

Uma Revisão sistemática da Literatura sobre conhecimentos, habilidades, atitudes e competências desejáveis para auxiliar a aprendizagem de programação

Mychelline Souto Henrique¹, Patrícia C. de A. R. Tedesco¹

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

{msh,pcart}@cin.ufpe.br

Abstract. *Learning programming can become more efficient, so it is desirable to the requirements, knowledge, attitudes and / or skills. In literature one of the most quoted skills is logical reasoning. Investigate the aspects of the purpose of this article. The method used is a Systematic Review of Literature (RSL), articles published without SBIE (2012 to 2016) and WalgProg (2015 to 2016). After three steps of the RSL, 26 articles were included in the research. Among the knowledge cited are: the mathematical, computational, syntax and semantics of the language. Some skills mentioned are: know how to correct possible errors, mentally calculate the output of the code from a specified input. About attitudes, some cited are: the ability to reflect on mistakes and take time to practice. Finally, as the most cited competencies are: creativity and capacity for abstraction.*

Resumo. *A aprendizagem de programação pode se tornar mais efetiva, para isso é desejável que os alunos possuam alguns conhecimentos, habilidades, atitudes e/ou competências. Na literatura uma das habilidades mais citadas, é o raciocínio lógico. Investigar sobre esses aspectos é o objetivo deste artigo. O método utilizado é uma Revisão Sistemática da Literatura (RSL), foram investigados artigos publicados no SBIE (2012 à 2016) e WalgProg (2015 à 2016). Após as três etapas da RSL, 26 artigos foram incluídos na pesquisa. Dentre os conhecimentos citados, estão: o matemático, o computacional, a sintaxe e semântica da linguagem. Algumas habilidades citadas, são: saber corrigir os possíveis erros, calcular mentalmente a saída do código a partir de uma determinada entrada. Em relação às atitudes, algumas são: capacidade de refletir sobre os erros e reservar um tempo destinado a praticar. Por fim, as competências mais citadas são: criatividade e capacidade de abstração.*

1. Introdução

O ensino de programação e suas complexidades têm sido tema de vários trabalhos nacionais. Há muito tempo procura-se encontrar alternativas que auxiliem o aluno a compreender conceitos relacionados a programação de computadores.

Segundo Pimentel, França e Omar (2003): “a mesma aula é dada para quem sabe muito, pouco, ou nada sobre determinado tópico. Isto gera um círculo de injustiças que

condena muitos alunos à não-aprendizagem, gerando sucessivas reprovações e/ou colocando no mercado de trabalho profissionais sem uma formação consistente”.

A discussão sobre o ensino de programação e os “conhecimentos” necessários para que os alunos obtenham um resultado satisfatório, tem sido alvo de pesquisa há muitos anos. Percebe-se que é preciso definir claramente o que é conhecimento, habilidade, atitude e competência. Desta forma o professor poderá compreender melhor os aspectos que estão relacionados ao processo de aprendizagem, a fim de criar alternativas viáveis para estimular no aluno tais “conhecimentos”, caso isso seja possível.

Na área da educação existem poucas iniciativas para mapear competências relacionadas a uma organização curricular. O trabalho de Rezende, Segre e Campos (2004) afirma que:

“A literatura encontrada registra apenas princípios básicos para uma pedagogia baseada em competências (BURNIER, 2004), listagem de competências e habilidades a serem desenvolvidas no ensino profissional superior, por área de conhecimento (BRASIL, 2001), uma metodologia para o estabelecimento de perfis profissionais para certificação profissional baseada em competências (BRASIL, 2000) e um conjunto básico de habilidades agrupadas em diversas competências básicas, as quais formam a base para uma educação orientada para a competência de crianças e jovens até cerca de dezoito anos de idade (CHAVES, 2004)”.

Os autores Rezende, Segre e Campos (2004) propõem em seu trabalho um método interativo para realizar o mapeamento de competências, que é dividido em três etapas: (i) construção da matriz de competência por IES; (ii) construção da matriz de competência das empresas, por área de conhecimento; (iii) cruzamento da matriz de competência da IES e da empresa com o projeto pedagógico.

Diante do cenário exposto, o objetivo deste artigo é investigar quais os conhecimentos, habilidades, atitudes e competências são citadas em trabalhos nacionais sobre o processo de ensino-aprendizagem de programação.

