não digitais (2).	78	21%
Metodologias	Pensamento Computacional (11); Gamificação (6); Uso de DOJOs (5); Objetos de aprendizagem (4); Computação Desplugada (3); Metodologias ativas (3); Sala de aula invertida (2); Ensino de Programação com Neurociência (1); outras metodologias (33).	68	17%
Robótica	Uso da robótica	41	10%
Outras Estratégias	Oficinas de Ensino de Programação (15); Relato de Experiências (11); Mineração de dados (5); Redes Sociais (2); Redes Bayesianas (2); Arquitetura para Ecossistemas de Software (1); Arquitetura Pedagógica (1); Linguagem de Programação (1); Transposição didática (1); Padrão pedagógico (1); Portfólio (1); Animações (1); Portal Web (1); Detecção automática de evidências (1).	44	11%
Total		390	100%

Nos dados apresentados, verificaram-se estudos que utilizam mais de uma abordagem, como a utilização conjunta de Pensamento Computacional com Scratch, ou Jogos e realidade virtual, robótica e jogos, entre outras. Sendo assim, cada estudo pode ser contabilizado em mais de uma abordagem.

Pode-se notar que a maioria dos estudos (159 – 41%) foram agrupados na categoria Ferramentas de Softwares. A ferramenta Scratch foi a que mais apareceu em uso pelos trabalhos. Isso se deve, em parte, por se tratar de uma ferramenta que oferece uma interface de fácil utilização e que não envolve uma grande complexidade para sua manipulação. Estudos como o [E362] sugere que o “*Scratch pode ser uma alternativa para os alunos que nunca tiveram contato com programação de computadores tenham*

uma primeira experiência menos traumática com uma linguagem de programação desmistificado seu pensamento original de que programar é uma tarefa difícil”.

Destacamos ainda que 21% dos estudos utilizaram Jogos, dentre os quais os digitais são a maioria. Também foi percebido um número relevante de artigos que abordaram o uso de Jogos Sérios (JS) para o processo de ensino e aprendizagem de programação. JS são jogos para ensinar, treinar ou promover hábitos saudáveis e mudança social (RITTERFIELD, CODY e VORDERER, 2009). Cabe ressaltar, que 72% dos alunos que utilizam jogos digitais para aprender de programação fazem isso usando jogos já existentes, como, Lightbot Jr, The Foos e Robot School [E348]. Os outros 28% aprendem programando seus jogos, como destacado em [E22]. Embora pouco frequente, alguns trabalhos fazem uso de jogos não digitais. No estudo [E333], por exemplo, é relatada a experiência do desenvolvimento de uma oficina de jogos de tabuleiros para melhorar o desempenho dos alunos nas disciplinas básicas de programação. Segundo os autores, ocorreu melhoria do desempenho dos alunos após a realização das oficinas. Desta forma, isto pode indicar um espaço de pesquisa que pode ser melhor explorado pela comunidade acadêmica.

Em termos da categoria Metodologias, que englobou 17% dos estudos, destaca-se a aplicação do Pensamento Computacional (PC). O PC consiste no conjunto de habilidades e atitudes vinculadas na realização de uma ação, cujo principal objetivo é o da resolução de problemas, sendo assim relevante para todas as pessoas (CAVALCANTE, COSTA e ARAUJO, 2016). Isso é exemplificado pelos resultados do estudo [E97], que analisou o efeito do PC no desempenho de estudantes na educação básica. Pode ser verificado uma correlação moderada entre o desempenho dos alunos no ENEM e suas habilidades em programação, e que alunos proficientes em programação tem médias estatisticamente melhores em cada uma das áreas abordadas no ENEM.

O emprego da Robótica correspondeu a 10%. Como reportado em [E143], o uso da robótica beneficiou de forma lúdica o processo de aprendizado de programação e melhora o desempenho em termos de notas e aprovações dos alunos. Outras abordagens foram citadas em 11% dos estudos. Dentre essas, destacaram-se oficinas de ensino de programação e relato de experiências. No estudo [E246], através de uma oficina com alunos do ensino médio, foram apresentados os benefícios de se aprender programação, desenvolvendo o raciocínio lógico e estimulando o interesse pela computação dos alunos do ensino fundamental II.

