

Comparação entre o panorama internacional e nacional sobre o Ensino e a Aprendizagem de Introdução à Programação no Ensino Superior

Rodrigo Pessoa Medeiros^{1,2}, Taciana Pontual Falcão³, Geber Lisboa Ramalho¹

¹Centro de Informática - Universidade Federal de Pernambuco (UFPE)
Recife, PE – Brasil

²Mestrado Profissional em Engenharia de Software – CESAR School
Recife – PE – Brasil

³Departamento de Computação – Universidade Federal Rural de Pernambuco (UFRPE)
Recife - PE - Brasil.

{rpm@cesar.school, taciana.pontual@ufrpe.br, glr@cin.ufpe.br}

Abstract. *Introductory programming courses are part of the curricula of several undergraduate degrees. However, the understanding of the main problems experienced by students and teachers are still diffuse. This tertiary study aims to verify to what extent the teaching and learning challenges of introductory programming in Brazil and abroad are aligned, contributing to defining solutions adequate to the national context. To reach this goal, two systematic literature reviews (SLR) on this theme were analyzed - one international and one national. The results indicate that problem solving in programming appears as a crucial concept for the introduction to programming. National studies (Brazil) reinforce basic knowledge as important deficit, and both SRL reinforce the need for further studies on methods and tools for teaching programming.*

Resumo. *Disciplinas de introdução à programação fazem parte de currículos de vários cursos de graduação, porém ainda é difuso o entendimento sobre quais são, de fato, os principais problemas vivenciados por alunos e professores. Este estudo terciário tem como objetivo verificar em que medida os desafios de ensino e aprendizagem de introdução à programação no Brasil e no exterior estão alinhados, contribuindo para a definição de soluções adequadas ao contexto nacional. Para isso, foram analisadas duas revisões sistemáticas da literatura (RSL) sobre o tema - uma internacional e uma nacional. Os resultados indicam que a resolução de problemas em programação aparece como conceito crucial para a introdução à programação. Os estudos nacionais reforçam os conhecimentos de base como déficits importantes e ambas RSL reforçam a necessidade de estudos mais aprofundados nos métodos e ferramentas para o ensino de programação.*

1. Introdução

Disciplinas de introdução à programação fazem parte de vários currículos de graduação dos currículos dos diversos cursos de graduação em computação (ciência, engenharia e licenciatura em computação, sistemas de informação, entre outros) e são basilares para o sucesso dos estudantes no decorrer do curso. Porém, ainda é difuso o entendimento sobre quais são, de fato, os principais problemas vivenciados por alunos e professores no processo de ensino e aprendizagem de programação (Medeiros, 2019).

Este artigo tem como objetivo responder à pergunta de pesquisa: Em que medida os problemas enfrentados no ensino e aprendizagem de introdução à programação no Ensino Superior no Brasil são diferentes daqueles enfrentados no exterior? É importante saber se existem e quais são tais diferenças, porque se houver, não podemos simplesmente aplicar os métodos e soluções preconizados na literatura internacional sem os devidos questionamentos e adaptações para a realidade local.

Para responder à pergunta de pesquisa, executamos um estudo terciário. Existem estudos primários que focam em pontos específicos do ensino (por exemplo, Kinnunen, 2009; Teague, 2011) e da aprendizagem de programação (por exemplo, Lahtinen, Ala-Mutka e Järvinen, 2005; Bennedsen e Caspersen, 2007). E existem estudos secundários, que são compilações de estudos primários para responder perguntas de pesquisa, como as revisões sistemáticas da literatura (RSL) (por exemplo, Ramos *et al.*, 2015; Vihavainen, Airaksinen e Watson, 2014). Já os estudos terciários analisam as contribuições de estudos secundários, como feito no presente trabalho.

