



Ferramentas de Apoio ao Ensino Introdutório de Programação: um Mapeamento Sistemático

Diego Eugenio F. Pereira, Universidade Federal de Itajubá, diegoefp_90@hotmail.com

Rodrigo Duarte Seabra, Universidade Federal de Itajubá, rodrigo@unifei.edu.br

Adler Diniz de Souza, Universidade Federal de Itajubá, adlerunifei@gmail.com

Resumo: Ao longo dos anos, várias ferramentas foram propostas na literatura visando auxiliar e apoiar o ensino de programação, sobretudo em disciplinas introdutórias. Essas ferramentas podem auxiliar o aprendizado dos estudantes ao mesmo tempo em que proporcionam dinamismo e inovação, contribuindo para o aumento da motivação no estudo do tema tratado e, por conseguinte, reduzindo os índices de evasão. Esta pesquisa visa analisar ferramentas educacionais de apoio ao ensino introdutório de programação por meio de um mapeamento sistemático da literatura. A principal contribuição do estudo consistiu na apresentação de uma classificação das ferramentas levantadas em nove categorias, podendo auxiliar docentes e discentes na busca de recursos mais adequados às suas necessidades.

Palavras-chave: ensino de programação, algoritmos, cursos introdutórios, ferramentas.

Support Tools for Programming Introductory Teaching: a Systematic Mapping

Abstract: Over the years, several tools have been proposed in the literature to assist and to support the teaching of programming, especially in introductory courses. These tools can help students' learning while providing dynamism and innovation, contributing to increasing motivation for studying the topic addressed and, therefore, reducing dropout rates. This research aims to analyze educational tools to support the introductory programming teaching through a systematic mapping of the literature. The main contribution of the study consisted in presenting a classification of the tools raised into nine categories, being able to assist teachers and students in the search for more adequate resources to their needs.

Keywords: teaching programming, algorithms, introductory courses, tools.

1. INTRODUÇÃO

Disciplinas introdutórias de programação são encaradas como um grande desafio por professores e estudantes. Vários são os motivos que tornam difícil o ensino e o aprendizado do tema, sendo a base dos cursos de computação no Brasil de acordo com o currículo de referência da Sociedade Brasileira de Computação (SBC, 2005), e também no mundo. Para Rocha *et al.* (2010), essas disciplinas possuem uma alta evasão, representando um obstáculo nos cursos de Tecnologia da Informação e afins. Segundo Salleh *et al.* (2013), a programação é um processo que envolve atividades como planejamento, *design*, testes e depuração, e seu aprendizado exige o entendimento de conceitos complexos e abstratos por parte dos estudantes. De Souza *et al.* (2018) defendem que um dos fatores determinantes que leva ao mau desempenho e aproveitamento dos discentes na disciplina é a metodologia aplicada em sala de aula. Dessa forma, muito se discute na literatura sobre ferramentas e metodologias de auxílio ao ensino de programação.



De acordo com Sheard *et al.* (2009), ferramentas para programação estão entre os três maiores problemas relacionados ao ensino e aprendizagem do assunto. Em seu estudo, Denner *et al.* (2012) relatam que várias ferramentas e estratégias de ensino são implementadas na preparação de estudantes de programação, por meio de contos narrativos, jogos, simulações e técnicas de visualização. De modo geral, essas ferramentas buscam melhorar o engajamento e o interesse dos estudantes frente às disciplinas da área. Apesar de as pesquisas sobre o ensino e o aprendizado de programação estarem em uma crescente, ainda há espaço para sintetizar os resultados relevantes de pesquisas na área, especialmente no contexto de ferramentas de apoio ao ensino introdutório de programação. Nesse contexto, entende-se por ensino introdutório de programação a apresentação dos fundamentos de programação, resolução de problemas, estruturas de dados e lógica, assim como a construção dos primeiros algoritmos por alunos que, normalmente, não possuem experiência prévia em programação. Este trabalho buscou identificar ferramentas de apoio ao ensino de programação existentes na literatura, por meio de um mapeamento sistemático, classificando-as em categorias. O estudo visa contribuir com futuros pesquisadores, docentes e discentes na busca por ferramentas que auxiliem determinada área do ensino de programação.

O artigo está estruturado da seguinte forma: a Seção 2 apresenta a metodologia de pesquisa utilizada, assim como as questões de pesquisa e as etapas e processos executados. A Seção 3 apresenta as ferramentas identificadas. Na Seção 4 é realizada uma discussão sobre as questões de pesquisa definidas na Seção 2. Finalmente, a Seção 5 apresenta as considerações finais do estudo.