O restante deste artigo está organizado da seguinte forma. A Seção 2 apresenta o referencial teórico, que trata sobre os conceitos relacionados à aprendizagem e trabalhos relacionados ao tema. A Seção 3 apresenta a metodologia utilizada para alcançar o objetivo deste trabalho, que é baseada em uma RSL. A Seção 4 aborda os resultados, que são baseadas em cinco perguntas de pesquisa. Por fim, a Seção 5 apresenta as conclusões e trabalhos futuros.

2. Referencial teórico

2.1 Aprendizagem

Não há um consenso sobre o conceito de aprendizagem (Illeris 2015). Existem várias teorias similares, cujo conceito mais tradicional é a aquisição de conhecimentos e habilidades. Em uma visão mais atual, a aprendizagem trata de dimensões emocionais e sociais (desenvolver competências). Existem dois processos distintos durante a aprendizagem: o processo externo (interação entre o indivíduo e o ambiente) e o processo psicológico interno (elaboração e aquisição).

A Tabela 1 apresenta as três etapas do processo de aprendizagem. Elas foram utilizadas para classificar os trabalhos selecionados na RSL. O propósito é investigar em qual das etapas existem mais iniciativas no ensino de programação.

Tabela 1. Etapas do processo de aprendizagem

Processo de aprendizagem		
Etapa de aquisição (AQ)	Etapa de retenção (R)	Etapa de aplicação (A)
Contato com a informação	Memorização da informação adquirida	Aplica a informação na prática
Entender o que está lendo	Memória de longo prazo	Realizar testes/ Notas
Memória curto prazo/ forma superficial		Preparar uma apresentação

2.2 Conceito de conhecimento, habilidade, atitude e competência

Quando se trata de conhecimento, habilidade, atitude e competência, o campo de estudo relacionado é a administração, ou seja, o ambiente empresarial. De acordo com Chiavenato (2004):

- Conhecimento (Saber): é o resultado de aprender a aprender e aumentar continuamente o conhecimento.
- Habilidade (Saber como fazer): consiste em utilizar conhecimentos, métodos, técnicas e equipamentos necessários para o desempenho de tarefas específicas, por meio da experiência e educação.
- Atitude (Saber fazer acontecer): é o comportamento ativo e proativo, ele nos leva a exercitar a habilidade de um determinado conhecimento.

A competência é a junção do conhecimento, habilidade e atitudes. Conforme afirma Brasil/CNE (1999) competência é “a capacidade de mobilizar múltiplos recursos numa mesma situação, entre os quais os conhecimentos adquiridos na reflexão sobre as questões pedagógicas e aqueles construídos na vida profissional e pessoal, para responder às diferentes demandas das situações de trabalho”.

Os termos: habilidade e competência é mencionado pelo pesquisador europeu Perrenoud (1997) e o norte-americano Carroll (1993), porém segundo a afirmação de Primi et al. (2001): “o conceito associado aos termos difere para ambos, tendo em vista que o que os europeus denominam como competência, os norte-americanos têm denominado como habilidades cognitivas”. Portanto, percebe-se, que tais conceitos podem utilizar termos distintos.

2.3 Habilidades e Competências no Curso de Ciência da Computação

O trabalho de Rezende, Segre e Campos (2007) lista algumas competências pretendidas para os egressos do curso de computação. Um estudo de caso foi realizado envolvendo três IES e três empresas do Rio de Janeiro. Os resultados classificam as competências em: cognitivas técnicas e metodológicas, pessoais, sociais e as mistas.

Pimentel et al. (2003) apresentaram em seu trabalho um modelo de avaliação das competências e habilidades na disciplina de programação de forma individualizada. Um Sistema Tutores Inteligentes (STI) foi utilizado para realizar um acompanhamento contínuo da aprendizagem e prover um ambiente personalizado. O trabalho não deixa claro quais são as habilidades e competências desejáveis, apenas uma pirâmide com níveis de conhecimento é apresentada, eles são classificados em: extra, complementar e exigido.

O trabalho de Silva et al. (2015) consiste em uma RSL sobre ensino - aprendizagem de programação. A quinta pergunta de pesquisa (QP5 - Quais habilidades e competências estão sendo exploradas nos alunos) é de interesse deste trabalho. Os resultados classificam as competências em: cognitiva, social e emocional. Nas habilidades cognitivas foram apresentadas as seguintes características: resolver problemas, estabelecer conclusões lógicas, planejar e tomar decisões. Em relação às

habilidades sociais, as características apresentadas foram: lidar com regras, cooperar e elaborar. As habilidades emocionais destacam: autoconfiança, autoestima e autoavaliação.