Quanto ao público-alvo da proposta, verificou-se que 38% dos artigos abordaram programação no ensino médio. Já outros 19% apresentaram trabalhos no contexto do nível superior e 7% reportam estudos aplicados no ensino médio integrado ao técnico em informática. Outros 7% dos artigos destinam-se ao ensino fundamental I e 15% ao ensino fundamental II. Somente 3% das pesquisas destinam-se ao ensino infantil. Este mapeamento ainda apresenta que 11% dos artigos se aplicam a qualquer nível de ensino.

Nota-se ainda que a maioria dos estudos estão sendo aplicados no contexto da educação básica. Esses dados evidenciam a política de incentivo ao ensino de programação na educação básica. Esse resultado corrobora com CSTA (2011), o qual afirma que a introdução de conceitos de computação na educação básica, enquanto ciência, é de fundamental importância.

QP2: Que tipos de estudos estão sendo realizados por pesquisadores que investigam o processo de ensino e aprendizagem de programação?

No que diz respeito aos tipos de estudos que estão sendo realizados, foram encontrados quatro tipos diferentes: Estudo de Caso, Experimento Controlado, Quasi-experimentos e Pesquisa-ação. A utilização de estudo de caso foi o método mais usado (127 / 32%). Estudos de caso visam observar um atributo específico e estabelecer o relacionamento entre atributos diferentes (TRAVASSOS, GUROV e AMARAL, 2002). Um exemplo é o trabalho de [E8], cujo objetivo foi ensinar e estimular programação de jogos digitais no ensino médio. O método de Pesquisa-ação, segundo Cruz, Santos e Andrade (2005) é um processo interativo entre os pesquisadores e os participantes que atuam juntos em ciclos de atividades particulares, incluindo diagnóstico de problemas, ação de intervenção e aprendizagem reflexiva. O uso deste método foi reportado em 7 estudos (4% do total de artigos), como destacado no estudo [E354].

A utilização de experimentos controlados foi reportada em 3% dos estudos (5 artigos), sendo utilizados grupos de controle para verificar a eficácia da abordagem em relação ao aprendizado. Para Travassos, Gurov e Amaral (2002), o objetivo do experimento é manipular uma ou mais variáveis e manter as outras fixas medindo o efeito do resultado, que no contexto dos estudos analisados eram o efetivo aprendizado de conceitos de programação. No estudo [E22] é apresentado os resultados de experimento controlado utilizando o *design* em quadrado latino de formatos de videoaulas para ensinar programação de jogos digitais no ensino médio. Além disso, 3 estudos (1% do total) aplicaram quasi-experimento, que é um estudo empírico semelhante a um experimento controlado, só que a atribuição de tratamentos não é baseada em aleatorização (WOHLIN *et. al.* 2000). Em [E115] é apresentado um exemplo desse tipo de estudo. Por fim, é importante salientar que em 60% dos estudos (248 artigos) não foram reportados os tipos de estudos utilizados, sendo um ponto que precisa ser melhor observado pela comunidade acadêmica.

QP3: Quais as técnicas de coleta e análise de dados utilizadas nos estudos?

Em relação à técnica de coleta e análise de dados, na maioria dos estudos (201 / 51%) foram utilizados questionários com a técnica principal para coleta os dados. Nestes casos, geralmente foram utilizadas perguntas para se avaliar a opinião dos participantes em relação aspectos de cada estudo. Por exemplo, para identificar a motivação em participar do estudo e aspectos relacionados a aprendizagem. Outra técnica utilizada foi a observação, que é realizada *in loco* no momento da execução dos estudos empregada para analisar o comportamento dos participantes. Esta foi utilizada em 22% (88) dos artigos. Em contrapartida, 17 trabalhos (6%) relatam a utilização de entrevistas para se obter *feedback*. A análise de *logs* de ferramentas foi utilizada em 11 estudos (4%) para verificar por exemplo os erros frequentemente cometidos pelos alunos. Além disso, métricas de completude, análise multivariada e *clustering* ilustram o percurso de aprendizagem percorrido pelos participantes. O baixo percentual verificado sugere que muitos pesquisadores não estão fazendo uso de *log* para a fundamentação de seus resultados, baseando-se demasiadamente na opinião dos participantes.

