

## Caracterizando a pesquisa sobre autoavaliação na aprendizagem de programação para iniciantes

Rozelma Soares de França<sup>1</sup>, Patrícia Cabral de Azevedo Restelli Tesdesco<sup>1</sup>

<sup>1</sup>Centro de Informática – Universidade Federal de Pernambuco (UFPE)  
Caixa Postal 15.064 – 91.501-970 – Recife – PE – Brasil

{rsf2, pcart}@cin.ufpe.br

**Abstract.** *The teaching of programming is considered one of the biggest challenges of Computer Education area and the self-assessment can be integrated into pedagogical practices to improve the learning experience of students. In this context, this paper presents a systematic mapping of the literature that aims to plot the landscape of the research on self-assessment of learning programming for beginners. The results indicate that in recent years there was an increase of interest in the area, that the main research focus is on higher education and that beginners in programming can assess their own learning accurately, which can contribute to effective learning.*

**Resumo.** *O ensino de programação é considerado um dos grandes desafios da área de Educação em Computação e a autoavaliação pode ser integrada em práticas pedagógicas como forma de melhorar a experiência de aprendizagem dos estudantes. Neste contexto, este artigo apresenta um mapeamento sistemático da literatura que visou identificar como se caracteriza a pesquisa em autoavaliação da aprendizagem de programação para iniciantes. Os resultados obtidos mostram que nos últimos anos houve um aumento de interesse pela temática, que a maior parte dos estudos foca no ensino superior e que os iniciantes em programação são capazes de avaliar o próprio aprendizado com precisão, o que poderá contribuir para a efetivação da aprendizagem.*

### 1. Introdução

As disciplinas introdutórias de programação são a base para o desenvolvimento lógico e algorítmico de graduandos da área de Computação. Ainda, são essenciais para a construção da fundamentação necessária para o entendimento de tópicos mais avançados de disciplinas relacionadas. Contudo, ensinar programação é dos grandes desafios da educação em Computação (CASPERSEN; KÖLLING, 2009) e pesquisas apontam que são várias as dificuldades enfrentadas pelos estudantes ao iniciarem seus estudos na área.

Nesse contexto, é fundamental encarar a avaliação e a aprendizagem como intimamente relacionadas, assumindo a avaliação como parte integrante do processo de aprendizagem (BORRALHO; FIALGO, 2012). Assim, mais que enxergar a avaliação como um instrumento com caráter somativo, destinado a certificar a aprendizagem dos estudantes no final do semestre, por exemplo, é necessário tê-la como um instrumento *para a aprendizagem*, em que os educandos desempenhem um papel mais ativo no processo avaliativo, numa atividade crítica constante.

Nesta perspectiva, segundo Earl (2003), os estudantes poderão monitorar a sua própria aprendizagem e usar o *feedback* deste monitoramento para fazer ajustes e adaptações na forma de aprender. Isto exigirá que o professor ajude o estudante a desenvolver, praticar e tornar-se confortável com a reflexão e com uma análise crítica da sua própria aprendizagem. Neste cenário, a autoavaliação pode assumir um importante papel no processo avaliativo (EARL;

KATZ, 2006). Segundo Boud (2013), a autoavaliação é uma habilidade necessária para a aprendizagem ao longo da vida. Além disso, ela tem um papel importante na efetivação da aprendizagem.

No contexto da Computação, evidências empíricas apontam que práticas de autoavaliação podem ser integradas com sucesso, melhorando a experiência de aprendizagem dos educandos (NGAI *et al.*, 2010). Todavia, ao analisar-se a literatura da área não se constata a publicação de estudos que apresentem um panorama sobre o assunto, sendo necessário, portanto, a análise da produção científica sobre o processo de autoavaliação da aprendizagem de Computação, especialmente em disciplinas introdutórias de programação. Desse modo, neste trabalho são apresentados os resultados de um mapeamento sistemático de literatura que contou com a análise de estudos publicados em importantes eventos e periódicos nacionais e internacionais que abordam Educação em Computação e/ou Informática na Educação.

