

Mapeamento sistemático sobre ferramentas digitais online para o ensino-aprendizagem de algoritmos e programação no ensino superior

Beatriz Vendrame, Emely Albernaz, Rodrigo Sacchi, Valguima Odakura

¹Faculdade de Ciências Exatas e Tecnologia (FACET)
Universidade Federal da Grande Dourados (UFGD)
Dourados – MS – Brasil

{beatrizvendrame, emelyalbernaz, valguima.odakura}@gmail.com

rodrigoscacchi@ufgd.edu.br

Abstract. *This article presents a systematic mapping of the literature on on-line digital tools for teaching-learning algorithms and programming in higher education. The initial classes of algorithms and programming of undergraduate courses in Computing have high rates of failure and evasion. In order to contribute to this scenario, this paper aims to identify the existing software and applications that aid in the teaching-learning process of algorithms and programming, as well as to identify if the use of such tools contributes to the improvement of learning.*

Resumo. *Neste artigo é apresentado um mapeamento sistemático da literatura sobre ferramentas digitais online para o ensino-aprendizagem de algoritmos e programação no ensino superior. As disciplinas iniciais de algoritmos e programação dos cursos de graduação em Computação possuem altos índices de reprovação e evasão. Buscando contribuir neste cenário, este trabalho tem o objetivo de identificar os softwares e aplicações existentes que auxiliam no processo de ensino-aprendizagem de algoritmos e programação, bem como identificar se a utilização de tais ferramentas contribui para a melhoria da aprendizagem.*

1. Introdução

As disciplinas de algoritmos e programação compõem uma parte fundamental dos currículos dos cursos da área de Computação. No entanto, as disciplinas que abordam esses conteúdos costumam ter altas taxas de evasão e reprovação no mundo todo. No trabalho de [Watson and Li 2014] foram pesquisados 161 cursos de Ciência da Computação que ocorreram entre os anos de 1979 a 2013 em 51 diferentes instituições de 15 países distintos. Como resultado, há uma taxa média de aprovação de 67,7%. Embora não pareça uma taxa extremamente baixa, quando as mesmas são analisadas por países, elas revelam diferenças significativas. No Brasil, por exemplo, a taxa de aprovação é de 45%, ou seja, o índice de reprovação é altíssimo, e ainda o trabalho sugere que tais índices não tendem a melhorar com o tempo.

Neste contexto, os métodos tradicionais de ensino têm sido questionados, e argumenta-se sua insuficiência para atender às necessidades individuais dos estudantes.

Uma das razões apontadas é que no ensino tradicional todos os estudantes devem aprender no mesmo ritmo, não atendendo às especificidades de aprendizagem de cada um. Esta concepção não contempla os diferentes estilos de aprendizagem de cada estudante [Felder et al. 1988].

Entre os problemas identificados na aprendizagem de programação tem-se que, por um lado, diversas vezes os professores se preocupam mais em ensinar uma linguagem de programação e seus detalhes sintáticos, ao invés de estimular a resolução de problemas usando a linguagem de programação. Neste sentido, a linguagem deveria servir como uma ferramenta para expressar ideias e algoritmos. Por outro lado, alguns métodos de estudos seguidos por vários estudantes não são apropriados para aprendizagem de programação, como no caso que creem que vão aprender a programar através da leitura de um livro didático, não compreendendo que sua atividade principal deveria ser resolver inúmeros problemas de programação. Além disso, relata-se a falta de persistência dos alunos, que frequentemente desistem de solucionar um problema caso não encontrem uma solução rapidamente [Gomes and Mendes 2007].

Considerando esses aspectos, este trabalho tem como objetivo o desenvolvimento de um mapeamento sistemático das produções científicas nacionais e internacionais que tratam de ferramentas digitais *online* para o ensino-aprendizagem de algoritmos e programação, tais como objetos de aprendizagem, web sites, aplicativos, jogos, ambientes de desenvolvimento integrados, entre outras, para cursos de graduação.