A estatística descrita e inferencial foi utilizada em 10 estudos (5%) para descrever e sumarizar os resultados com a ajuda testes de hipóteses, como Mann-Whitney, e com análise de correlação. Já a estatística multivariada foi utilizada em 4 estudos (4%). A análise de conteúdo foi aplicada em 3 estudos, correspondendo a 2% do total. Por fim, outras técnicas utilizadas foram: Grupo focal (1), Análise de discurso (1) e Métricas da rede social (1), correspondendo a 3%. Ressaltamos que um estudo pode apresentar mais de uma técnica de análise de dados, por exemplo, o caso de questionário e entrevista, empregados no estudo [E12], e análise de *logs* e observação, aplicado no estudo [E41].

É importante ressaltar que 8% dos estudos não empregaram nenhuma técnica de coleta de dados. Isso é observado principalmente nos estudos que são ainda propostas ou que não foram testados em sala de aula ou ambiente propício. De fato, a falta de aplicação de instrumentos validados e confiáveis na avaliação em pesquisas brasileiras é uma situação que não colabora para a consolidação dessas abordagens no ensino de programação. Esse fato é observado também por outros estudos, como o apresentado em Silva *et. al.* (2018).

QP4: Quais benefícios e limitações estão sendo relatados pelo uso das abordagens propostas?

Através da leitura dos artigos selecionados, alguns benefícios e limitações podem ser observados. Estudos comentam sobre as dificuldades dos estudantes em abstrair a aprendizagem de programação e outros argumentam a falta de motivação para a realização de atividades e exercícios de programação. Por exemplo, o estudo [E15] diz que *“a maior dificuldade encontrada refere-se a questões relacionadas à capacidade de abstração, como a utilização da noção de tempo e de laços de repetição, e a construção mental da sequência de ações necessárias para a obtenção de um determinado resultado e sua posterior materialização na forma de um programa de computador”*. Já o estudo [E17] destaca como limitações manter a atenção e o foco no conteúdo, gerenciar tempo de aula e acompanhar o crescimento cognitivo em diferentes ritmos. Alguns estudos como o [E55] destacam ainda a falta de infraestrutura das escolas, como por exemplo laboratórios de informática defasados. Outras limitações encontradas nos estudos são o viés de seleção, a falta de grupo de controle e instrumentos de avaliação insuficientes para apontar evidências sobre a aprendizagem destacada nos estudos.

Com relação aos benefícios relatados, destacam-se nos estudos: estímulo ao raciocínio lógico e a resolução de problemas [E254 e E262]; seleção e organização de informações, podendo ajudar os alunos em disciplinas como matemática e em atividades presentes do seu cotidiano. Além disso, pode ajudar no desenvolvimento de habilidades como as de: sequenciar eventos ou ideias [E294]; realizar operações mentais a partir de experiências concretas [E353]; criatividade [E262], cognição e trabalho em equipe [E39]. Por fim, foi possível observar trabalhos abordando a melhora nos índices de aprovação, como apresentado nos estudos [E115, E120 e E138].

QP5: Os estudos desenvolvidos suportam quais tipos de Tecnologia Assistiva?