É possível encontrar no site das IES¹ a informação sobre as habilidades e competências relacionadas ao curso de Ciência da Computação, geralmente elas são baseadas em documentos da ACM/IEEE² e SBC (Sociedade Brasileira de Computação). Porém, percebe-se que não existe uma clareza sobre esses conceitos. Além disso, essas informações estão relacionadas às competências a serem desenvolvidas no egresso dos cursos de computação. Porém, se faz necessário compreender quais são os conhecimentos, habilidades, atitudes e competências, o aluno precisa possuir para que a sua aprendizagem seja mais efetiva no ensino de programação. Desta forma, o professor poderá propor alternativas que propicie ao aluno desenvolvê-las ao longo do processo de aprendizagem, ou que ele as utilize, caso já as possuam.

3. Metodologia

Esta pesquisa tem caráter exploratório, ou seja, configura-se em uma RSL. Ela está dividida em três fases, a saber: planejamento, condução e documentação.

Na fase de planejamento foram definidas: questões de pesquisa, fontes de busca e critérios de inclusão e exclusão. Na fase de condução foi realizada a seleção dos trabalhos, para isso foram utilizadas estratégias de busca. Logo após, os dados foram extraídos, isso ocorreu durante a leitura dos artigos selecionados.

Na fase de documentação os dados foram sintetizados, e o relatório foi gerado. As subseções a seguir irão descrever de forma detalhada cada fase. A RSL é baseada no protocolo proposto por Kitchenham e Charters (2007).

3.1 Planejamento

- **Questões de pesquisa**

A RSL teve a seguinte questão central: Q. Quais são os conhecimentos, habilidades atitudes e/ou competências necessárias para que a aprendizagem de programação se torne mais efetiva? Para complementar a resposta a essa pergunta, as seguintes questões foram definidas:

QP1 - Quais as abordagens vêm sendo utilizada para o ensino de programação?

QP2 - Qual é o tipo de metodologia de ensino (tradicional ou inovadora) utilizada no processo de ensino - aprendizagem de programação?

QP3 - Quais os níveis de escolaridade as abordagens estão focadas?

QP4 - Em qual das etapas do processo de ensino - aprendizagem as abordagens estão focadas?

- **Fontes de busca**

Para a RSL levou-se em consideração a busca manual de artigos. Os anais foram selecionados em eventos nacionais (SBIE e WalgProg), tendo em vista a importância destes para a comunidade de Informática na Educação. A busca manual é importante

¹ UFJF (goo.gl/742tss), UESB (goo.gl/58YU8W), UFABC (goo.gl/kcdCfn), UFPB-Campus IV (goo.gl/skxtDK), UFERSA (goo.gl/oGNPZa), UFSC (goo.gl/SzCaku).

² goo.gl/biSBVz

porque ela pode diminuir problemas de indexação de artigos que estão localizados nas bibliotecas digitais, e podem existir outros estudos relevantes que não utilizam palavras-chave previstas em *strings* de busca. A Tabela 2 apresenta os respectivos sites das fontes de buscas utilizadas.

Tabela 2. Fontes de busca e links de acesso

Simpósio Brasileiro de Informática na Educação (SBIE)	http://www.br-ie.org/index.php/anais/bie
Workshop de Ensino em Pensamento Computacional, Algoritmos e Programação (WalgProg)	http://www.br-ie.org/pub/index.php/wcbie/issue/archive

• Critérios de Inclusão e exclusão

A inclusão dos trabalhos é determinada pela relevância em relação às questões levantadas. Os critérios de exclusão foram definidos para explicitar os trabalhos irrelevantes a essa pesquisa, eles estão detalhados na Tabela 3.