O restante do artigo está organizado da forma que se segue: Na Seção 2 são descritos os trabalhos relacionados. Na Seção 3 é apresentada a metodologia de mapeamento sistemático. Na Seção 4 são apontados e discutidos os resultados obtidos, além de discutir as ameaças à validade do estudo. Finalmente, na Seção 5 são relatadas as considerações finais e indicados os trabalhos futuros.

2. Trabalhos Relacionados

O trabalho de [Barbosa and Júnior 2013] descreve um mapeamento sistemático sobre ferramentas para o apoio no ensino de algoritmos e estruturas de dados, tendo como enfoque identificar e classificar as ferramentas existentes. No trabalho de [Marcolino and Barbosa 2015] é relatado um mapeamento sistemático sobre softwares educacionais para o ensino de programação, identificando ferramentas e aplicações educacionais existentes na literatura.

O primeiro trabalho está focado nas ferramentas para algoritmos e estruturas de dados e o segundo na programação. Neste trabalho consideramos ferramentas para os dois temas, uma vez que estão intrinsecamente ligados. Diferente dos dois trabalhos descritos, as ferramentas consideradas neste mapeamento são apenas as *online*.

3. Método de Pesquisa

Um Mapeamento Sistemático (MS) é uma forma de identificar, avaliar e interpretar todos os trabalhos disponíveis relevantes a uma determinada área de pesquisa, para que se tenha o conhecimento de quais tópicos já foram abordados naquela área, fazendo uma análise dos resultados com o intuito de mostrar a frequência de publicações por categoria ou ainda para mostrar quais tópicos ainda estão abertos para pesquisa [Petersen et al. 2008, Kitchenham 2004].

Para conduzir um mapeamento sistemático é necessário um processo de planejamento, em que se registre todos os passos executados para que se possa, ao final, apresentar os resultados obtidos e análises realizadas. Para nortear esse processo, pode-se seguir um método para o desenvolvimento de mapeamento sistemático, descrevendo cinco passos a serem seguidos [Petersen et al. 2008]:

- (I) Definição de questões de pesquisa;
- (II) Realização da pesquisa de estudos primários relevantes;
- (III) Triagem dos documentos;
- (IV) Esquema de classificação;
- (V) Extração de dados e mapeamento.

As questões de pesquisa são elaboradas de acordo com os objetivos que se pretende alcançar. Já as chaves de pesquisas, ou *string* de busca, na maioria das vezes, são extraídas a partir das próprias questões de pesquisa, sendo esta uma estratégia definida por [Kitchenham 2004]. No que se diz respeito à triagem dos documentos, são aplicados critérios de inclusão e exclusão para seleção dos trabalhos através da análise de título, palavras-chave e resumo. Na extração de dados, busca-se responder às questões de pesquisas propostas, ilustrando os dados obtidos através de tabelas e gráficos.

Sendo assim, neste trabalho é realizado um Mapeamento Sistemático baseado na metodologia descrita por [Petersen et al. 2008, Petersen et al. 2015].

3.1. Questões de Pesquisa

Para atingir os objetivos deste mapeamento sistemático foram elaboradas as seguintes questões de pesquisa:

- QP1: Quais as ferramentas *online* de aprendizagem mais utilizadas no ensino superior para as disciplinas de algoritmos e programação?
- QP2: As ferramentas consideraram uma teoria de aprendizagem em seu desenvolvimento?
- QP3: De que forma as ferramentas foram avaliadas?
- QP4: As ferramentas contribuíram para a melhoria do processo de ensino-aprendizagem?

3.2. String de Busca

A elaboração da *string* de busca, utilizada na maioria das bases digitais, segue o método definido por [Kitchenham 2004]. Este método seleciona as principais palavras-chaves a partir das questões de pesquisa, utilizando os parênteses para criar buscas mais complexas, o conector OR para combinar sinônimos e termos alternativos de cada palavra-chave e o conector AND para combinar as palavras-chave. Os termos compostos devem ser fechados com aspas e o símbolo "*" usado como caractere coringa. A utilização dessas especificações pode alterar significativamente a quantidade de estudos à serem retornados.