2. DEFINIÇÃO DA PESQUISA

O estudo em tela foi realizado por meio de um mapeamento sistemático da literatura. Petersen *et al.* (2015) argumentam que o mapeamento sistemático fornece uma visão geral de uma área de pesquisa para avaliar a quantidade de evidências existentes sobre um tópico de interesse, sendo indicado, segundo Wohlin (2012), quando a questão de pesquisa é ampla e se deseja obter uma visão abrangente do estado da arte ou prática sobre determinado assunto. Isso posto, a escolha do mapeamento é interessante para o objetivo desta pesquisa, que consistiu em identificar estudos que desenvolveram ou apontaram o uso de ferramentas didáticas para o ensino e aprendizado de algoritmos e programação em cursos introdutórios.

2.1 Questões de Interesse

O objetivo geral deste estudo consistiu em fornecer uma visão geral das ferramentas de apoio ao ensino e aprendizagem de programação disponíveis, catalogando-as por suas categorias. Para tanto, foram definidas três questões de pesquisa.

Q1: Quais são as ferramentas didáticas para o ensino e aprendizagem de algoritmos e lógica de programação para cursos introdutórios?

Uma vez identificadas as ferramentas, tem-se as seguintes questões:

Q2: Como a ferramenta se categoriza?

As variadas ferramentas podem ser classificadas em diferentes categorias, por exemplo: ambiente de programação visual, robôs para programação, ferramentas de correção e avaliação, dentre outras. Os nomes das categorias foram retirados dos artigos nos quais as ferramentas foram propostas ou analisadas.



Q3: Quais são os principais resultados apresentados pelos artigos selecionados? Quais as ferramentas mais promissoras?

Dentre as ferramentas de apoio ao ensino de programação, algumas possuem maior destaque ou apresentaram resultados positivos ao auxiliarem o processo educacional, a retenção de conhecimento e o aumento do engajamento dos estudantes.

2.2 Execução da Pesquisa

Para definir a pesquisa e delinear seu escopo foram estabelecidos critérios para garantir a viabilidade de sua execução, a abrangência do estudo e o acesso aos dados. A pesquisa foi realizada em 19 de junho de 2020 e conduzida a partir de bibliotecas digitais por meio de seus motores de busca. Após a identificação dos trabalhos, foi realizado um processo de *snowballing* retroativo para o levantamento de outros artigos. A pesquisa foi executada utilizando seis bases de dados: Scopus, IEEE, Compendex, ACM Digital Library, Springer Link e Science Direct. Essas bibliotecas digitais foram escolhidas por possuírem um grande acervo de artigos relacionados à Ciência da Computação, além de serem as bases mais utilizadas na área. Este estudo compreende os documentos disponíveis nessas bases de dados. A expressão final para busca em cada base de dados pode ser visualizada no Quadro 1.

Quadro 1 – Expressões de busca nas bibliotecas digitais. Fonte: Os autores.

Biblioteca	Expressão de Busca
Scopus	("algorithms and data structures tools" OR "algorithms teaching tools" OR "programming logic teaching tools" OR "programming teaching tools" OR "algorithms and data structures teaching tools") AND ("introductory programming courses" OR "introductory courses" OR "Novice" OR "introductory" OR "novice programming" OR "Teaching of Introductory Programming")
IEEE	((("All Metadata":algorithms and data structures tools) OR ("All Metadata":algorithms teaching tools) OR ("All Metadata":programming logic teaching tools) OR ("All Metadata":programming teaching tools) OR ("All Metadata":algorithms and data structures teaching tools)) AND ((("All Metadata":introductory programming courses) OR ("All Metadata":introductory courses) OR ("All Metadata":Novice) OR ("All Metadata":introductory) OR ("All Metadata":novice programming) OR ("All Metadata":Teaching of Introductory Programming)))
Compendex	("algorithms and data structures tools" OR "algorithms teaching tools" OR "programming logic teaching tools" OR "programming teaching tools" OR "algorithms and data structures teaching tools")
ACM Digital Library	[[All: "algorithms and data structures tools"] OR [All: "algorithms teaching tools"] OR [All: "programming logic teaching tools"] OR [All: "programming teaching tools"] OR [All: "algorithms and data structures teaching tools"]] AND [[All: "introductory programming courses"] OR [All: "introductory courses"] OR [All: "novice"] OR [All: "introductory"] OR [All: "novice programming"] OR [All: "teaching of introductory programming"]]
Springer Link	("algorithms and data structures tools" OR "algorithms teaching tools" OR "programming logic teaching tools" OR "programming teaching tools" OR "algorithms and data structures teaching tools") AND ("introductory programming courses" OR "introductory courses" OR "Novice" OR "introductory" OR "novice programming" OR "Teaching of Introductory Programming")
Science Direct	("algorithms and data structures tools" OR "algorithms teaching tools" OR "programming logic teaching tools" OR "programming teaching tools" OR "algorithms and data structures teaching tools")