Tabela 3. Critérios de Inclusão e exclusão

Critérios de Inclusão (CI)	Critérios de exclusão (CE)
Artigos que citam conhecimentos, habilidades, atitudes ou competências desejáveis para uma aprendizagem mais efetiva de programação.	Artigos que utilizam a robótica como estratégia para o ensino de programação
Artigos que relatam iniciativas em qualquer fase do processo (aquisição, retenção ou avaliação) de ensino-aprendizagem de programação.	Artigos que investigam competências relacionadas ao pensamento computacional desenvolvidas durante a aprendizagem de programação
Artigos completos (Full Papers)	Artigos publicados antes de 2012
	Artigos sobre Revisões ou Mapeamentos Sistemáticos

3.2 Condução

• Estratégia de busca e seleção dos trabalhos

A seleção dos trabalhos foi conduzida apenas por buscas manuais, ou seja, não foi utilizado *string* de busca para a pesquisa dos artigos. A estratégia utilizada é composta por três etapas: (i) leitura do título (foi considerada a presença da palavra programação, em português e inglês (*programming*), resumo e palavras-chave (se houver); (ii) leitura da introdução e conclusão, e a exclusão dos trabalhos duplicados; (iii) leitura completa dos trabalhos e extração dos dados.

• Extração dos dados

Foram incluídos 26 Estudos Primários, os dados extraídos foram organizados em uma planilha, contendo os seguintes campos: identificador (EP = Estudo Primário), ano de publicação, meio da publicação, título e as respostas das questões de pesquisa.

3.3 Documentação

Com o objetivo de organizar e facilitar a síntese dos dados, cada artigo foi organizado em ordem cronológica crescente (por ano de publicação). Logo após, as informações foram sendo preenchidas e deram origem as respostas para as questões de pesquisa. A fase de documentação será descrita na Seção 4, tendo em vista que ela faz parte dos resultados gerados após a síntese dos dados.

4. Resultados e discussões

A primeira etapa incluiu 66 artigos, considerando os critérios de inclusão e exclusão pré-definidos e as estratégias de busca. A segunda etapa contemplou 26 artigos, ou seja, foram excluídos 40 artigos, desses, apenas 1 foi por repetição. Portanto, durante a terceira etapa da RSL foram lidos 26 trabalhos, todos eram relevantes para essa pesquisa e seus dados foram extraídos.

4.1 Respostas para as questões de pesquisa

A Tabela 4 apresenta um resumo das respostas à pergunta central desta pesquisa, por questão de espaço não foi apresentado aqui todas as respostas dos 26 estudos. A tabela completa com todas as informações coletadas pode ser vista em: goo.gl/A6VHC2.

• **Q - Quais são os conhecimentos, habilidades, atitudes e/ou competências necessárias para que a aprendizagem de programação se torne mais efetiva?**

Os conhecimentos desejáveis são relacionados a área matemática e computacional. Além disso, é preciso que o aluno domine a linguagem de programação utilizada, no aspecto de sintaxe e semântica. A língua materna é um conhecimento essencial, pois o aluno necessita compreender a contento os enunciados dos problemas propostos. O conhecimento sobre o domínio (situação problema) abordado também é de suma importância para que a aprendizagem do aluno se torne mais efetiva. Em relação as habilidades, o raciocínio lógico é bastante citado, pois ele é base para a resolução de problemas. Além disso, o aluno deve saber estruturar um programa e calcular mentalmente a saída do código a partir de uma determinada entrada, havendo uma reflexão sobre a execução do programa. Já em relação as atitudes desejadas, o aluno deve ser capaz de refletir sobre como refinar as soluções, se dedicar e manter uma continuidade nos estudos. O fato do aluno ser participativo e construir seu próprio conhecimento também é uma atitude no aluno de programação, ou seja, ele deve ser proativo. Por fim, a competência mais citada é a criatividade e a capacidade de abstração. Outras estão relacionadas com a capacidade cognitiva do aluno em compreender os enunciados, e de dominar uma forma de pensar complementemente diferente.

Tabela 4. Lista parcial dos conhecimentos, habilidades, atitudes e competências segundo os estudos

	Conhecimentos (Saber)	Habilidades (Saber como fazer)	Atitudes (Saber fazer acontecer)	Competências
SBIE	1. Sintaxe da linguagem de programação. 2. Conceitos matemáticos e computacionais. 3. Lógica de programação 4. Domínio da linguagem de programação	1. Raciocínio lógico 2. Saber estruturar o programa 3. Corrigir os possíveis erros 4. Reconhecer padrões de resolução de problemas do mundo real	1- Reflexão sobre como refinar as soluções 2- Propor soluções estruturalmente coesas 3- Motivação pessoal 4- Estudo contínuo e dedicação	1- Criatividade 2- Analisar a própria performance 3- Abstrações
WalqProg	5. Pensar algorítmicamente 6. Organização física e lógica baseadas em construtores relacionados à Matemática	5. Aplicar as estruturas de programação de maneira adequada do ponto de vista sintático e semântico. 6. Calcular mentalmente a saída do	5- Refletir sobre os exemplos estudados 6 -Ser participativo para construir seu próprio conhecimento 7- Investir tempo em atividade extraclasse	4- Capacidade de ler enunciados e entendê-los de forma clara e analítica

