

O ensino de engenharia de software no nível superior: um mapeamento sistemático

Maria Elaine Silva dos Santos^[1], Tamires Siqueira Rocha^[2], Mirko Barbosa Perkusich^[3]

[1]elaine.santos@academico.ifpb.edu.br. [2]tamires.siqueira@academico.ifpb.edu.br. [3]mirko.perkusich@ifpb.edu.br. Instituto Federal da Paraíba - Campus Monteiro.

RESUMO

Décadas atrás, dificilmente se imaginaria que os softwares se tornariam indispensáveis para negócios, ciências e engenharias. Assim, a Engenharia de Software (ES) tornou-se pilar dos sistemas de informação, essenciais para diversas áreas da atividade humana. Ensinar os tópicos relacionados a essa área é, entretanto, um desafio em geral, tendo em vista as suas características, composta essencialmente por conteúdos fortemente teóricos e de difícil demonstração. Por essas razões, pesquisadores em todo o mundo têm desenvolvido, ao longo dos anos, ferramentas e metodologias com o propósito de melhorar o processo ensino-aprendizagem em Engenharia de Software. O presente trabalho sumariza os resultados de uma pesquisa do tipo mapeamento sistemático da literatura que visa pesquisar, de forma abrangente, a literatura disponível acerca de um dado assunto, buscando sumarizar, relacionar e discutir os dados por eles apresentados. Assim, o presente mapeamento sistemático objetivou pesquisar, em dois conceituados canais de publicação de artigos, a literatura disponível acerca de ferramentas e metodologias utilizadas mundialmente, com a finalidade de melhorar o processo de ensino-aprendizagem em tópicos relacionados à Engenharia de Software no ensino superior. Os resultados apresentados demonstram que os EUA se destacam como o maior publicador de artigos na temática investigada; além disso, verificou-se que, nos canais pesquisados, os trabalhos centraram-se, majoritariamente, em abordagens metodológicas para melhorar o aprendizado dos estudantes.

Palavras-chave: Mapeamento Sistemático. Engenharia de Software. Ensino-aprendizagem. Metodologias. Ferramentas.

ABSTRACT

Decades ago, it was hard to imagine that softwares would become indispensable for business, science and engineering. Thus, Software Engineering (ES) has become a pillar of information systems, essential for several areas of human activity. However, teaching topics related to this area is a challenge around the world in view of its characteristics, composed essentially of dense theory and the difficulty to demonstrate content. For these reasons, researchers around the world have developed, over the years, tools and methodologies with the purpose of improving the teaching-learning process in Software Engineering. The present work summarizes the results of a systematic mapping of literature research that aims to comprehensively search the available literature on a given subject seeking to summarize, relate and discuss the data presented by them. Thus, the present systematic mapping aimed to search the available literature about tools and methodologies used worldwide in order to improve the teaching-learning process in topics related to Software Engineering in higher education in two well known channels for publishing articles. The results presented show that the USA stands out as the largest publisher of articles on the researched theme, in addition, it was found that, in the channels surveyed, the works focused mainly on methodological approaches to improve student learning.

Keywords: *Systematic Mapping. Software Engineering. Teaching-learning. Methodologies. Tools.*

1 Introdução

“Engenharia de Software (ES) é a aplicação de uma abordagem sistemática, disciplinada e quantificável ao desenvolvimento, operação e manutenção de software; isto é, a aplicação de engenharia para software” (ARDIS *et al.*, 2015, p. 11, tradução nossa). Pressman e Maxim afirmam que “[...] ela abrange um processo, um conjunto de métodos (práticas), e um leque de ferramentas que possibilitam aos profissionais desenvolverem software de altíssima qualidade” (PRESSMAN; MAXIM 2016, p. 14).

Por diversos motivos, os estudantes de disciplinas relacionadas à ES possuem dificuldades em assimilar os conhecimentos apresentados em sala de aula, principalmente, pela própria natureza dos conteúdos abordados, grande carga teórica, por serem mais abstratos e pouco demonstráveis na prática, como afirmam os autores Desai e Joshi (2012).

Os professores consideram fácil ensinar a Engenharia de Software consultando explicações sobre os vários modelos de ciclo de vida e fases do desenvolvimento de software, como Planejamento, Requisitos, Projeto, Teste, Validação e Manutenção etc. Mas a observação é que os conceitos não podem ser ilustrados facilmente no quadro-negro como pode ser aprendido através da experiência. O problema das palestras em sala de aula é que os conceitos não são muito bem ilustrados e, na maioria das vezes, fica monótono com o palestrante falando a maior parte do tempo. Falta de [sic] visualização do processo e não há experiência prática nas atividades do ciclo de vida (DESAI; JOSHI, 2012, p. 1, tradução nossa).

Esse pensamento é também compartilhado por Paiva (2012), que ressalta que as dificuldades acometem não apenas os estudantes mas também os professores. Segundo esse estudioso, “O ensino de ES exige um grau muito grande de abstração e conceitos teóricos que, muitas vezes, torna [sic] o ensino dos seus conteúdos um verdadeiro desafio para quem leciona e para quem aprende” (PAIVA, 2012, p. 26).

Por estas razões, diversos pesquisadores ao redor do mundo vêm desenvolvendo novas formas de facilitar o aprendizado dos estudantes acerca dos conteúdos essenciais para a sua formação, como justificam Lemos, Cunha e Saraiva (2019).

Diante da influência de aspectos didáticos e pedagógicos no aprendizado de ES e suas subáreas, vários pesquisadores têm proposto diferentes formas de estimular o pensamento crítico, a habilidade de solução e a aprendizagem de conceitos fundamentais (LEMOS; CUNHA; SARAIVA, 2019, p. 2).