A literatura sobre ensino e aprendizagem de introdução à programação ainda diverge no diagnóstico dos problemas, além de apresentar reflexões de forma pouco sistemática (Medeiros, 2019, p. 154). Além disso, também existe pouca literatura comparando a situação brasileira e mundial. Uma das honrosas exceções é o estudo de Ortiz e Pereira (2019) que, porém, trata apenas de Pensamento Computacional, uma parte do problema mais amplo de Introdução à Programação, de interesse do nosso estudo. Ademais o estudo não foca exclusivamente no Ensino Superior. Outro estudo interessante é o de Ramos *et al.* (2015), comparando a realidade nacional versus a realidade estrangeira sobre ensino e aprendizagem de introdução à programação no Ensino Superior. Esse estudo foi uma RSL entre 2001 e 2014 nos eventos nacionais, onde os resultados foram comparados com a realidade internacional, mas sem que fosse feita uma RSL dos estudos internacionais. Apesar de muito rico, entendemos haver espaço para aprofundar a questão tanto porque há artigos publicados depois de 2014 e porque é possível fazer uma comparação mais rigorosa das literaturas nacional e internacional se nos basearmos em RSLs.

Por fim, foram realizadas RSL sobre analisadores de código nas disciplinas de programação (Barbosa, 2015), ambientes de ensino de programação (Silva, 2015), uso de jogos nas disciplinas de programação (Da Silva, Medeiros, Aranha, 2014; Scaico, Scaico, 2016), juiz online no ensino de programação (Francisco, Júnior e Ambrósio, 2016), robótica aplicada ao ensino de programação (Almeida, Netto, 2015),

conhecimentos, habilidades, atitudes e competências para auxiliar na aprendizagem de programação (Henrique, Tedesco, 2017) e sobre ensino e aprendizagem de programação (Aureliano e Tedesco, 2012; Gomes e Mendes, 2015). Não vamos citar neste artigo os estudos primários, mas eles estão analisados nos estudos secundários.

Neste artigo, apresentamos um estudo terciário comparando duas RSL: uma sobre o contexto internacional (Medeiros, Ramalho e Pontual Falcão, 2018), em que 89 artigos internacionais foram analisados; e outra sobre o contexto nacional (Medeiros, Pontual Falcão e Ramalho, 2020), incluindo 69 artigos nacionais. Ambas RSL focaram em publicações sobre introdução à programação no Ensino Superior publicados entre 2010 e 2016 e tiveram as mesmas perguntas de pesquisa, referentes às habilidades e conhecimentos prévios para um aluno iniciante aprender a programar, às dificuldades dos estudantes na aprendizagem de programação, e aos desafios enfrentados pelos professores ao ensinar introdução à programação - permitindo assim a análise comparativa.

Para identificar e analisar as diferenças entre os resultados dos dois estudos, focamos nos aspectos que apareceram nos resultados das duas RSL, mas com uma diferença que discricionariamente estipulamos em acima de 10 pontos percentuais. Nos casos de aparecer em apenas uma RSL focamos em aspectos com, ao menos, 5 pontos percentuais. Os aspectos analisados nas duas RSL são as habilidades prévias esperadas dos estudantes de introdução à programação, as dificuldades dos estudantes e os desafios dos docentes.

2. Habilidades prévias dos estudantes

A Tabela 1 apresenta uma comparação das habilidades prévias de que os estudantes precisam para ter sucesso na disciplina de introdução à programação no Ensino Superior e que foram mapeadas nas duas RSL analisadas.