A busca realizada com essas expressões retornou um total de 339 artigos, distribuídos entre as bibliotecas digitais da seguinte maneira: Scopus – 39 artigos; IEEE – 285 trabalhos; Compendex – 2; ACM Digital Library – 6; Springer Link – 3; e Science Direct – 4. Como nem todos os documentos encontrados estavam alinhados com a questão de pesquisa, esses artigos foram submetidos a uma triagem para que

somente os artigos relevantes, para o propósito do estudo, fossem utilizados. O *link* a seguir contém a listagem com os artigos selecionados no mapeamento sistemático realizado:

https://drive.google.com/file/d/18bjhbwsk-mWUeOhxbDke84_xDqNCAZrQ/view

O processo de triagem, representado pela Figura 1, foi conduzido de acordo com os seguintes passos: (i) foi aplicado um filtro para apenas *journals* na biblioteca IEEE, removendo resultados que referenciavam conferências, devido ao elevado número de trabalhos identificados caso este não fosse aplicado, resultando em 78 artigos; (ii) na etapa seguinte, foram removidos artigos duplicados e foi aplicado um primeiro filtro, utilizando um critério de inclusão e sete de exclusão, sendo esses últimos descritos no Quadro 2. O critério de inclusão consistiu em selecionar publicações que mencionassem ferramentas de apoio ao ensino introdutório de programação, lógica de programação e algoritmos e estruturas de dados. Os critérios foram aplicados na leitura dos títulos e *abstracts* dos documentos, resultando em 26 artigos. Finalmente, foi realizada uma seleção mais criteriosa por meio da leitura completa dos artigos remanescentes, obtendo-se 20 artigos. A eles se somaram dois trabalhos provenientes do processo de amostragem por *snowballing*, resultando em 22 artigos (Quadro 3), sendo 12 da biblioteca Scopus, quatro da IEEE, um da Compendex, um da ACM, um da Springer Link, um da Science Direct e dois oriundos do *snowballing*. Dos artigos removidos durante a última etapa, dois deles foram excluídos por apresentarem metodologias de ensino e não ferramentas, dois por tratarem de ferramentas para disciplinas avançadas de programação, um por apresentar uma ferramenta não validada e outro por apenas avaliar o ensino de programação, sem necessariamente oferecer qualquer tipo de apoio.

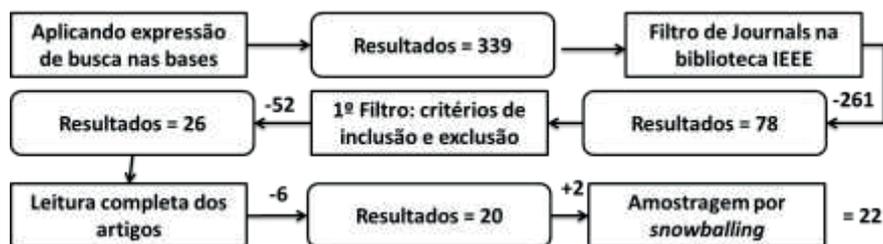


Figura 1 – Processo de triagem dos artigos. Fonte: Os autores.

Quadro 2 – Conjunto de critérios de exclusão. Fonte: Os autores.