Assim, além desta seção introdutória, este artigo está organizado como segue: na Seção 2 é apresentado o método de pesquisa empregado neste estudo; na Seção 3 os resultados obtidos são apresentados, segmentados em três categorias: experiências didáticas, ferramentas e estudos teóricos. Por fim, na Seção 4 os resultados são discutidos e as considerações finais são apresentadas.

## 2. Metodologia

O objetivo desta pesquisa é identificar tendências e lacunas centrais na pesquisa sobre autoavaliação na aprendizagem de programação para iniciantes a partir da produção científica nacional e internacional de modo a contribuir com o avanço da pesquisa na área. O método adotado foi o de mapeamento sistemático da literatura o qual, segundo Kitchenham *et al.* (2010), representa uma forma particular de revisão sistemática e identifica e analisa, de maneira mais geral, a pesquisa sobre um tópico específico, fornecendo evidências para questões mais amplas relacionadas a tendências de pesquisa.

### 2.1. Questões de pesquisa

Este estudo foi norteado pela seguinte questão central de pesquisa:

- **Q:** Qual é estado atual da pesquisa sobre autoavaliação na aprendizagem de programação para iniciantes?

Para orientar os processos de seleção de estudos, extração e análise dos dados, síntese e apresentação dos resultados, as questões específicas de pesquisas foram elaboradas e são listadas a seguir:

- **Q1.1:** Como a autoavaliação tem sido evidenciada em experiências didáticas no processo de ensino-aprendizagem de programação para iniciantes?
- **Q1.2:** Como a autoavaliação tem sido apoiada em tecnologias educacionais voltadas ao processo de ensino-aprendizagem de programação para iniciantes?
- **Q1.3:** Com a autoavaliação tem sido discutida em estudos teóricos que tratam do processo de ensino-aprendizagem de programação para iniciantes?

### 2.2. Critérios de inclusão e exclusão

Foram incluídos os trabalhos científicos que tratavam de autoavaliação na aprendizagem de programação para iniciantes. Os estudos que atenderam a pelo menos um dos critérios de exclusão listados a seguir foram retirados do processo de análise:

- Artigos não acessíveis na íntegra, considerando a disponibilidade fornecida pelo Portal da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) ou pela oferta gratuita na web;
- Arquivos no formato .ppt, .pptx, palestras, relatórios de workshop, tutoriais, relatórios técnicos e resumos;
- Estudos secundários e terciários;
- Estudos que tratam da autoavaliação apenas como trabalhos futuros;
- Estudos que não sejam claramente sobre a autoavaliação no contexto do ensino-aprendizagem de programação para iniciantes.

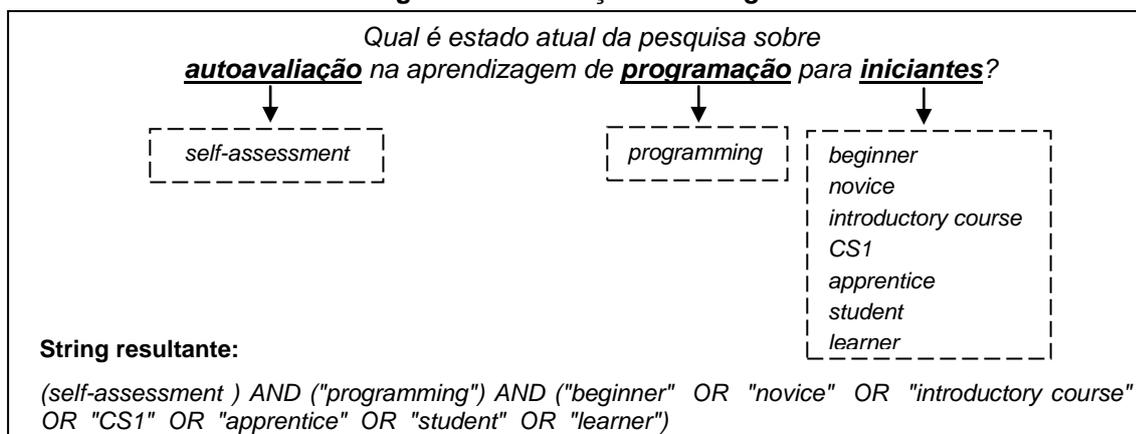
### 2.3. Fontes de dados, estratégia de busca e seleção dos estudos