Tabela 1. Comparação das habilidades dos estudantes

| Categoria | Método | Porcentagem das publicações internacionais | Porcentagem das publicações nacionais |
|--------------------------------------|---|--|---------------------------------------|
| Habilidades referentes à programação | Resolução de problemas | 18% | 16% |
| | Raciocínio lógico | 0% | 29% |
| | Abstração | 7% | 17% |
| | Habilidade em matemática | 17% | 17% |
| | Conhecimento prévio em programação, pensamento computacional e ciências | 12% | 9% |
| | Conhecimento básico em inglês | 6% | 0% |
| | Conhecimento básico em português / interpretação de texto | 0% | 9% |
| | Criatividade | 4% | 4% |

| | | | |
|---------------------------------|---|----|----|
| Habilidades educacionais gerais | Pensamento crítico, curiosidade e habilidade de discussão | 4% | 3% |
| | Concentração e memorização | 0% | 3% |
| | Hábito de estudo e pesquisa e gerenciamento de tempo | 2% | 3% |
| | Maturidade | 0% | 1% |

Sobre as habilidades classificadas como “referentes à programação”, o tópico de raciocínio lógico foi mencionado apenas na RSL nacional, em 29% das publicações. Nessas publicações percebemos estudos que abordam o incentivo do professor em estimular o raciocínio lógico-matemático do aluno, além de reforçarem que é uma habilidade que pode ajudar na programação sendo oriunda das disciplinas de matemática no Ensino Básico.

A abstração foi o quarto tópico nas publicações internacionais com 7% do total de artigos e o segundo na nacional com 17% das publicações, uma diferença de 10 pontos percentuais. Nos dois estudos o entendimento da abstração é semelhante e se aproxima ou é descrito como parte da habilidade de pensamento computacional proposta por Wing (2006). O tópico conhecimento prévio em programação, pensamento algorítmico e ciências foi o terceiro mais mencionado no estudo internacional com 12% das publicações e o quarto no estudo nacional com 9%, uma diferença de 3 pontos percentuais. Entretanto, é um tópico relevante para ser discutido, pois na RSL internacional, os estudos apenas mencionam conhecimento prévio em programação. Já na RSL nacional, os estudos discutem também sobre pensamento algorítmico e ciências. Na RSL internacional, os estudos comentam sobre o conhecimento prévio em computação como um preditor de sucesso. Por outro lado, na RSL nacional, aparece um argumento que o conhecimento básico em programação pode facilitar o início das disciplinas introdutórias. Isso pode ser um indício de que é preciso inserir o ensino de programação na Educação Básica de forma universal.

Nos tópicos sobre as habilidades classificadas como “educacionais gerais” não há diferenças significativas a não ser os itens de conhecimento básico de inglês estarem no estudo internacional e o de interpretação de texto e conhecimento em português estarem no estudo nacional - ambos podem ser tratados como equivalentes. Sobre a habilidade de interpretação de texto, os pesquisadores nacionais indicam que esta pode ser importante no auxílio da resolução de problemas de programação. Ainda nas publicações nacionais, essa temática da interpretação de texto aparece em 6 dos 69 artigos, o que pode apontar para um problema grande de formação acadêmica na educação básica em português. Já sobre o conhecimento em inglês, no estudo internacional foi comentado que estudantes fluentes em inglês estão mais propensos a serem bem-sucedidos no ato de programar. Esse último argumento pode ter a ver com as linguagens de programação na sua grande maioria serem realizadas na língua inglesa, e, portanto, os comandos estarão nesse idioma.

3. Dificuldades dos estudantes

A Tabela 2 mostra a comparação em relação às dificuldades dos estudantes encontradas nos dois estudos. As comparações serão feitas seguindo as categorias propostas nas RSL.