Identificador	Descrição
CE-01	Não serão selecionadas publicações nas quais as palavras-chave não estejam presentes no título, <i>abstract</i> , <i>keywords</i> e/ou texto da publicação.
CE-02	Não serão selecionadas publicações que descrevam e/ou apresentem “ <i>keynote speeches</i> ”, tutoriais, cursos, <i>workshops</i> e similares.
CE-03	Não serão selecionadas publicações que mencionam ferramentas de apoio a programadores profissionais.
CE-04	Não serão selecionadas publicações que mencionam ferramentas de apoio ao ensino de programação para disciplinas avançadas de programação.
CE-05	Não serão selecionadas publicações que mencionam ferramentas algorítmicas de apoio ao ensino de outras disciplinas além de introdução à programação.
CE-06	Não serão selecionadas publicações que mencionam estratégias e /ou metodologias de apoio ao ensino de programação sem o uso de uma ferramenta.
CE-07	Não serão selecionadas publicações que mencionam ferramentas algorítmicas de apoio ao ensino de introdução à programação que não tenham sido validadas.

**Quadro 3** – Artigos incluídos no mapeamento sistemático. Fonte: Os autores.

Nº	Artigo	Percentil
1	AMRO, J. S.; ROMLI, R. Investigation on the learning programming techniques via mobile learning application. In: 4th International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE). IEEE, p. 1-7, 2019.	N/A
2	RESNYANSKY, D. <i>et al.</i> The potential of augmented reality for computer science education. In: IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE). IEEE, p. 350-356, 2018.	H5 Index: 11
3	ULLAH, Z. <i>et al.</i> The effect of automatic assessment on novice programming: strengths and limitations of existing systems. <i>Computer Applications in Engineering Education</i> , v. 26, n. 6, p. 2328-2341, 2018.	CiteScore: 62nd. H5 Index: 22
4	SALLEH, N. <i>et al.</i> Cloud-based learning system for improving students' programming skills and self-efficacy. <i>Journal of Information and Communication Technology</i> , v. 17, n. 4, p. 629-651, 2018.	CiteScore: 55th
5	BAUER, A. <i>et al.</i> Dragon architect: open design problems for guided learning in a creative computational thinking sandbox game. In: Proceedings of the 12th International Conference on the Foundations of Digital Games. p. 1-6, 2017.	CiteScore: 12nd
6	KYFONIDIS, C. <i>et al.</i> Block-C: A block-based programming teaching tool to facilitate introductory C programming courses. In: IEEE Global Engineering Education Conference (EDUCON). IEEE, p. 570-579, 2017.	H5 Index: 22
7	SOUZA, D. M. <i>et al.</i> A systematic literature review of assessment tools for programming assignments. In: 29th International Conference on Software Engineering Education and Training (CSEET). IEEE, p. 147-156, 2016.	Qualis B2
8	GARCIA, P. G. F.; DE LA ROSA, F. RoBlock–web app for programming learning. <i>International Journal of Emerging Technologies in Learning (iJET)</i> , v. 11, n. 12, p. 45-53, 2016.	CiteScore: 60th. H5 Index: 24
9	KOORSSE, M. <i>et al.</i> Programming assistance tools to support the learning of IT programming in South African secondary schools. <i>Computers & Education</i> , v. 82, p. 162-178, 2015.	CiteScore: 97th
10	MAJOR, L. <i>et al.</i> The effectiveness of simulated robots for supporting the learning of introductory programming: a multi-case case study. <i>Computer Science Education</i> , v. 24, n. 2-3, p. 193-228, 2014.	CiteScore: 77th
11	CHOU, C.; SUN, P. An educational tool for visualizing students' program tracing processes. <i>Computer Applications in Engineering Education</i> , v. 21, n. 3, p. 432-438, 2013.	CiteScore: 62nd. H5 Index: 22
12	CAMP, O.; WOODWARD, R. ICT Enhanced learning experience for an introductory object oriented programming course-A case study. In: International Conference on Computer Supported Education. SCITEPRESS, p. 16-25, 2011.	Qualis B4. H5 Index: 12
13	MAJOR, L. <i>et al.</i> Systematic literature review: teaching novices programming using robots. <i>IET Software</i> , v. 6, n. 6, p. 502-513, 2012.	Qualis B4. H5 Index: 20
14	SILVA-MACEDA, G. <i>et al.</i> More time or better tools? A large-scale retrospective comparison of pedagogical approaches to teach programming. <i>IEEE Transactions on Education</i> , v. 59, n. 4, p. 274-281, 2016.	H5 Index: 27
15	MARTÍNEZ, L. G. <i>et al.</i> Using a mobile platform for teaching and learning object oriented programming. <i>IEEE Latin America Transactions</i> , v. 16, n. 6, p. 1825-1830, 2018.	CiteScore: 61st. H5 Index: 25
16	OLIER, A. J. <i>et al.</i> Design and implementation of a teaching tool for introduction to object oriented programming. <i>IEEE Latin America Transactions</i> , v. 15, n. 1, p. 97, 2017.	CiteScore: 61st. H5 Index: 25
17	JÚNIOR, A. M. <i>et al.</i> Programming teaching tools feature assessment associated with Brazilian curriculum base obtained through BPL platforms analysis. In: International Conference on Human-Computer Interaction. Springer, Cham, p. 380-389, 2019.	Qualis: B2. H5 Index: 17
18	PAPADAKIS, S. <i>et al.</i> Novice programming environments. Scratch & app inventor: a first comparison. In: Proceedings of the 2014 Workshop on Interaction Design in Educational Environments. p. 1-7, 2014.	CiteScore: 12nd
19	AL-BAYATY, A.; MARTINEZ, C. CLIP 4 Robotics: a click-based programming language. In: International Conference on Human-Computer Interaction. Springer, Cham, p. 3-10, 2018.	CiteScore: 18th
20	SALLEH, S. M. <i>et al.</i> Analysis of research in programming teaching tools: an initial. <i>Procedia-Social and Behavioral Sciences</i> , v. 103, p. 127-135, 2013.	N/A
21	POWERS, K. <i>et al.</i> Tools for teaching introductory programming: what works?. In: Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education. p. 560-561, 2006.	Qualis: A2. H5 Index: 31
22	FLANNERY, L. P. <i>et al.</i> Designing ScratchJr: support for early childhood learning through computer programming. In: Proceedings of the 12th International Conference on Interaction Design and Children. p. 1-10, 2013.	CiteScore: 12nd