O processo utilizado para procura por estudos primários foi o de busca eletrônica e manual. Foi realizada busca manual nos anais dos últimos cinco anos do Simpósio Brasileiro de Informática na Educação (SBIE), do Workshop de Informática na Escola (WIE), do Workshop sobre Educação em Computação (WEI) e da Revista Brasileira de Informática na Educação (RBIE), considerados os principais eventos e revista que tratam de ensino de computação e informática na educação no Brasil.

A busca automática, por sua vez, foi realizada em dois engenhos de busca, o ACM Digital Library e o Elsevier Science Direct, refinando seus resultados aos principais eventos e revistas internacionais da área. São eles: ACM Technical Symposium on Computer Science Education (SIGCSE), Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE), International Computing Education Research Conference (ICER), Australasian Computing Education Conference (ACE), Koli Calling International Conference on Computing Education Research (Koli Calling), ACM Transactions on Computing Education (TOCE), Computers and Education (C&E).

A *string* utilizada na busca automática foi construída a partir de três termos extraídos da questão geral de pesquisa, como ilustrado na Figura 1. Considerando os sinônimos para os termos encontrados na literatura, a *string* de busca foi construída pela combinação desses termos os quais foram concatenados através dos operadores booleanos “OR” e “AND”.

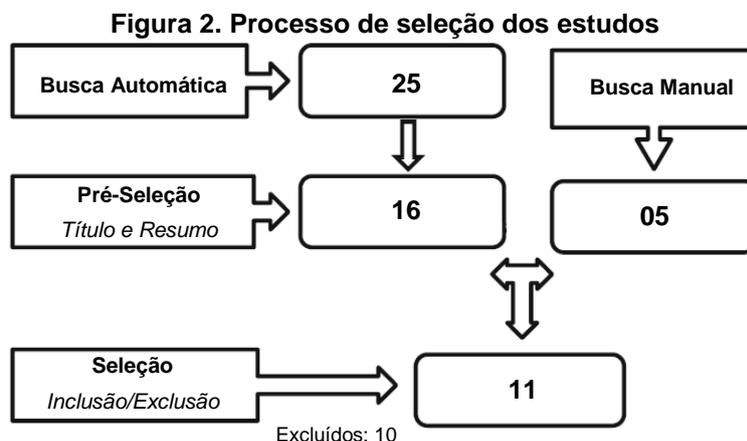
**Figura 1. Construção da *string* de busca**



Na busca manual, foram verificados os sites contendo os anais dos referidos eventos e revista acessando os artigos completos e resumos expandidos, lendo seus títulos e resumos e

excluindo os estudos não relevantes a esta pesquisa. Os artigos pré-selecionados tiveram sua introdução e conclusão lidas e foram excluídos os considerados irrelevantes a esta investigação.

A lista de estudos retornados na busca automática também foi analisada. Os critérios de inclusão e exclusão foram aplicados a cada um dos trabalhos após a leitura de seus títulos, resumos, introdução e conclusão. Este processo selecionou os documentos considerados relevantes para extração e análise dos dados, como demonstrado na Figura 2.



#### 2.4. Classificação dos estudos

Os estudos retornados foram lidos na íntegra e classificados em três categorias:

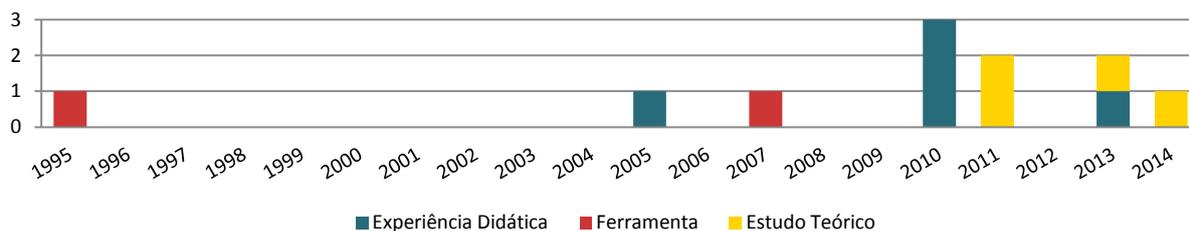
- *Experiência Didática*: o estudo aplica ou propõe atividades para iniciantes em programação havendo validação dos resultados da experiência;
- *Ferramenta*: o estudo propõe uma ferramenta para promover a autoavaliação e apresenta uma validação da proposta;
- *Estudo Teórico*: o trabalho traz uma argumentação teórica baseada em pesquisa exploratória, documental ou relato de experiência. Propostas de atividades didáticas ou ferramentas, sem validação com estudantes ou professores, também são enquadradas nesta categoria.

#### 2.5. Processo de extração dos dados

Para cada artigo selecionado foram extraídos seu título, autor(es), onde foi publicado, ano de publicação e instituição do(s) autor(es). Além disso, para os estudos classificados como experiência didática foram extraídas informações sobre o material didático ou metodologia usada na intervenção, objetivo da validação da experiência, técnica de coleta de dados, nível de ensino e resultados obtidos. Para os trabalhos classificados como ferramenta foram extraídos os objetivos da ferramenta e de sua validação, bem como as técnicas de coleta de dados utilizadas, o nível de ensino dos participantes e os principais resultados alcançados. Já os estudos teóricos foram analisados, verificado o nível de ensino para o qual as atividades são destinadas e sintetizando os seus argumentos.

### 3. Resultados

O gráfico apresentado na Figura 3 apresenta a distribuição temporal dos estudos incluídos nesta revisão. Como pode ser observado, o primeiro estudo sobre autoavaliação em programação, incluído neste mapeamento, foi publicado em 1995 e o interesse pela pesquisa na área aumentou nos últimos anos.

**Figura 3. Distribuição temporal dos estudos selecionados**

Nas subseções seguintes, os dados extraídos dos estudos selecionados serão analisados e a apresentação dos resultados será segmentada considerando as três categorias descritas anteriormente: experiências didáticas, ferramentas e estudos teóricos. A organização dos resultados de acordo com tal classificação possibilitará responder as perguntas específicas de pesquisa definidas neste estudo.

### 3.1. Experiências didáticas

#### 3.1.1. Nível de escolaridade

Os estudos foram classificados de acordo com o nível de escolaridade para os quais as pesquisas se destinam. A partir desta análise, foi constatado que todos os estudos classificados como Experiência Didática sobre autoavaliação foram realizados no contexto do ensino superior. Há, desse modo, uma carência de estudos que tratam da autoavaliação da aprendizagem de programação para estudantes que estejam cursando os níveis de ensino fundamental e médio.

#### 3.1.2. Materiais didáticos e metodologias utilizadas

Foram identificados, a partir dos estudos classificados como Experiência Didática, os materiais e metodologias utilizadas nas intervenções e os resultados são apresentados na Tabela 1. Como pode ser observada, a maior parte das pesquisas utiliza formulários ou questionários como insumos na formação de iniciantes em programação na habilidade de autoavaliação. No trabalho de Ngai *et al.* (2010) é proposto um formulário que possibilita ao estudante avaliar-se em três dimensões de proficiências em programação: depuração, codificação e programação. Já Alaoutinen (2010), Alaoutinen *et al.* (2010) e Honig (2013) utilizaram questionários para que os aprendizes classificassem suas habilidades em programação utilizando a taxonomia de Bloom. Por outro lado, o sistema QuizPACK é usado por Brusilovsky *et al.* (2005) para envolver os estudantes na autoavaliação do conhecimento de programação.

**Tabela 1. Materiais e metodologias usadas na autoavaliação da aprendizagem de programação**

MATERIAL / METODOLOGIA	OCORRÊNCIAS
Formulário/Questionário	4
QuizPACK	1

#### 3.1.3. Objetivos das pesquisas

Quanto aos objetivos dos estudos categorizados como Experiência Didática, eles são diversos. Na pesquisa de Ngai *et al.* (2010) objetivou-se verificar se há correlação entre a autoavaliação dos estudantes e avaliação do docente, observando também se há relação entre o desempenho

dos educandos nas avaliações contínuas, verificado pelo professor, e o desempenho deles no teste final.