Tabela 2. Comparação das dificuldades dos estudantes

| Categoria | Habilidade | Porcentagem das publicações internacionais | Porcentagem das publicações nacionais |
|---------------------------------|--|--|---------------------------------------|
| Formulação do problema | Resolução de problemas | 26% | 30% |
| | Abstração | 7% | 26% |
| | Pensamento algorítmico e raciocínio lógico | 3% | 14% |
| Expressão da solução | Sintaxe da linguagem da programação | 22% | 19% |
| | Estruturas de controle | 13% | 6% |
| | Estruturas de dados | 6% | 3% |
| | Estruturas de código | 4% | 1% |
| Execução da solução e avaliação | Depuração | 15% | 19% |
| | Rastreamento de código | 10% | 0% |
| | Raciocínio simultâneo | 0% | 1% |
| | Validação de soluções a partir de análise crítica a soluções de outros autores | 0% | 1% |
| | Reconhecimento de padrões | 0% | 1% |
| Conhecimento de base | Interpretação de texto / compreensão de enunciados | 0% | 19% |
| | Matemática | 0% | 12% |
| | Inglês | 0% | 3% |
| Comportamento | Motivação e engajamento | 40% | 36% |
| | Gerenciamento de tempo | 17% | 7% |
| | Habilidade de estudo | 17% | 6% |
| | Confiança | 10% | 3% |
| | Percepção de programação como uma disciplina complexa | 7% | 0% |
| | Maturidade | 0% | 4% |
| | Dificuldade de socialização e pedir ajudar | 0% | 3% |
| | Autorregulação | 0% | 3% |
| | Criatividade | 0% | 1% |

Algumas questões ficam evidentes na categoria de “formulação do problema”. A abstração é o segundo tópico mais comentado em ambos os estudos. Porém aparece mais nas publicações nacionais com 26%, uma diferença de 19 pontos percentuais. Na

RSL internacional, os estudos relacionam as questões sobre abstração a dificuldade do estudante de imaginar e compreender termos que não têm equivalente em nosso cotidiano como variável. Na RSL nacional, abstração é usada para comentar sobre outras dificuldades, como baixo conhecimento em matemática. Outro ponto de diferença que interage com abstração é o de pensamento algorítmico e o raciocínio lógico, mencionado em ambos estudos como o terceiro tópico mais comentado. Este tema aparece com uma diferença de 11 pontos percentuais a mais de publicações no estudo nacional. O entendimento do conceito de pensamento algorítmico é o mesmo em ambas RSL, no entanto, no estudo nacional reforçam-se habilidades esperadas do raciocínio lógico como tentar, observar, avaliar e deduzir que não estão sendo bem empregadas no Ensino Superior.

Na categoria “execução da solução e avaliação”, a maior diferença foi rastreamento de código aparecer apenas nas publicações internacionais. Nessas publicações, os autores mencionam que essa habilidade pode ajudar o estudante a ter uma visão mais holística da programação, entendendo o que está acontecendo em cada fase da execução do código, além da importância no processo de aprendizagem sobre programação pelo estudante.

Foi criada uma categoria “conhecimentos de base” para agrupar temas que apareceram apenas nas publicações nacionais. Nossa hipótese é que a falta desses conhecimentos de base torna ainda mais difícil a realidade dos alunos relatada nas publicações nacionais na aprendizagem de introdução à programação, e influencia tópicos da categoria de comportamento, como confiança e motivação.

Já sobre a categoria “comportamento” o tópico de gerenciamento de tempo foi o segundo tópico em ambos estudos, porém teve 10 pontos percentuais a mais de publicações no estudo internacional. Como a aprendizagem de programação é lenta e gradual (Robins, Rountree e Rountree, 2003; Lahtinen, Ala-Mutka e Järvinen, 2005) ambos estudos comentam a dificuldade dos estudantes em organizar tempo suficiente para essa disciplina. Outra diferença é o tópico de habilidade de estudo, terceiro tópico em ambos estudos, com uma diferença de 6 pontos percentuais a mais de publicações no estudo internacional. No estudo nacional, porém, é comentado que o estudante não está preparado para um novo paradigma de estudo, com mais liberdade e responsabilidades no Ensino Superior. Apesar disso, pelo senso comum os estudantes brasileiros não são vistos como organizados e disciplinados. Nossa hipótese para essa diferença é que outras temáticas como o déficit de conhecimentos de base como português e matemática ainda são o foco dos estudos nacionais, enquanto os estudos internacionais conseguem fazer estudos específicos como os direcionados à habilidade de estudo. A percepção da programação como uma disciplina complexa é um tópico abordado apenas no estudo internacional. Comenta-se que essa visão pode afastar e desmotivar estudantes que não sabem o que se espera do profissional formado no curso.