7. Domínio da língua materna (escrita), interpretação de textos e enunciados. 8. Características básicas da linguagem. 9. Conhecimento da situação-problema abordada.	código a partir de uma determinada entrada de forma que haja uma reflexão sobre a execução do programa apresentado. 7. Saber corrigir os erros de sintaxe 8. Saber definir casos básicos de teste	para realizarem leitura criteriosa do enunciado que receberam.	5- Realizar a ação de análise e reflexão crítica. 6- Dominar uma forma completamente nova de pensar
---	---	--	--

- QP1 – Quais as abordagens vêm sendo utilizadas para o ensino-aprendizagem de programação?

A Tabela 5 apresenta as abordagens mais utilizadas no processo de ensino-aprendizagem de programação. Em relação a aquisição do conhecimento, percebe-se que os ambientes de programação (27%) são os mais citados, o Scratch, por exemplo, tem disso bastante utilizado para auxiliar o ensino de programação introdutório.

Na fase de avaliação do conhecimento, as iniciativas estão sendo em relação a ambientes de apoio a submissão e avaliação automática de trabalhos (19%). Os estudos justificam o seu uso pela dificuldade do professor em dar feedback ao aluno, já que as turmas de programação são grandes. Os STIs (12% dos estudos) também são usados durante o processo de avaliação, pois eles conseguem propor feedback ao aluno de maneira satisfatória.

Tabela 5. Abordagens utilizadas no processo de ensino-aprendizagem citadas nos estudos

Abordagens	Qtde	%
Ambientes de programação (Alice, Portugol e Scratch)	7	27%
Ambientes de apoio a submissão e avaliação automática de trabalhos	5	19%
STI (Sistema Tutor Inteligente)	3	12%
Jogos	2	8%
Juízes Online + Moodle	1	4%
Aula tradicional + Juiz Online	1	4%
Método de Ensino de Programação Mediado por Simulação (MPMS)	1	4%
Metáfora lúdicas + Jogo	1	4%
Vídeo aula + Scratch	1	4%
Moodle	1	4%
Gamificação	1	4%
Computação Desplugada + Moodle + Facebook + WhatsApp + VisuAlg + Scratch	1	4%
Metodologia Dojo	1	4%

- QP2 – Qual é o tipo de metodologia de ensino (tradicional ou inovadora) utilizada no processo de ensino - aprendizagem de programação?

A Figura 1 mostra que 88% das abordagens utilizam metodologia inovadora, ou seja, não seguem o método tradicional de ensino (apenas aulas expositivas). Porém, 12% dos estudos ainda utilizam a combinação do método tradicional com o método inovador de ensino.

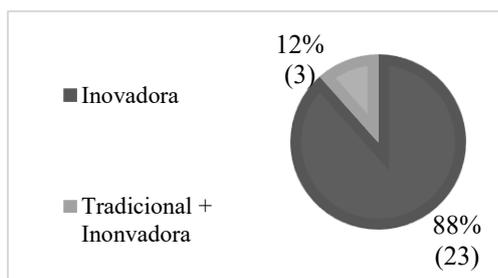


Figura 1- Metodologia de ensino utilizada nos estudos

- QP3 – Quais os níveis de escolaridade as abordagens estão focadas?

A Figura 2 apresenta a classificação dos níveis de escolaridade indicados nos estudos. Foi verificado que 62% dos artigos abordaram programação no contexto da educação superior. Apenas 11% dos artigos apresentaram iniciativas no contexto o ensino médio, 8% dos trabalhos com iniciativas no médio integrado com o técnico profissionalizante e apenas 1% no ensino fundamental. Isso evidência que se faz necessário ter iniciativas nesses níveis de escolaridade.