Já Alaoutinen *et al.* (2010) estavam interessados em saber se os estudantes são capazes de avaliar a própria aprendizagem e de reconhecer o correspondente nível de conhecimento na taxonomia de Bloom. De modo complementar a este estudo, Alaoutinen (2010) examina os fatores que afetam a precisão da avaliação e o desempenho dos estudantes em um curso introdutório de programação.

Por outro lado, Honig (2013) objetivava identificar o interesse dos educandos sobre um ambiente de programação visual, o App Inventor, e se com o uso de tal ambiente era possível alcançar os objetivos de aprendizagem pretendidos. Nesse contexto, a autoavaliação foi utilizada para verificar como os aprendizes avaliam o próprio conhecimento e tais resultados foram comparados com a avaliação docente. Já Brusilovsky *et al.* (2005) buscavam envolver os aprendizes no uso do sistema QuizPACK para sua autoavaliação do conhecimento.

#### 3.1.4. Técnicas de coleta de dados

As técnicas de coleta de dados utilizadas nas Experiências Didáticas foram levantadas e são apresentadas na Tabela 2. Cada estudo utilizou um ou mais instrumentos e, como pode ser observado, boa parte deles utilizou formulários/questionários de autoavaliação, por exemplo, para coletar dados durante as intervenções didáticas no contexto de programação.

As notas obtidas pelos estudantes ao final do curso ou por meio de pré e pós-teste também têm sido utilizadas, sendo esta última em menor recorrência. Constatou-se também a análise de *logs* de um sistema em uma experiência didática na autoavaliação da aprendizagem de programação.

**Tabela 2. Técnicas de coleta de dados usadas nas experiências didáticas de autoavaliação da aprendizagem de programação**

TÉCNICAS	OCORRÊNCIAS
Formulário/Questionário	4
Logs do sistema	1
Notas	4
Pré e Pós-Teste	1

#### 3.1.5. Resultados obtidos

Ao analisarem-se os resultados obtidos pelos estudos categorizados como Experiência Didática constata-se efeitos positivos da autoavaliação na aprendizagem de iniciantes em programação. Ngai *et al.* (2010) encontraram uma forte correlação entre a autoavaliação dos estudantes e avaliação do docente, o que sugere que os aprendizes são capazes de entender os critérios de classificação e avaliar-se de forma justa e objetiva.

Alaoutinen *et al.* (2010) constataram que, em geral, os estudantes parecem ser bastante precisos na avaliação do próprio conhecimento, embora os mais jovens tenderem a superestimar ou subestimar suas habilidades. Com o uso da taxonomia de Bloom no questionário de autoavaliação, eles também identificaram que os níveis mais inferiores da taxonomia parecem ser mais fáceis de distinguir pelos estudantes. Seus resultados ainda apontam que há correlação estatisticamente significativa entre a autoavaliação e os resultados obtidos nos exames do curso.

Alaoutinen (2010) identificou os estilos de aprendizagem dos estudantes de acordo com o modelo de Felder e Silverman (1988) e constatou que os estilos de aprendizagem ativo, sensorial, visual e sequencial estão correlacionados com uma autoavaliação mais precisa. Os resultados também mostram que estudantes reflexivos foram melhores em programação que os ativos e que a experiência educacional faz com que a autoavaliação seja mais precisa, enquanto a idade não tem efeito. Para alguns educandos, foi difícil colocar os seus conhecimentos sobre a escala de autoavaliação utilizada e eles gostariam de boas instruções e exemplos para fazer isso.

Já os resultados alcançados por Honig (2013) apoiam o uso de categorias cognitivas, como as da taxonomia de Bloom, na avaliação da aprendizagem em Ciência da Computação. Ao comparar a autoavaliação dos estudantes com a avaliação do professor o autor percebeu que as diferenças foram maiores nos três primeiros níveis da taxonomia. Em suma, Brusilovsky *et al.* (2005) observaram, com as intervenções, ganho na aprendizagem dos estudantes e aumento na interação com o sistema QuizPACK.