4. Desafios enfrentados pelos professores

Na Tabela 3 é apresentada a comparação dos desafios dos professores mapeados nos dois estudos.

Tabela 3. Desafios enfrentados por professores

| Desafios | Porcentagem das publicações internacionais | Porcentagem das publicações nacionais |
|--|--|---------------------------------------|
| Métodos e ferramentas para o ensino de programação | 37% | 68% |
| Problemas de escalabilidade e infraestrutura | 12% | 33% |
| Manter a motivação, o envolvimento e a persistência dos alunos | 19% | 17% |
| Comunicação professor-aluno e feedback | 18% | 17% |
| Linguagem de programação | 6% | 9% |
| Currículo e sequências instrucionais | 3% | 10% |
| Enfrentamento do conhecimento matemático inadequado dos alunos | 3% | 6% |
| Valorização do erro | 0% | 7% |

Alguns dados chamam atenção. O primeiro tópico, “métodos e ferramentas para o ensino de introdução à programação”, teve uma diferença grande de 31 pontos percentuais. Os argumentos nos dois estudos são semelhantes, principalmente na dificuldade de encontrar dados da efetividade dos métodos de ensino e ferramentas. Nas publicações internacionais, aparece um argumento interessante de repensar os atuais métodos de ensino e da adaptação dos métodos para melhorar a habilidade de resolução de problemas. Já nas publicações nacionais, comenta-se que as ferramentas utilizadas nas disciplinas de introdução à programação foram projetadas para o público infanto-juvenil, como ALICE e Scratch, e estão sendo utilizadas nas universidades. Apesar desse cenário, é notório o esforço dos pesquisadores em ambos estudos na criação e na adaptação de métodos de ensino e ferramentas.

O tópico “problemas de escalabilidade e infraestrutura” teve uma diferença de 21 pontos percentuais. Os problemas mais mencionados no estudo internacional são a heterogeneidade dos alunos, o tamanho das turmas e as limitações de recursos da equipe docente. Nas publicações nacionais percebemos 14 de 69 artigos comentando sobre problemas de infraestrutura, pouco abordados nos estudos internacionais. Fica evidente que os pesquisadores nacionais têm mais dificuldades nesse item com poucos monitores e mentores e problemas de quantidade de computadores nos laboratórios. Um dos argumentos mencionados no problema de escalabilidade nas publicações nacionais é o número grande de estudantes por sala. Porém isso é um argumento questionado na literatura, por exemplo, Ramos e colaboradores (2015) comentam que:

“a realidade brasileira do ensino de CS1 pode ser diferente de outros países, onde grandes turmas com até 200 ou 300 alunos são formadas para palestras e depois os alunos são alocados a tutores para aulas práticas em laboratório. No Brasil, o Ministério da Educação não recomenda turmas com mais de 50 alunos e as aulas de laboratório normalmente são conduzidas pelo próprio professor da disciplina. Assim,

diferenças podem existir entre a efetividade das práticas testadas no exterior e as do nosso país” (2015, p. 319)

O tópico “currículo e sequências instrucionais” teve uma diferença de 7 pontos percentuais. Nos estudos internacionais esse tópico é comentado basicamente quando são citadas as entidades que sugerem modificações de currículo como a *Association for Computing Machinery* (ACM) e o *Institute of Electrical and Electronics Engineers* (IEEE), mas as discussões não são muito aprofundadas. Nas publicações nacionais percebe-se o esforço dos pesquisadores na mudança do currículo e ementas das disciplinas para suprir as dificuldades encontradas atualmente e se adequar às exigências da Sociedade Brasileira de Computação. Nossa hipótese é que essas mudanças realizadas nos currículos nacionais são mais estruturantes e complexas para suprir as carências de formação de base.