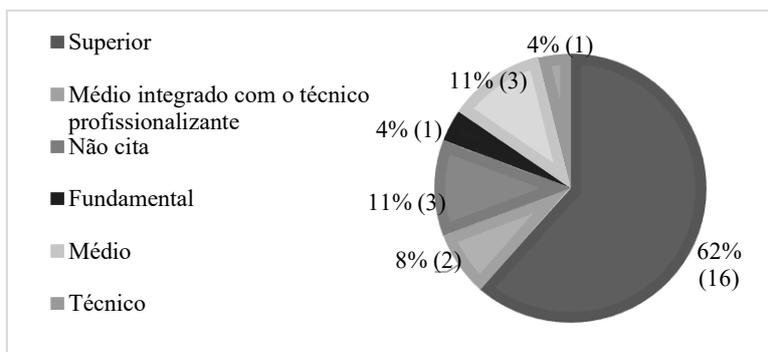


Figura 2- Níveis de escolaridade de abordados no estudo

- QP4 – Em qual das etapas do processo de ensino - aprendizagem as abordagens estão focadas?

A Figura 3 mostra que as iniciativas em relação ao processo de ensino de programação se concentram na fase de aquisição (42%) e avaliação do conhecimento (42%). Portanto, percebe-se que essas fases devem ser consideradas as mais críticas, principalmente a avaliação, pois alguns trabalhos afirmam que é difícil acompanhar os alunos, pois as turmas são grandes, essa é uma problemática bastante conhecida na comunidade de informática na educação.

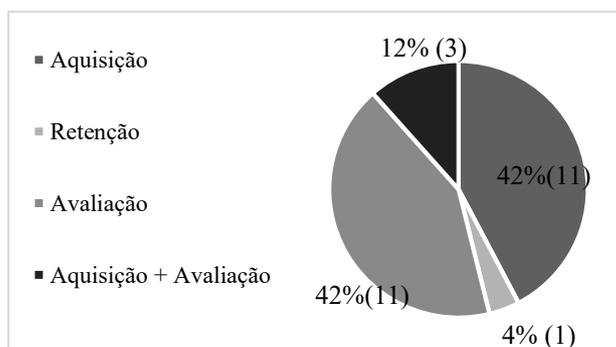


Figura 3. Iniciativas na área de programação de acordo com as fases do processo de aprendizagem

5. Conclusões e trabalhos futuros

Nesta pesquisa, apresentamos os resultados de uma RSL sobre os trabalhos relacionados ao processo de ensino-aprendizagem de programação, publicados nos últimos 4 anos no cenário nacional, no SBIE e WalgProg (duas últimas edições, 2015 e 2016). A busca realizada nesta revisão resultou na pré-seleção de 66 artigos, dentre os quais 26 foram incluídos para a extração de dados.

Os resultados gerados mostraram que existem vários Conhecimentos (C), Habilidades (H), Atitudes (A) e Competências (C) desejáveis para tornar a aprendizagem de programação mais efetiva. Alguns exemplos são: (C) Sintaxe da linguagem de programação, conceitos matemáticos e computacionais, domínio da linguagem de programação. (H) Raciocínio lógico, saber estruturar o programa, reconhecer padrões de resolução de problemas do mundo real. (A) Reflexão sobre como refinar as soluções, propor soluções estruturalmente coesas, estudo contínuo e dedicação. (CP) Criatividade, analisar a própria performance e abstração.

Sobre os tipos de abordagens utilizadas, destaca-se uma preferência pelo uso de Ambientes de Programação (Alice, Portugol e Scratch) para auxiliar o processo de ensino-aprendizagem de programação, com 27% dos estudos. Pôde-se analisar que 19% dos estudos estão relacionados com o uso de Ambientes de apoio a submissão e avaliação automática de trabalhos. A proposta de se utilizar STI para auxiliar o ensino-aprendizagem de programação também é abordada em 12% dos estudos. As demais propostas foram: Juízes Online, jogos, gamificação, metodologias (MPMS e Dojo), facebook, WhatsApp, Moodle, dentre outros.

Acerca da metodologia utilizada, 88% dos estudos são baseados em iniciativas inovadoras, ou seja, aquelas que utilizam tecnologia para auxiliar o processo de ensino-aprendizagem. Foi observado também uma maior publicação de artigos no contexto da educação superior (62%). Essa evidência mostra a necessidade de mais iniciativas que abordem o ensino de programação nos níveis fundamental, médio e técnico. Por fim, os estudos se concentram na fase de aquisição e avaliação do conhecimento, com 42% dos estudos para ambas as fases. Percebe-se que iniciativas como essa precisam ser disseminadas, principalmente porque a área de programação necessita de alternativas viáveis para auxiliar o processo de ensino-aprendizagem. Quando compreendemos as dificuldades e necessidades dos alunos é possível traçar estratégias de ensino.