3. RESULTADOS

O mapeamento sistemático empregado nesta pesquisa identificou 113 ferramentas didáticas que apoiam, de alguma forma, o ensino de algoritmos e lógica de programação em cursos introdutórios. Após a análise das ferramentas encontradas e a remoção das direcionadas às disciplinas avançadas e repositórios de exercícios e respostas de programação, no total, restaram 100 ferramentas, sintetizadas, por categorias, no Quadro 4.

Quadro 4 – Categorias das ferramentas identificadas. Fonte: Os autores.

Categorias	Ferramentas
1) Ambiente de programação visual e drag&drop	Jpie, Karel Universe, Blockly, Code, Raptor, BlueJ, Jeliot, Iconic Programmer, Visual Logic, RoboMind, ViLLE, PlanAni, jGRASP, Squeak eToys, Block-C, Osmo, Tica, JHAVÉ, JFLAP, TRAKLA2, Verificator, ETV, ProTracer, RoBlock, Keobot, CLIP 4 Robotics, ProfessorJ
2) Robôs para programação	Lego Mindstorms, iCreate, Scribbler, RoboLab
3) Ambiente de programação móvel	App Inventor
4) Ambiente móvel de programação	PLSOOP, SoloLearn, Programming Hub, Mimo, Enki, CppCode, Grasshopper
5) Correção e avaliação de atividades/feedback	AUTOMATED EVALUATION OF PROGRAMMING ASSIGNMENT, AUTOGRADE, AUTOLEP, AUTOMARK, BOSS, GAME, GLAB, HOGG, JARPEB, JASSESS, KASSANDA, MARMOSET, MOOSHAK, ONLINE ASSESSMENT MANAGEMENT SYSTEM, ONLINE JUDGE, OTO, PASS, PROGRAMHELPER, PROGTEST, SAC, TRILLIUM, WEB-CAT, WEBWORK, Style++, ASSYST, TRY, eGrader, CourseMarker, AutoGradeMe, SSBG, SS-APAG, Athene, Test and Assessment Tool Suite
6) Ambiente de programação com uso de Realidade Aumentada	AR Scratch, AlgoBlocks, ToonTalk, Quetzal, Code Bits, AR Spot
7) Ambiente de programação baseado em jogos	Scratch, Alice, Kodu Game Lab, Toon Talk, Stack Em Up, Scratch Jr, Jeroo, Greenfoot, Dragon Architect
8) Ambiente de programação com simulação 3D	3D LED CUBE, Processing
9) Ambiente de programação baseado em nuvem	RECT, Java Programming Laboratory (JPL), TeamtreeHouse, Codeacademy

Uma relação contendo as ferramentas supracitadas com os artigos selecionados neste trabalho está disponível no *link* disponibilizado na seção 2.2.