O assunto da valorização do erro foi um tópico mencionado por 7% das publicações nacionais e não foi mencionado pelas internacionais. As publicações nacionais reforçam esse assunto como um processo em transformar essa falha em algo natural e intrínseco do processo de aprendizagem do estudante. Nossa hipótese é que o processo cultural e o contexto nacional de educação podem ter influenciado para essa tendência aparecer apenas no estudo nacional.

5. Considerações finais

Esse estudo terciário teve como propósito a análise comparativa de duas RSL - uma internacional e outra nacional - sobre o aprender e o ensinar introdução à programação nos cursos do Ensino Superior, visando a identificar particularidades do contexto brasileiro e assim contribuir para definição adoção de métodos e políticas adequadas a nível nacional, evitando a mera transposição de soluções propostas em países de realidades diferentes.

A temática da resolução de problemas em programação aparece como conceito crucial para a introdução à programação. Embora tal conceito esteja presente de forma sistemática nos dois estudos, suas definições ainda são genéricas, ausentes ou inconsistentes, principalmente quando se discute resolução de problemas em computação. Além disso, as razões para as limitações dos alunos na resolução de problemas não são detalhadas.

Um ponto é que no estudo nacional as dificuldades se concentram mais, proporcionalmente, na etapa de “formulação do problema” do que na “expressão da solução” (Tabela 2). Também há forte discrepância na categoria “comportamento”, com mais problemas relatados na literatura internacional (91%) do que na nacional (63%). Isso indica que a natureza da dificuldade dos alunos brasileiros parece ser outra.

Outro ponto está relacionado à necessidade de melhores métodos e ferramentas específicas para as etapas de formulação de problemas e expressão da solução. Isso pressupõe que o entendimento e a expressão do problema sejam raízes para o aprimoramento do aprendizado de introdução à programação. Porém, mais evidências empíricas são necessárias.

Destaca-se a discussão sobre a deficiência dos conhecimentos prévios dos estudantes, principalmente, em matemática e na sua língua materna – nesse artigo foram analisados problemas nos conhecimentos em inglês e em português. Nesse cenário, esforços para ensinar programação e o pensamento computacional para crianças e adolescentes no Ensino Básico podem ajudar a melhorar esta situação para as futuras gerações de estudantes de programação. Métodos e métricas devem ser estabelecidos para avaliar a eficácia dessa abordagem.

Referências

- Almeida, T. O., e Netto, J. F. De M. (2015). “Robótica Pedagógica Aplicada ao Ensino de Programação: Uma Revisão Sistemática da Literatura”. In Anais do 26 Simpósio Brasileiro de Informática na Educação, p. 597-606.
- Aureliano, V. C. O., e Tedesco, P. C. de A. R. (2012). “Ensino-aprendizagem de Programação para Iniciantes: uma Revisão Sistemática da Literatura focada no SBIE e WIE”. In Anais do 23 Simpósio Brasileiro de Informática na Educação.
- Barbosa, A., Correia, A., Costa, D., e Costa, E. (2015). “Um mapeamento sistemático sobre analisadores de código em disciplinas de programação”. In Anais do 26 Simpósio Brasileiro de Informática na Educação, p. 1235-1244.
- Bennedsen J., e Caspersen, M. E. (2007). “Failure rates in introductory programming”, *ACM SIGCSE Bull.*, vol. 39, no. 2, p. 32–36.
- Da Silva, T. R., Medeiros, T. J. T. J., e Aranha, E. (2014). “Jogos Digitais para Ensino e Aprendizagem de Programação: uma Revisão Sistemática da Literatura”. In Anais do 25o Simpósio Brasileiro de Informática na Educação, p. 692-701.
- Francisco, R. E., Júnior, C. X. P., e Ambrósio, A. P. (2016). “Juiz Online no ensino de Programação Introdutória - Uma Revisão Sistemática da Literatura”. In Anais do 27o Simpósio Brasileiro de Informática na Educação, p. 11-20.
- Gomes, A., e Mendes, A. J. (2015). “A teacher’s view about introductory programming teaching and learning: Difficulties, strategies and motivations”. In Proceedings - Frontiers in Education Conference, FIE.
- Henrique, M. S., e Tedesco, P. C. de A. R. (2017). “Uma Revisão sistemática da Literatura sobre conhecimentos, habilidades, atitudes e competências desejáveis para auxiliar a aprendizagem de programação”. In Anais dos Workshops do VI Congresso Brasileiro de Informática na Educação (WCBIE 2017), p. 1162-1171.
- Kinnunen, P. (2009). “Challenges of teaching and studying programming at a University of Technology—Viewpoints of students, teachers and the University”, Tese de doutorado, Dept. Comput. Sci. Eng., Helsinki Univ. Technol., Espoo, Finland.
- Lahtinen, E., Ala-Mutka, K., e Järvinen, H.-M. (2005). “A study of the difficulties of novice programmers”, *ACM SIGCSE Bulletin*, v. 37, n. 3, p. 14–18.