### 3.2. Ferramentas

#### 3.2.1. Nível de escolaridade

Os estudos categorizados como Ferramenta sobre autoavaliação no contexto de programação para iniciantes têm como foco o nível superior. Assim, novamente, os resultados sugerem que há uma lacuna de pesquisa sobre o tema no contexto da educação básica.

#### 3.2.2. Ferramentas propostas

Dois estudos foram incluídos na categoria Ferramenta. Um deles foi realizado por Kay *et al.* (2007) que propõem o Reflect, um sistema inteligente de ensino de programação que tem como objetivo promover a reflexão mostrando ao estudante um modelo de seu progresso no processo de autoavaliação. Nele, o processo de autoavaliação dá-se da seguinte forma: O estudante lê e avalia exemplos de soluções fornecidos e já avaliados pelo professor. Em seguida, ele deve fornecer e avaliar sua solução a um determinado problema utilizando critérios que o professor definiu para essa tarefa. Na apresentação do problema há uma lista de objetivos de aprendizagem que o professor gostaria que os educandos atingissem com a tarefa proposta. Ao ler e avaliar exemplos de soluções, bem como escrever a sua própria resposta ao problema, os estudantes precisam prestar atenção a tais objetivos. As avaliações dos estudantes são comparadas com a avaliação do professor e a discrepância entre elas indica o quão bem os aprendizes compreenderam os objetivos de aprendizagem. A partir disso, o modelo de aluno é atualizado e os estudantes podem explorar o porquê de o professor pensar diferente deles e aprender como o educador resolve os problemas, de modo a alcançarem um novo nível de compreensão das questões propostas.

Outra ferramenta é descrita por Schorsch (1995). Ela fornece *feedbacks* ao estudante sobre erros em seu código, como erros lógico e de estilo, por exemplo. Os *feedbacks* informam o que está errado, incluindo uma amostra de código correto como exemplo para o aprendiz. O objetivo é que com as mensagens de diagnóstico relatadas pelo sistema o estudante avalie o próprio conhecimento, localizando e corrigindo os erros identificados.

#### 3.2.3. Objetivos das pesquisas

No que se refere aos objetivos dos trabalhos classificados como Ferramenta, constatou-se que Kay *et al.* (2007) estavam interessados em obter uma melhor compreensão do comportamento da aprendizagem dos estudantes ao usar o sistema Reflect. Já Schorsch (1995) desejava

identificar se com o uso do sistema proposto os estudantes haviam aprendido a utilizar habitualmente um adequado estilo de programação.

#### 3.2.4. Técnicas de coleta de dados

Em relação às técnicas de coleta de dados utilizadas nos estudos categorizados como Ferramenta, elas são exibidas na Tabela 3. Observou-se que os *logs* obtidos a partir da interação do estudante com os sistemas têm sido usados no processo de coleta e análise de dados. Ainda, entrevistas com estudantes e/ou professor são realizadas. Também, as notas obtidas pelos educandos ao fim do estudo têm sido consideradas.

**Tabela 3. Técnicas de coleta de dados usadas nas validações de ferramentas para autoavaliação da aprendizagem de programação**

TÉCNICAS	OCORRÊNCIAS
Entrevista	1
Logs do sistema	2
Notas	1

#### 3.2.5. Resultados obtidos

Na categoria Ferramenta, os resultados obtidos por Kay *et al.* (2007) sugerem que o sistema Reflect ajuda os estudantes a compreender os objetivos do professor para as tarefas definidas e a refinar o seu próprio trabalho em relação a essas metas.

Já na pesquisa de Schorsch (1995) observou-se que a maioria dos estudantes que utilizaram seu sistema demonstrou incorporar melhores práticas de programação. Também se identificou que alguns aprendizes não leem as mensagens de erro totalmente. Por outro lado, os professores notaram um aumento acentuado da qualidade dos códigos produzidos.