A *string* de busca foi definida por um processo de testes da *string* de busca nas bases, analisando os resultados retornados e realizando ajustes na *string* de modo a obter as melhores taxas de aceitação entre as bases e a quantidade de resultados retornados, de modo a ser adequado para a execução deste trabalho. A *string* de busca resultante foi:

```
((software OR tool OR app) AND (education) AND (web OR online) AND (programming OR algorithm*) AND (graduation OR undergraduate OR college OR "higher education"))
```

Durante o processo de definição da *string* de busca, verificou-se que os resultados retornados nas bases nacionais mudavam substancialmente com a alteração de algumas palavras, utilizando a tradução da *string* em língua inglesa para a língua portuguesa. Deste modo, a fim de não ameaçar a validade da pesquisa perdendo alguns trabalhos, optou-se por não utilizar a mesma *string* das bases internacionais nas bases nacionais. A decisão foi adaptar a *string* original para atingir um maior número de estudos nas bases nacionais. Sendo essa especificação um artifício que permite facilitar um contexto de replicação futura deste estudo. Como resultado desta adaptação, foi obtida a seguinte *string* de busca:

```
(software OR ferramenta OR aplicativo*) AND (aprendiz* OR ensino OR método*) AND (web OR online OR digital) AND (programação OR algoritmo*)
```

3.3. Fontes de Busca

A técnica utilizada para procurar os estudos se deu através das buscas automatizadas em um conjunto bibliotecas digitais, utilizando as *strings* de busca definidas. As bibliotecas digitais utilizadas na busca automatizada foram:

- Base Nacional: Repositório da Comissão Especial de Informática na Educação (CEIE), que conta com as publicações: Workshops do Congresso Brasileiro de Informática na Educação (WCBIE), Revista Brasileira de Informática na Educação (RBIE), Simpósio Brasileiro de Informática na Educação (SBIE), Workshop de Informática na Escola (WIE)¹.
- Bases Internacionais: *ACM Digital Libray*², *IEEE Xplore*³ e *Scopus*⁴.

Na escolha das bases foi considerada uma mescla entre bases internacionais amplamente reconhecidas na Computação, bem como as fontes nacionais de Informática na Educação da SBC. Outras bases poderiam ter sido incluídas, o que resultaria em uma cobertura mais abrangente, porém, o trabalho a ser realizado também cresceria bastante.

3.4. Seleção dos Estudos Primários

Com a finalidade de filtrar somente os artigos que apresentam alguma relação com a temática definida, foram definidos os critérios de inclusão e exclusão. Como critérios de inclusão considera-se: artigos disponíveis para *download*; artigos completos; artigos que respondam a pelo menos uma das questões de pesquisa. Como critérios de exclusão tem-se: artigos incompletos; artigos repetidos; artigos sem acesso; artigos que não contemplam o escopo do mapeamento; relatórios técnicos, documentos que estão na forma de resumos ou apresentações.

¹Repositório de publicações da CEIE, disponível em: <http://www.br-ie.org/pub/index.php/index>

²ACM Digital Libray, disponível em: <https://dl.acm.org/>

³IEEE Xplore, disponível em: <https://ieeexplore.ieee.org/Xplore/home.jsp>

⁴Scopus, disponível em: <https://www.scopus.com/search/form.uri?display=basic>

3.5. Ameaças à Validade

Para assegurar que o processo de seleção fosse imparcial, as questões de pesquisa e os critérios de inclusão e exclusão foram definidos antes do início do mapeamento. A triagem dos estudos foi realizada por duplas de pesquisadores a fim de cessar quaisquer dúvidas, além de reduzir os riscos de interpretações pessoais. Em caso de conflito entre a dupla de avaliadores, um terceiro avaliador procedeu com a análise, dirimindo discordâncias. Considerando que foram utilizadas as buscas automáticas em 4 bases de dados, é possível que estudos relevantes ao mapeamento não tenham sido incluídos.

4. Resultados

Esta seção exhibe os resultados encontrados no mapeamento sistemático, obtidos a partir dos estudos selecionados e da análise dos mesmos. A lista dos trabalhos selecionados para o mapeamento estão disponíveis para consulta em <http://bit.ly/2DB8KKf>.