4. DISCUSSÃO

Em resposta à primeira questão de pesquisa, os artigos estudados analisaram e/ou apresentaram diversas ferramentas para apoio ao ensino introdutório de programação. Foram encontradas 100 ferramentas didáticas que se encontram divididas em nove categorias, cada qual com sua contribuição e apoio para áreas que são de grande importância ao ensino de algoritmos e lógica de programação. As ferramentas foram agrupadas em categorias de acordo com suas funcionalidades e/ou objetivos que pretendem alcançar, respondendo a segunda questão da pesquisa. O destaque fica para a categoria de ferramentas direcionadas à correção e avaliação de atividades/*feedback*, com o maior número de *software* encontrados. A grande quantidade de alunos presentes em turmas introdutórias de programação é notoriamente uma dificuldade existente no ensino de programação. Por conta disso, essas ferramentas podem auxiliar o docente na tarefa de correção e avaliação de diversas atividades simultaneamente. Em segundo lugar e presente como uma subcaracterística das outras categorias, os ambientes de programação visuais facilitam e auxiliam o entendimento dos conceitos abstratos de



programação por meio de recursos visuais, tornando o ensino mais simples, tangível e prazeroso.

O grande aumento da presença de dispositivos móveis como *smartphones* e conexões móveis de dados faz dos ambientes móveis de programação uma boa alternativa para apoio ao ensino de programação, bem como o ensino do desenvolvimento de aplicações móveis. Em outra categoria, o uso de robôs, realidade aumentada, jogos e elementos visuais 3D trazem dinamismo e inovação para o ensino de programação, aumentando a empolgação e a motivação dos estudantes, fazendo-os participar ativamente da construção de seu conhecimento. Por último, os ambientes de programação baseados em nuvem trazem uma quantidade variada de cursos e instruções que podem auxiliar o estudante na busca de conhecimentos sobre programação, além de auxiliar os docentes na seleção de materiais didáticos para serem explorados em sala de aula.

Com vistas a responder a terceira questão de pesquisa, as ferramentas com mais citações, entre os artigos selecionados, foram Scratch e Alice, sendo citadas por cinco artigos diferentes, seguidas por LegoMindstorms, Greenfoot, Bluej, Jeliot e ViLLe, com três citações cada. Entre as ferramentas de aprendizado móvel a preferida foi Mimo, seguida da SoloLearn e ProgrammingHub. Mimo se mostrou uma ferramenta mais fácil de ser utilizada por iniciantes (AMRO; ROMLI, 2019). No estudo realizado por Resnyansky *et al.* (2018), os estudantes apresentaram melhor desempenho e entendimento dos tópicos, além de satisfação e prazer ao aprender. Ademais, as taxas de reprovação caíram pela metade ao utilizarem ferramentas com uso de realidade aumentada.

Entre as ferramentas para correção e avaliação de atividades/*feedback*, HoGG é a que possui maior base de testes e a ferramenta AutoLep não apresentou limitações no estudo de Ullah *et al.* (2018). Em Souza *et al.* (2016), as ferramentas que possuem o maior número de características desejadas nessa categoria são: PASS, ProgramHelper e WEB-CAT; já as ferramentas mais completas e que aceitam maior número de linguagens de programação são Marmoset e MOOSHAK. Os autores defendem que a seleção da ferramenta dependerá das necessidades do docente. Para Abdullahi (2020), ambientes de programação baseados em nuvem, como o CodeAcademy, possibilitam que alunos resolvam problemas de programação em menos tempo.

A ferramenta Dragon Architect, apresentada por Bauer *et al.* (2017), aumentou o engajamento de estudantes de programação, bem como seu alcance ao incentivar não-programadores a programar. A validação apresentada em Olier *et al.* (2017) mostrou evidências de que o grupo experimental analisado registrou um nível mais alto de consolidação dos conceitos de programação orientada a objetos ao utilizar a ferramenta Greenfoot. Já a ferramenta Block-C teve sucesso ao eliminar erros de sintaxe de seus usuários ao prover uma interface gráfica que tornou mais fácil a assimilação dos conceitos por parte dos estudantes (KYFONIDIS *et al.*, 2017).