### 3.3. Estudos teóricos

#### 3.3.1. Nível de escolaridade

Assim como os estudos classificados como Experiências Didáticas e Ferramentas, a maioria dos Estudos Teóricos tem como foco estudantes do ensino superior. Somente o trabalho de França e Tedesco (2014) dá suporte à autoavaliação de programação a educandos do ensino médio.

#### 3.3.2. Síntese

Nos estudos teóricos mapeados, Lewis *et al.* (2011) apresentam um modelo de cinco fatores que influenciam as decisões de estudantes de se formar em Computação. No estudo, é descrito como os educandos medem sua capacidade em termos de velocidade, notas e experiência anterior em programação e como eles fazem interpretações e decisões com base nestas medições.

Tendo em vista que fatores que contribuem para o sucesso ou o insucesso podem afetar a autoeficácia, motivação, sucesso e a permanência de estudantes em um curso de Ciência da Computação, a pesquisa de Vivian *et al.* (2013) investiga quais fatores os alunos percebem como sendo as causas de seu sucesso ou insucesso na aprendizagem introdutória de programação. Já no estudo de Sirotheau *et al.* (2011) é proposta uma ferramenta que dá suporte

ao *feedback* colaborativo como forma de estimular a prática de autoavaliação nos iniciantes em programação.

Outro Estudo Teórico é apresentado por França e Tedesco (2014). O trabalho trata da autoavaliação e da avaliação por pares envolvendo programadores iniciantes. Em relação à autoavaliação, em suma, ela é realizada explicitamente antes e após a resolução de um problema de lógica de programação possibilitando que os estudantes pensem sobre si mesmos como solucionadores de problemas e reflitam sobre suas experiências contínuas de aprendizagem.

#### **4. Discussão e Considerações Finais**

Várias são as dificuldades encontradas por estudantes no processo de ensino-aprendizagem de introdução à programação, o que vem despertando o interesse de pesquisadores e professores por estratégias pedagógicas que minimizem tais problemas. Nesse contexto, a autoavaliação tem sido vista como uma abordagem eficaz por possibilitar que os educandos identifiquem as causas dos seus próprios erros e acertos acadêmicos pela análise crítica da própria aprendizagem.

Tendo em vista a aparece ausência de estudos que tracem um panorama sobre tal tema, este trabalho apresentou os resultados de um mapeamento sistemático realizado a partir da produção científica divulgada nos principais eventos e periódicos sobre Educação em Computação e Informática na Educação, objetivando-se com isso obter-se um panorama da pesquisa sobre autoavaliação na aprendizagem de programação para iniciantes.

A partir dos resultados obtidos com o mapeamento sistemático pode-se inferir que há a necessidade de pesquisas que tratem da autoavaliação na aprendizagem de programação para estudantes da educação básica. Nesta investigação, constatou-se que todos os estudos mapeados têm como público-alvo o ensino superior, exceto o estudo teórico apresentado por França e Tedesco (2014). Contudo, considerando-se a necessidade de promover o pensamento computacional na educação básica (WING, 2006), é mister investigar também como estudantes dos níveis fundamental e médio podem se beneficiar de atividades reflexivas durante a aprendizagem em conceitos computacionais.

Os resultados obtidos ainda revelam que há uma diversidade de temas objetos de interesse dos estudos e seus resultados apoiam a prática da autoavaliação no contexto de programação para iniciantes. Dentre os benefícios que podem ser alcançados destaca-se o impacto sobre a aprendizagem dos educandos.

Em relação aos materiais didáticos utilizados, foi observada, de forma incipiente, a proposição e utilização de ferramentas de suporte à autoavaliação de programação, podendo, investigações futuras, tratar com maior recorrência do uso e construção de recursos tecnológicos em suas intervenções didáticas. Ainda tendo em vista o uso de tecnologias educacionais no apoio à autoavaliação, pesquisadores podem investigar sobre como os dados obtidos a partir das interações dos estudantes com os sistemas podem ser tratados de modo a favorecer a aprendizagem dos envolvidos.