Dos estudos analisados, somente 9 estudos (2%) apresentaram suporte para pessoas com algum tipo de necessidade especial. Seis estudos [E48, E49, E69, E80, E147 e E215] desenvolveram pesquisas para pessoas com Deficiência Auditiva (Surdos). Como exemplo, destacamos o estudo [E80] que desenvolveu uma linguagem de programação baseada em LIBRAS para o ensino de lógica de programação para deficientes auditivos. Além disso, dois estudos [E44][E48] focam na Deficiência Visual. O estudo [E44], por exemplo, teve como objetivo criar e avaliar o uso de uma linguagem de programação chamada GoDonnie para alunos com deficiência visual. Por fim, o estudo [E215] desenvolveu um objeto de aprendizagem para o ensino de programação para deficientes físicos e auditivos através do Estímulo Visual. Considerando a quantidade e demanda de alunos com necessidades especiais, é interessante que sejam propostas mais pesquisas com esse foco.

5. Ameaças do estudo

No geral, as principais ameaças à validade são em relação a viés na seleção de publicações e a extração e análise dos dados. Como forma de mitigar esses riscos, este mapeamento

considerou a busca manual e exaustiva nos anais dos principais eventos e revistas nacionais da área. Mesmo assim, não é possível garantir que todos os estudos relevantes para o tema foram publicados nos locais analisados. O processo manual de extração de dados dos artigos também pode resultar em erros, sendo assim uma ameaça natural a esse tipo de trabalho. O desenvolvimento de um protocolo do MSL e o uso de uma estratégia de verificação da qualidade da revisão (por um segundo revisor) ajudou a garantir que este não foi o caso. Finalmente, é possível que estudos possam ter sido excluídos por terem títulos ou resumos mal elaborados, que não passassem completamente a ideia do trabalho. Para mitigar esse risco, em caso de dúvida, os artigos não eram excluídos sem a leitura completa do texto.

6. Considerações Finais

Esse artigo apresentou um MSL que permitiu analisar as pesquisas brasileiras com relação às abordagens utilizadas no ensino e aprendizagem de programação. O mapeamento se baseou em publicações dos principais veículos nacionais na área de Informática na Educação e Computação.

Em relação ao ensino de programação, percebeu predominância da aplicação de ferramentas. Dentre essas, observa-se que o tipo mais utilizado são as de apoio a programação, sendo o Scratch a ferramenta mais citada. É observada uma maior publicação de artigos no contexto da educação básica. Isso se deve a iniciativas que trabalham o ensino de computação desde as séries iniciais.

Outros resultados e tendências, no que diz respeito ao ensino e aprendizagem de programação usando, poderão ser indicados como: os estudos em maioria (87% das pesquisas, totalizando 341 artigos) não se preocuparam com adotar ou pelo menos reportar teorias de aprendizagem que fundamentassem o desenvolvimento das pesquisas; estudos de casos é tipo de mais comumente relatados e utilizados para avaliar os resultados dos estudos; questionários é a técnica principal para coleta de dados; os estudos como suporte para pessoas com algum tipo de necessidade especial são ainda incipientes; O SBIE é o evento que apresenta a maior quantidade de estudos para a temática em questão; iniciativas de ensino de programação voltadas para mulheres tem aumentado, com o objetivo de encorajar a inserção das mulheres na área de tecnologias.

Os estudos primários analisados são uma importante fonte de informações sobre as diversas mudanças enfrentadas nessa área ao longo dos anos. Mesmo assim, esta revisão corrobora com Scaico e Queiroz (2013), que mesmo mediante diversas iniciativas, as pesquisas acerca do processo de ensino e aprendizagem detêm evidências tímidas sobre habilidades cognitivas que os métodos atuais desenvolvem, assegurando que o estudante seja capaz de resolver desafios além muros escolares. Isso se deve em partes ainda a poucos resultados que demonstrar a eficácia das aplicações das abordagens, a baixa aplicação de métodos empíricos de pesquisas e a falta de abordagens teorias/pedagógicas nos estudos.

Portanto, através dos resultados deste mapeamento, este artigo contribuiu com uma visão geral do panorama nacional capaz de servir de ponto de partida para várias outras pesquisas, incentivando principalmente que novos estudos ataquem os pontos fracos observados e destacados neste mapeamento sistemático.