4.1. Informações Gerais dos Estudos Primários

Ao realizar as buscas automáticas com as *strings* de busca definidas nas bases selecionadas, a quantidade de estudos retornados totalizou 711 trabalhos primários, como pode ser visualizado na Tabela 1. Na primeira etapa de seleção, todos os estudos primários retornados pelas bases de dados foram avaliados através da leitura dos títulos, resumos e palavras-chaves, aplicando os critérios de inclusão e exclusão, a fim de selecionar somente estudos relevantes para responder às questões de pesquisa. Deste modo, o conjunto inicial de 711 estudos retornados foi reduzido para 68 artigos. Essa redução se deu devido a grande maioria dos estudos relatar sobre a utilização de algum algoritmo e não sobre o ensino de algoritmos em si. Na segunda etapa de seleção, que constitui a etapa final dessa análise, os 68 artigos foram lidos na íntegra, e este total de estudos decaiu para apenas 50 estudos.

Tabela 1. Processo de seleção dos estudos.

	Fontes	Estudos Retornados	1ª Seleção Estudos Incluídos	2ª Seleção Estudos Incluídos
Internacional	ACM	360	28	20
	IEEE	96	18	14
	SCOPUS	221	10	8
Nacional	CBIE	6	3	1
	RBIE	8	2	0
	SBIE	20	7	7
	TOTAL	711	68	50

O processo de seleção dos trabalhos foi realizado por 2 pesquisadores, e em casos de eventuais desacordos entre os dois, utilizou-se um terceiro pesquisador a fim de sanar quaisquer divergências que pudessem ameaçar à validade do presente estudo. As buscas pelos trabalhos nas bases foram conduzidas durante o mês de abril de 2018, abrangendo o período de 2010 até março de 2018. Uma vez que o objetivo do trabalho é encontrar ferramentas *online*, buscou-se trabalhos mais recentes, iniciando as buscas no ano de 2010.

Considerando o conjunto final de estudos primários incluídos no mapeamento sistemático, ressalta-se que a maioria dos estudos selecionados estavam disponíveis nas bibliotecas digitais internacionais, correspondendo a 84% dos trabalhos e apenas 16% deles na base nacional, como pode ser observado na Figura 1. Atribui-se esse resultado ao fato da busca ter sido realizada em mais bases internacionais, além de o número de publicações nessas bases ser bem maior quando comparado com as nacionais.

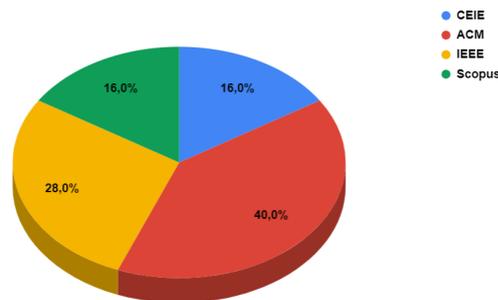


Figura 1. Distribuição dos estudos primários de acordo com as bases.

Para a realização da pesquisa, foi considerada uma janela temporal entre os anos de 2010 e 2018. A Figura 2 ilustra a evolução da publicação dos estudos primários ao longo dos anos. A quantidade de trabalhos apresentou um declínio entre os anos de 2014 e 2016, superando esse valor no ano seguinte. Já o ano de 2018 não deve ser considerado em sua totalidade, pois só estão incluídos nesta pesquisa aqueles que foram publicados até março de 2018, período de realização das buscas.

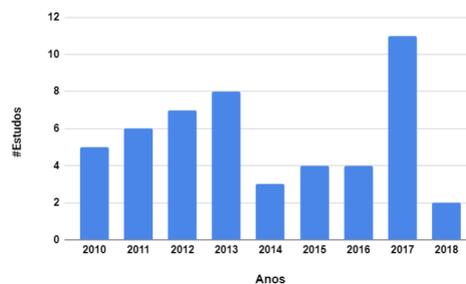


Figura 2. Distribuição dos estudos primários ao longo dos anos.