Ro-Block, da categoria de ambientes de programação visual, alcançou seu objetivo de ser uma ferramenta autodidática e escalável, tendo sucesso ao auxiliar alunos da educação básica e de nível médio a aprenderem programação (GARCIA; DE LA ROSA, 2016). RoboMind elevou as notas e o entendimento do grupo de teste para conceitos de programação como *loops* (KOORSSE *et al.*, 2015). Code e Blockly, apresentadas em (JÚNIOR *et al.*, 2019), melhoraram o raciocínio lógico e matemático, além de gerarem maior interesse em matemática e programação por parte de seus usuários. A ferramenta CLIP 4 Robotics foi considerada mais fácil do que Alice e Scratch em vários conceitos fundamentais de programação (AL-BAYATY; MARTINEZ, 2018). A ferramenta ProTracer, apresentada em (CHOU; SUN, 2013),



auxiliou os estudantes em suas habilidades de rastreamento de variáveis, contribuindo também para que os docentes identificassem erros de rastreamento e fluxo nos códigos de seus alunos ao prover *feedback*. O uso de simuladores de robôs em (MAJOR *et al.*, 2014) possibilitou que os discentes testados completassem mais tarefas de programação, aumentando também seu prazer ao programar. A revisão sistemática realizada por Major *et al.* 2012 apontou que robôs são ferramentas eficazes, podendo auxiliar programadores iniciantes em seus estudos.

Papadakis *et al.* 2014 apontaram que a ferramenta Scratch é mais apropriada para ensinar jovens estudantes onde o objetivo principal é ser usada no ensino de um currículo de programação, proporcionando um contato mais agradável. Para esses autores, a ferramenta AppInventor é ideal para estimular os estudantes no que tange à programação móvel. Outras ferramentas que merecem destaque, de acordo com (POWERS *et al.*, 2006), são Alice e Jeroo – ferramentas narrativas que apoiam a programação ao contar histórias, Jpie e Karel Universe como ambientes de programação visual que suportam a construção de programas por meio de uma interface *drag&drop*, e Raptor, Iconic Programmer e VisualLogic como ferramentas de modelo de fluxo.

5. CONSIDERAÇÕES FINAIS

As ferramentas identificadas nesta pesquisa visam, de maneiras variadas, contribuir para a diminuição da dificuldade inerente ao ensino e aprendizado de programação em cursos introdutórios. Nesse contexto, este trabalho forneceu um levantamento das ferramentas didáticas desenvolvidas para esse objetivo, bem como uma classificação em categorias. A categorização proposta poderá auxiliar o leitor na busca de ferramentas para resolver determinado tipo de problema. Para tanto, foi realizado um mapeamento sistemático da literatura, no qual foram extraídos e analisados 339 artigos oriundos de bibliotecas digitais consideradas referências na área. Após o processo de seleção e aplicação de filtros de inclusão e exclusão, foram obtidos 22 artigos relacionados ao objetivo do artigo. Como trabalho futuro sugere-se uma análise e classificação de ferramentas didáticas com base em seus diferentes paradigmas de programação.

REFERÊNCIAS BIBLIOGRÁFICAS

ABDULLAHI, M. S. I. Cloud-based learning system for improving students' programming skills and self-efficacy. **Journal of Information and Communication Technology**, v. 17, n. 4, p. 629-651, 2020.

AL-BAYATY, A.; MARTINEZ, C. CLIP 4 Robotics: a click-based programming language. In: **International Conference on Human-Computer Interaction**. Springer, Cham, p. 3-10, 2018.

AMRO, J. S.; ROMLI, R. Investigation on the learning programming techniques via mobile learning application. In: **4th International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE)**. IEEE, p. 1-7, 2019.

BAUER, A. *et al.* Dragon architect: open design problems for guided learning in a creative computational thinking sandbox game. In: **Proceedings of the 12th International Conference on the Foundations of Digital Games**. p. 1-6, 2017.



CAMP, O.; WOODWARD, R. ICT Enhanced learning experience for an introductory object oriented programming course-A case study. In: **International Conference on Computer Supported Education**. SCITEPRESS, p. 16-25, 2011.

CHOU, C.; SUN, P. An educational tool for visualizing students' program tracing processes. **Computer Applications in Engineering Education**, v. 21, n. 3, p. 432-438, 2013.

DENNER, J. *et al.* Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts?. **Computers & Education**, v. 58, n. 1, p. 240-249, 2012.

DE SOUZA, N. G. *et al.* Proposta de uma metodologia para apoiar os processos de ensino e de aprendizagem de lógica de programação na modalidade de educação a distância. **Educação, Cultura e Comunicação**, v. 9, n. 18, 2018.