Analisando-se os estudos obtidos a partir de fontes de dados nacionais, constata-se que todos aqueles incluídos neste mapeamento podem ser classificados como estudos teóricos, por apresentarem pesquisas com caráter descritivo, tais como a proposição de ferramentas sem a aplicação de método empírico para avaliação de suas propostas. Diante do cenário exposto, percebe-se que a pesquisa sobre autoavaliação na aprendizagem de Computação, especialmente introdução à programação, ainda está desenvolvendo, principalmente no contexto brasileiro,

e os resultados obtidos neste trabalho poderão direcionar investigações e discussões sobre a temática apresentada.

## 5. Agradecimentos

Rozelma Soares de França agradece à CAPES pela concessão de bolsa de mestrado no período de realização desta pesquisa.

## Referências

- Alaoutinen, S. (2010). Effects of learning style and student background on self-assessment and course performance. In *Proceedings of the 10th Koli Calling International Conference on Computing Education Research*, 5-12.
- Alaoutinen, S., & Smolander, K. (2010). Student self-assessment in a programming course using bloom's revised taxonomy. In *Proceedings of the 15th ITiCSE*, 155-159.
- Borrvalho, A., Fialho, I., & Cid, M. (2012). Aprendizagem no ensino superior: relações com a prática docente. *Ensino Superior - Inovações e qualidade na docência*. Porto, Portugal.
- Boud, D. (2013). *Enhancing learning through self-assessment*. Routledge.
- Brusilovsky, P., & Sosnovsky, S. (2005). Engaging students to work with self-assessment questions: A study of two approaches. In *Proceedings of the 41st SIGCSE*, 251-255.
- Caspersen, M. E., & Kolling, M. (2009). STREAM: A first programming process. *ACM Transactions on Computing Education (TOCE)*, 9(1), 4.
- Earl, L. (2003). *Assessment as Learning: Using Classroom Assessment to Maximize Student Learning*. Thousand Oaks, CA: Corwin.
- Earl, L., & Katz, S. (2006). *Rethinking classroom assessment with purpose in mind: assessment for learning, assessment as learning and assessment of learning*. Manitoba Education, Citizenship and Youth.
- Felder, R. M., & Silverman, L. K. (1988) *Learning and teaching styles in engineering education*. *Engineering education*, 78(7), 674-681.
- França, R. S., & Tedesco, P. C. D. A. R. (2014). Um modelo colaborativo para a aprendizagem do pensamento computacional aliado à autorregulação. In *Anais do SBIE*, 25(1), 1133-1142.
- Honig, W. L. (2013) Teaching and assessing programming fundamentals for non majors with visual programming. In *Proceedings of the 18th ITiCSE*, 40-45.
- Kay, J., Li, L., & Fekete, A. (2007) Learner reflection in student self-assessment. In *Proceedings of the 9th ACE*, 66, 89-95.
- Kitchenham, B. A., Budgen, D., & Brereton, P. (2010). The value of mapping studies: a participantobserver case study. In *Proceedings of the 14th EASE*, 25-33.
- Lewis, C. M., Yasuhara, K., & Anderson, R. E. (2011) Deciding to major in computer science: a grounded theory of students' self-assessment of ability. In *Proceedings of the 7th ICER*, 3-10.
- Ngai, G., Lau, W. W., Chan, S. C., & Leong, H. V. (2010). On the implementation of self-assessment in an introductory programming course. In *Proceedings of the ACM SIGCSE Bulletin*, 41(4), 85-89.
- Schorsch, T. (1995) CAP: an automated self-assessment tool to check Pascal programs for syntax, logic and style errors. In *Proceedings of the ACM SIGCSE*, 168-172.
- Sirotheau, S., Brito, S. R., Silva, A. S., Eliasquevici, M. K., Favero, E. L., & Tavares, O. L. (2011). (2011) Aprendizagem de iniciantes em algoritmos e programação: foco nas competências de autoavaliação. In *Anais do SBIE*, 22(1).
- Vivian, R., Falkner, K., & Falkner, N. (2013) Computer science students' causal attributions for successful and unsuccessful outcomes in programming assignments. In *Proceedings of the 13th Koli Calling International Conference on Computing Education Research*, 125-134.
- Wing, J. M. (2006) Computational thinking. *Communications of the ACM*, 49(3), 33-35.