Os pesquisadores dos trabalhos considerados têm origem em 18 países distintos, conforme ilustrado na Figura 3. O total das publicações de cada país é relativamente maior quando comparado com a quantidade de estudos selecionados, isto acontece porque alguns estudos foram desenvolvidos com a participação de dois ou mais pesquisadores com origem em instituições de países distintos.

4.2. Respostas às Questões de Pesquisa

Esta seção apresenta os resultados deste mapeamento sistemático como respostas aos questionamentos levantados nas quatro questões de pesquisas propostas.

QP1: Quais as ferramentas *online* de aprendizagem mais utilizadas no ensino superior para as disciplinas de algoritmos e programação?

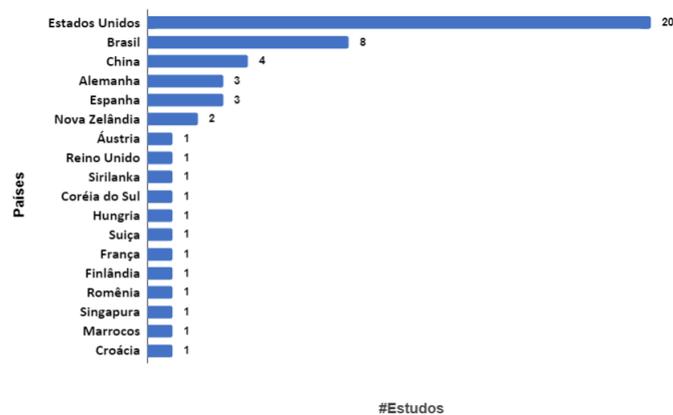


Figura 3. Distribuição pelos países das instituições dos autores.

Para responder essa questão de pesquisa, os 50 artigos selecionados foram lidos na íntegra e foram extraídos os nomes das ferramentas utilizadas em cada um. O resultado aponta que não há uma ferramenta consolidada para ensino-aprendizagem de algoritmos e programação no ensino superior, uma vez que foram reportadas 50 ferramentas distintas. A lista com o nome de todas as ferramentas está disponível em <http://bit.ly/2DB8KKf>.

Procurando extrair mais informações desses dados, buscou-se classificar as ferramentas encontradas. Para essa tarefa, utilizou-se a categorização apresentada por [Pears et al. 2007], que organiza as ferramentas para aprendizagem de programação em 5 categorias:

- Ferramentas de visualização de código, que focam na visualização estática de estruturas ou exibem os aspectos dinâmicos da execução do programa e são frequentemente associadas a ambientes de programação.
- Ferramentas de avaliação automática são aquelas que, além de determinar a correção, também podem analisar a eficiência do programa ou sua complexidade, através do uso de recursos de linguagem específicos, estilo de codificação e alguns aspectos do design do programa.
- Ferramentas de apoio à programação são aquelas que organizam componentes e recursos do programa em projetos, recursos de edição avançados específicos do idioma (por exemplo, sintaxe, realce, conclusão de código). Ou seja, que apoiam os alunos na criação de programas dentro de uma execução padrão, na maioria das vezes, dentro de um ambiente virtual.
- Micromundos, termo que designa ambientes de aprendizagem nos quais é possível explorar, descobrir e simular acontecimentos da vida real ou de fantasia, geralmente fazendo-se o uso de metáforas.
- Outros, ferramentas que usam estratégias específicas, não incluídas nas categorias anteriores.

As ferramentas foram agrupadas por tipo e ano de publicação, conforme a Figura 4.