- Medeiros, R. (2019). “Hello, world: uma análise sobre dificuldades no ensino e aprendizagem de introdução à programação nas universidades”. Tese de doutorado - Universidade Federal de Pernambuco. Recife, PE.
- Medeiros, R., Ramalho, G., e Falcão, T. (2018) “Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education”, IEEE Transaction on Education, Volume 62, Issue 2, p. 77-90.
- Medeiros, R. P., Falcão, T. P., Ramalho, e Lisboa, G. (2020). “Ensino e Aprendizagem de Introdução à Programação no Ensino Superior Brasileiro: Revisão Sistemática da Literatura”. In Workshop sobre Educação em Computação (WEI), p. 186-190.
- Ortiz, J., Pereira, R. (2019). “Ten Years of Initiatives to Promote Computational Thinking: A Systematic Mapping Study”, Journal on Computational Thinking (JCThink), v.3, n.1, p. 95-110.
- Ramos, V., Freitas, M., Galimbert, M., Mariani, A. C., e Wazlawick, R. (2015). “A Comparação da Realidade Mundial do Ensino de Programação para Iniciantes com a Realidade Nacional: Revisão sistemática da literatura em eventos brasileiros”. In Anais do 26 Simpósio Brasileiro de Informática na Educação, p. 318-327.
- Robins, A., Rountree, J., e Rountree, N. (2003). “Learning and Teaching Programming : A Review and Discussion”, Computer Science Education, vol. 13, n. 2, p. 133-172.
- Saico, A., e Saico, P. D. (2016). “Uso de Jogos em Cursos Introdutórios de Programação no Ensino Superior na Área de Computação: Uma Revisão Sistemática”. In Anais do 27 Simpósio Brasileiro de Informática na Educação, p. 549-558.
- Silva, P., Cavalcante, M. T. C., Fachine, J., e Costa, E. (2015). “Um Mapeamento Sistemático sobre Iniciativas Brasileiras em Ambientes de Ensino de Programação”. In Anais do 26o Simpósio Brasileiro de Informática na Educação, p. 367-375.
- Teague, D. (2011). “Pedagogy of introductory computer programming: A people-first approach”. Dissertação de mestrado, Dept. Inf. Syst., Queensland Univ. Technol., Brisbane, QLD, Australia.
- Vihavainen, A., Airaksinen, J. e Watson, C. (2014). “A systematic review of approaches for teaching introductory programming and their influence on success”. In *Proc. ICER*, Glasgow, U.K., p. 19–26.
- Wing, J. (2006). “Computational thinking”. *Commun. ACM*, v. 49, p. 33–35.