Agradecimentos

Os autores agradecem a Fundação de Amparo à Pesquisa do Maranhão (FAPEMA) pelo suporte a esta pesquisa, através do Edital FAPEMA N° 010/2019 – Geração Ciência e, em especial, ao IFMA – Campus São João dos Patos, por toda infraestrutura oferecida.

Referências

- Blatt, L.; Becker, V.; Ferreira, A. M. S. (2017) “Mapeamento Sistemático sobre Metodologias e Ferramentas de apoio para o Ensino de Programação”. In: Workshop de Informática na Escola – WIE, 2017.
- Cavalcante, A.; Costa, L. S.; Araujo, A. L. (2016) “Um Estudo de Caso Sobre Competências do Pensamento Computacional Estimuladas na Programação em Blocos no Code.Org”. In: Anais dos Workshops do V Congresso Brasileiro de Informática na Educação.
- Cruz, C. C. P.; Santos, C. F.; Andrade, L. C. V. (2015) “Plataforma Interativa para Internet (PII): Resultados de uma Proposta de Pesquisa-Ação”. In: Simpósio Brasileiro de Informática na Educação – SBIE, 2005.
- CSTA – Computer Science Teacher Association. (2011) “CSTA K-12 Computer Science Standards”. CSTA Standards Task Force. ACM – Association for Computing Machinery.
- Holanda, W. D.; Freire, L. P.; Coutinho, J. C. S. (2019) “Estratégias de ensino-aprendizagem de programação introdutória no ensino superior: uma Revisão Sistemática da Literatura”. In: Revista Novas Tecnologias na Educação – RENOTE.
- Petersen, K.; Vakkalanka, S.; Kuzniarz, L. (2015) “Guidelines for conducting systematic mapping studies in software engineering: An update”. In: Information and Software Technology, v. 64, pp. 1–18, Elsevier.
- Ritterfield, U.; Cody, M.; Vorderer, P. (Ed.). (2009) “Serious Games: Mechanisms and Effects”. Nova Iorque: Routledge.
- Scaico, P. D.; de Queiroz, R. J. G. B. (2013) “A educação do futuro: uma reflexão sobre aprendizagem na era digital”. In: Simpósio Brasileiro de Informática na Educação – SBIE, Campinas – SP.
- Silva, T. R.; Medeiros, T. J.; Aranha, E. H. S. (2015) “Jogos Digitais para Ensino e Aprendizagem de Programação: uma Revisão Sistemática da Literatura”. In: XXV Simpósio Brasileiro de Informática na Educação – SBIE.
- Silva, T. R.; Medeiros, T. J.; Medeiros, H.; Lopes, R.; Aranha, E. (2015) “Ensino-aprendizagem de programação: uma revisão sistemática da literatura”. In: Revista Brasileira de Informática na Educação. v. 23, n. 1, 2015.
- Silva, T. R.; Cordeiro, J. R.; Santos, R. S. F.; Santos, F. G.; Aranha, E. H. S.; Silva, F. G. (2018) “Uma Análise do Cenário Nacional do Uso de Jogos para o Ensino e Aprendizagem de Computação”. In: Simpósio Brasileiro de Jogos e Entretenimento Digital – SBGAMES, 2018.
- Souza, D. M.; Batista, M. H. S.; Barbosa, E. F. (2016) “Problemas e Dificuldades no Ensino e na Aprendizagem de Programação: Um Mapeamento Sistemático”. In: Revista Brasileira de Informática na Educação – RBIE.
- Travassos, G.; Gurov, D.; Amaral, E. (2002) “Introdução à Engenharia de Software Experimental”. In: Relatório Técnico ES-590/02-Abril, Programa de Engenharia de Sistemas e Computação, COPPE/UFRJ.
- Wohlin, C.; Runeson, P.; Host, M.; Ohlsson, C.; Regnell, B.; Wesslén, A. (2000) “Experimentation in Software Engineering: an Introduction”. Kluwer Academic Publishers.