É possível verificar que a maior parte dos estudos selecionados utilizam ferramentas de apoio à programação, totalizando 30 ferramentas, seguida das ferramentas para visualização e ferramentas de avaliação automática com 10 cada. Não houve relatos em nenhum dos estudos sobre ferramentas do tipo micromundos e nem de outras que não se

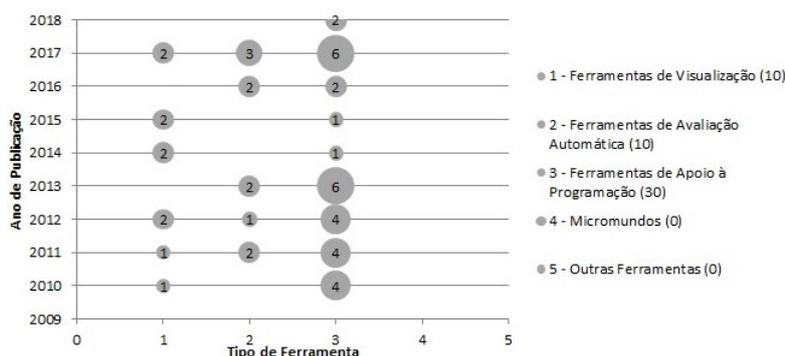


Figura 4. Tipos de ferramentas por ano de publicação.

incluíam em nenhum dos tipos descritos. Em relação aos anos, ressalta-se 2017 o ano com mais trabalhos, com 11 registros.

QP2: As ferramentas consideraram uma teoria de aprendizagem em seu desenvolvimento?

Esta questão de pesquisa está relacionada a concepção de aprendizagem que fundamenta a construção das ferramentas. Durante a leitura completa dos trabalhos, considerou-se apenas os casos em que os autores relatam utilizar uma teoria de aprendizagem. Este levantamento resultou em apenas dois trabalhos que mencionavam a utilização de alguma teoria de aprendizagem no seu desenvolvimento. As teorias reportadas foram o cognitivismo e o construtivismo.

A escassez de relatos sobre utilização de uma teoria de aprendizagem indica que as ferramentas para ensino-aprendizagem de algoritmos e programação estão sendo desenvolvidas sem esta preocupação. Deste modo, indagamos se este fator não estaria afetando a utilização de tais ferramentas, uma vez que não encontramos nenhuma ferramenta que tivesse maior adoção.

QP3: De que forma as ferramentas foram avaliadas?

Nesta questão intenciona-se saber de que maneira as ferramentas descritas foram avaliadas. Os resultados para essa questão podem ser visualizados na Figura 5. Foram utilizados questionários em 38% dos trabalhos, sendo estes questionários aplicados tanto para estudantes como para professores. Grande parte dos questionários foram aplicados por meio de formulários *online*. Além disso, na maioria das vezes estes foram aplicados antes e depois da utilização da ferramenta, seguido de uma análise qualitativa dos dados obtidos.

Outras formas de avaliação foram destacadas em 36% dos trabalhos, sendo estes: resultados coletados através das ferramentas, como notas, pontuações atribuídas na realização das atividades, qual a diferença de conhecimento antes e depois de utilizar a ferramenta, qual o grau de contribuição feita por estudantes durante o desenvolvimento de programas, entre outros.

Por fim, 26% dos estudos não relataram sobre nenhum método de avaliação das ferramentas desenvolvidas. Nesta categoria estão presentes protótipos, dentre os quais, alguns autores indicaram uma possível realização destas avaliações em estudos futuros.

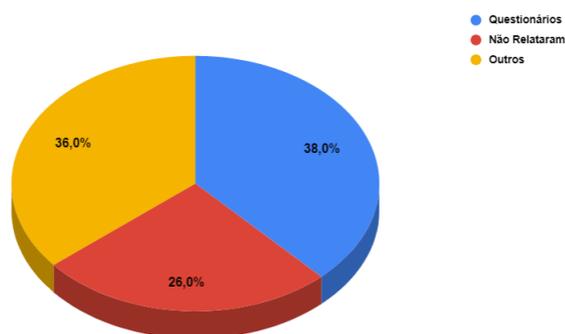


Figura 5. Formas de avaliação das ferramentas.

QP4: As ferramentas contribuíram para a melhoria do processo de ensino-aprendizagem?

Nesta questão buscou-se conhecer a contribuição das ferramentas para melhoria do processo de ensino-aprendizagem. A maioria dos trabalhos, 70%, relatou a contribuição das ferramentas no processo de ensino-aprendizagem, como mostra a Figura 6.