Este trabalho ainda está em andamento, como trabalhos futuros pretendemos fazer uma avaliação com especialistas, no que diz respeito aos conceitos de: conhecimento, habilidades, atitudes e competências, no contexto do ensino de programação. Além disso, pretende-se se investigar se a classificação apresentada na Tabela 4 (resultado parcial), condiz com a opinião dos especialistas. Pretende-se também realizar uma classificação das habilidades, assim como abordado em Silva et al. (2015). Por fim, pretendemos aumentar o escopo desta pesquisa, incluindo novas fontes de busca e a estratégia por *strings* de busca.

Referências

- Brasil. Certificação de competências profissionais: discussões. Brasília: OIT, TEM/FAT, 1999.
- Brasil. Metodologia para o estabelecimento de perfis profissionais. Projeto Estratégico Nacional. “Certificação Profissional Baseada em Competências. Brasília: 2000.

- Brasil/cne. Parecer n. 009/2001. Institui as Diretrizes Curriculares Nacionais para a Educação Profissional de Nível Técnico.
- Burnier, S. (2002). *Pedagogia das competências: Conteúdos e métodos*. SENAC Boletim Técnico, São Paulo: SENAC.
- Chaves, E. O. C. (2004). *Educação orientada para competências e currículo centrado em problemas*. Disponível em: <http://chaves.com.br>. Acesso em: 27 de agosto de 2017.
- Chiavenato, I. (2004). *Gestão de pessoas: e o novo papel dos recursos humanos nas organizações*. Rio de Janeiro, Elsevier.
- Carroll, J.B. (1993). *Human cognitive abilities: A survey of factor-analytic studies*. New York: Cambridge University Press.
- Illeris, K. *Teorias contemporâneas da aprendizagem*. Penso Editora, 2015.
- Kitchenham, B.; Charters, S. (2007) *Guidelines for performing Systematic Literature Reviews in Software Engineering*. In: Technical report, Ver. 2.3 EBSE Technical Report. EBSE.
- Oliveira, M.; Oliveira, E. (2014). *Metodologia de Diagnóstico e Regulação de Componentes de Habilidades da Aprendizagem de Programação*. In CSBC 2014-XXII WEI (Workshop sobre Educação em Computação).
- Perrenoud, P. (1997). *Construir as competências desde a escola*. Porto Alegre: Artes Médicas.
- Pimentel, E. P.; França, V. F.; Omar, N. (2003). “A identificação de grupos de aprendizes no ensino presencial utilizando técnicas de clusterização”. In: *Workshop de Educação em Computação*, 14. Rio de Janeiro. Anais do XIV Simpósio Brasileiro de Informática na Educação. Rio de Janeiro: SBC.
- Pimentel, E. P.; França, V. F.; Noronha, R. V.; Omar, N. (2003). *Avaliação contínua da aprendizagem, das competências e habilidades em programação de computadores*. In *Anais do Workshop de Informática na Escola* (Vol. 1, No. 1, pp. 533-544).
- Primi, R.; Santos, A. A. A.; Vendramini, C. M.; Taxa, F.; Muller, F. A.; Lukjanenko, M. D. F.; Sampaio, I. S. (2001). *Competências e habilidades cognitivas: diferentes definições dos mesmos construtos*. *Psicologia: teoria e pesquisa*, 17(2), 151-159.
- Rezende, L., SEGRE, L. M., CAMPOS, G. H. B (2004). *O modelo das competências e as implicações para o currículo do curso de ciência da computação*. In *Anais do XXIV Congresso da Sociedade Brasileira de Computação (WEI)*. Salvador (Vol. 2).
- Rezende, L., Segre, L. M., Campos, G. H. B. (2007). *Relação das Competências dos Egressos de Cursos de Ciência da Computação com a Expectativa de Empresas de TI no Rio de Janeiro*.
- Rotta, I. S. (2008). *Os desafios da organização do trabalho: o novo perfil dos trabalhadores e as principais tendências no século XXI*.
- Silva, T. R.; Medeiros, T. J.; Medeiros, H.; Lopes, R.; Aranha, E. (2015). *Ensino-aprendizagem de programação: uma revisão sistemática da literatura*. *Revista Brasileira de Informática na Educação*, v. 23, n. 1.