FLANNERY, L. P. *et al.* Designing ScratchJr: support for early childhood learning through computer programming. In: **Proceedings of the 12th International Conference on Interaction Design and Children**. p. 1-10, 2013.

GARCIA, P. G. F.; DE LA ROSA, F. RoBlock–web app for programming learning. **International Journal of Emerging Technologies in Learning (iJET)**, v. 11, n. 12, p. 45-53, 2016.

JÚNIOR, A. M. *et al.* Programming teaching tools feature assessment associated with Brazilian curriculum base obtained through BPL platforms analysis. In: **International Conference on Human-Computer Interaction**. Springer, Cham, p. 380-389, 2019.

KOORSSE, M. *et al.* Programming assistance tools to support the learning of IT programming in South African secondary schools. **Computers & Education**, v. 82, p. 162-178, 2015.

KYFONIDIS, C. *et al.* Block-C: A block-based programming teaching tool to facilitate introductory C programming courses. In: **IEEE Global Engineering Education Conference (EDUCON)**. IEEE, p. 570-579, 2017.

MAJOR, L. *et al.* The effectiveness of simulated robots for supporting the learning of introductory programming: a multi-case case study. **Computer Science Education**, v. 24, n. 2-3, p. 193-228, 2014.

MAJOR, L. *et al.* Systematic literature review: teaching novices programming using robots. **IET Software**, v. 6, n. 6, p. 502-513, 2012.

MARTÍNEZ, L. G. *et al.* Using a mobile platform for teaching and learning object oriented programming. **IEEE Latin America Transactions**, v. 16, n. 6, p. 1825-1830, 2018.

OLIER, A. J. *et al.* Design and implementation of a teaching tool for introduction to object oriented programming. **IEEE Latin America Transactions**, v. 15, n. 1, p. 97, 2017.



PAPADAKIS, S. *et al.* Novice programming environments. Scratch & app inventor: a first comparison. In: **Proceedings of the 2014 Workshop on Interaction Design in Educational Environments**. p. 1-7, 2014.

PETERSEN, K. *et al.* Guidelines for conducting systematic mapping studies in software engineering: an update. **Information and Software Technology**, v. 64, p. 1-18, 2015.

POWERS, K. *et al.* Tools for teaching introductory programming: what works?. In: **Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education**. p. 560-561, 2006.

RESNYANSKY, D. *et al.* The potential of augmented reality for computer science education. In: **IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)**. IEEE, p. 350-356, 2018.

ROCHA, P. S. *et al.* Ensino e aprendizagem de programação: análise da aplicação de proposta metodológica baseada no sistema personalizado de ensino. **RENOTE**, v. 8, n. 3, 2010.

SALLEH, N. *et al.* Cloud-based learning system for improving students' programming skills and self-efficacy. **Journal of Information and Communication Technology**, v. 17, n. 4, p. 629-651, 2018.

SALLEH, S. M. *et al.* Analysis of research in programming teaching tools: an initial. **Procedia-Social and Behavioral Sciences**, v. 103, p. 127-135, 2013.

SBC. **Currículo de Referência da Sociedade Brasileira de Computação para Cursos de Graduação em Computação e Informática**. 2005. Disponível em: <<https://www.sbc.org.br/documentos-da-sbc/send/131-curriculos-de-referencia/760-curriculo-de-referencia-cc-ec-versao2005>>. Acesso: Jun. 2020.

SHEARD, J. *et al.* Analysis of research into the teaching and learning of programming. In: **Proceedings of the Fifth International Workshop on Computing Education Research Workshop**. p. 93-104, 2009.

SILVA-MACEDA, G. *et al.* More time or better tools? A large-scale retrospective comparison of pedagogical approaches to teach programming. **IEEE Transactions on Education**, v. 59, n. 4, p. 274-281, 2016.

SOUZA, D. M. *et al.* A systematic literature review of assessment tools for programming assignments. In: **29th International Conference on Software Engineering Education and Training (CSEET)**. IEEE, p. 147-156, 2016.

ULLAH, Z. *et al.* The effect of automatic assessment on novice programming: strengths and limitations of existing systems. **Computer Applications in Engineering Education**, v. 26, n. 6, p. 2328-2341, 2018.

WOHLIN, C. **Experimentation in Software Engineering**. Berlin: Springer, 2012.