Nestes trabalhos verificou-se a melhora no aumento dos índices de aprovação, complementado pelos relatos de proporcionar um aprendizado flexível, facilitando a capacidade dos estudantes em compreender, depurar e desenvolver esquemas algorítmicos. Outro ponto a destacar é que os estudantes relatam se sentir motivados e interessados devido a grande satisfação com a qualidade e rapidez do *feedback*.

Apenas 6% dos trabalhos descrevem de maneira informal e subjetiva a melhoria. Por outro lado, 4% dos estudos relataram que não houve uma melhora no processo de ensino-aprendizagem, justificando que a escolha da ferramenta não teve um impacto significativo no sucesso da aprendizagem. Por fim, 20% dos estudos considerados não relataram a aplicação da ferramenta, embora alguns indicaram a possível realização desta em estudos futuros.

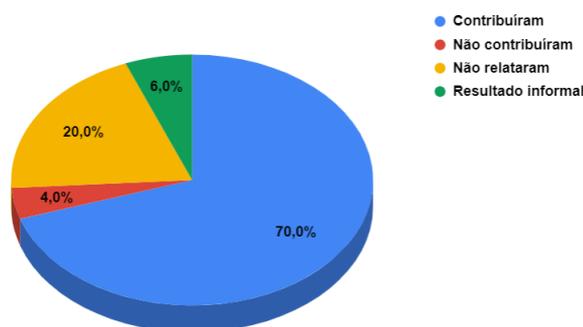


Figura 6. Contribuição das ferramentas no processo ensino-aprendizagem.

5. Considerações Finais

Neste artigo foi relatado um mapeamento sistemático da literatura nacional e internacional sobre as ferramentas digitais utilizadas para o ensino-aprendizagem de algoritmos e

programação em cursos de graduação. Após a seleção de trabalhos, foram analisados 50 artigos completos.

Não foi encontrado nos trabalhos predominância de nenhuma ferramenta, ao contrário, nos 50 trabalhos foram reportadas 50 ferramentas diferentes. Dentre essas ferramentas, observa-se que o tipo mais utilizado são as de apoio à programação, ou seja, que organizam componentes e recursos do programa, como sintaxe, realce, conclusão de código, auxiliando os estudantes na criação e execução de programas. Um resultado promissor é que 70% dos trabalhos afirmaram que as ferramentas contribuíram para a melhoria do processo de ensino-aprendizagem,

Como oportunidades para pesquisas futuras vislumbra-se investigar porque nenhuma das ferramentas conseguiu ser amplamente adotada. Entre as hipóteses está o fato de as ferramentas serem desenvolvidas por um pequeno grupo para solucionar um problema local, não alcançando um público mais amplo.

Referências

- Barbosa, W. A. and Júnior, P. A. P. (2013). Um mapeamento sistemático sobre ferramentas de apoio ao ensino de algoritmo e estruturas de dados. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 24, page 406.
- Felder, R. M., Silverman, L. K., et al. (1988). Learning and teaching styles in engineering education. *Engineering education*, 78(7):674–681.
- Gomes, A. and Mendes, A. J. (2007). Learning to program-difficulties and solutions. In *International Conference on Engineering Education-ICEE*, volume 2007.
- Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33(2004):1–26.
- Marcolino, A. and Barbosa, E. F. (2015). Softwares educacionais para o ensino de programação: Um mapeamento sistemático. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 26, page 190.
- Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., Devlin, M., and Paterson, J. (2007). A survey of literature on the teaching of introductory programming. In *ACM sigcse bulletin*, volume 39, pages 204–223. ACM.
- Petersen, K., Feldt, R., Mujtaba, S., and Mattsson, M. (2008). Systematic mapping studies in software engineering. In *EASE*, volume 8, pages 68–77.
- Petersen, K., Vakkalanka, S., and Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64:1–18.
- Watson, C. and Li, F. W. (2014). Failure rates in introductory programming revisited. In *Proceedings of the 2014 conference on Innovation & technology in computer science education*, pages 39–44. ACM.