Ademais, aspectos didáticos e pedagógicos têm deixado cada vez mais de ser uma preocupação exclusiva da academia, estendendo-se também à indústria, já que necessita de profissionais bem preparados para lidar com as demandas da área, como defende relatório técnico da Universidade da Califórnia (2014). Conseqüentemente, ao longo dos anos, essa preocupação tem se refletido no desenvolvimento de ferramentas e de diferentes metodologias diferenciadas, a fim de melhorar o ensino-aprendizagem na área de ES.

Deste modo, o interesse acadêmico por essa área de pesquisa tem propiciado um enriquecimento da literatura no que concerne a essa temática e, conseqüentemente, a obtenção de novos conhecimentos a partir dele. Assim, tendo em vista a relevância da temática exposta e da diversidade de materiais produzidos a seu respeito, vislumbrou-se um mapeamento sistemático com o propósito de sumarizar dados relevantes desses materiais por meio de um mapeamento sistemático da literatura pertinente ao tema.

De acordo com Kitchenham e Charters (2017), mapeamentos sistemáticos da literatura permitem a obtenção de novos conhecimentos, a partir de informações reconhecidas cientificamente a respeito de uma temática em específico. A partir dela, é possível traçar panoramas e plotar dados, a fim de guiar novas pesquisas no que concerne à temática pesquisada.

Assim, o presente trabalho, buscou traçar um panorama dos trabalhos publicados em dois canais de publicação na área de educação em Engenharia de Software, o *Transactions on Education*, um periódico e a *Conference on Software Engineering Education and Training*, buscando evidenciar lacunas e identificar dados recorrentes e relevantes acerca da temática em questão.

As demais seções do presente trabalho estão estruturadas do seguinte modo: a seção 2 delinea um breve panorama acerca da temática pesquisada. A seção 3 descreve a metodologia utilizada na referida pesquisa. A seção 4 apresenta os resultados encontrados e, por fim, as considerações finais são apresentadas na seção 5.

2 Referencial teórico

O mercado de Tecnologia da Informação (TI) é um dos mais valorizados atualmente. Na era da informação, os sistemas são a base de uma economia cada vez mais globalizada e dependente da tecnologia e de sistemas de informação eficientes.

Engenharia e tecnologia são domínios cruciais nas faculdades e nas indústrias, pois desempenham um papel vital na economia das nações e servem como meio para a participação ativa das nações na competitividade global, com o objetivo de atingir as metas da revolução industrial 4.0 (TALIB *et al.*, 2019, p. 69, tradução nossa).

Esse mercado, em vertiginosa ascensão, demanda mão-de-obra qualificada e preparada para trabalhar nos diferentes segmentos da área, sobretudo com utilização da ES. Ensinar os conteúdos fundamentais de uma área crítica e extremamente volátil como é a área de tecnologia e, conseqüentemente de ES, não é, entretanto, algo trivial. Na academia, os estudantes devem aprender satisfatoriamente os conteúdos lecionados, desenvolver bons hábitos e práticas. Afinal, “[...] essas habilidades serão utilizadas pelos alunos mais tarde durante seus estudos e também em sua vida profissional” (HEROUT; BRADA, 2016, p. 2, tradução nossa).

Assim, “com a complexidade do desenvolvimento de software, os profissionais são obrigados a entender os conceitos básicos de Software Engenharia” (MARQUES; SILVA, A.; SILVA, L., 2014, p. 821, tradução nossa). Além disso, as constantes mudanças na área de TI como um todo forçam as Instituições de Ensino a adequarem seus currículos às novas habilidades e conhecimentos demandados. Soma-se a isso, o problema da falta de motivação em alguns casos, causada, segundo Marques *et al.* (2014), pela própria natureza dos conteúdos de ES, que são frequentemente teóricos. Isso dificulta a aprendizagem dos alunos no atendimento das demandas de um mercado cada vez mais global.

Por essas razões, “um universo múltiplo de abordagens de ensino é adotado em torno do mundo, tanto para estudantes quanto para profissionais” (NYINKEU; NGATCHU, 2017, p. 112, tradução nossa). Essas abordagens se refletem no desenvolvimento de metodologias e ferramentas educacionais, com a finalidade de potencializar o aprendizado e a

compreensão dos conteúdos, trabalhar aspectos técnicos e humanísticos da ES, aumentar a qualidade do software, em grande parte dos casos, produzido pelos estudantes, aproximando-os das práticas que serão vivenciadas na indústria de software.

Frequentemente, os trabalhos com esta temática implementam melhorias para o ensino de uma ou mais subáreas da ES, tendo em vista sua diversidade de tópicos. Desta forma, uma abordagem bastante explorada pelos pesquisadores diz respeito ao uso de jogos com propósitos didáticos.

Os jogos educativos são uma poderosa abordagem de aprendizado para o ensino de Engenharia de Software nas quais os alunos são incentivados a participar, ajudando os educadores a simular ambientes, melhorando as habilidades dos alunos e estimulando grupos individuais e experiências sociais (MONSALVE; WERNECK; LEITE, 2011, p. 39, tradução nossa).

Melhorias educacionais do ensino-aprendizagem, por meio de ferramentas educacionais de software ou não, são propostas exploradas de maneira similar nos trabalhos de Steppe, Chin e Tuck (2017), de Pinto *et al.* (2017) e de Harout e Brada (2016). Esses estudos abordaram em seus trabalhos ferramentas de software que incentivam melhorias no aprendizado dos estudantes e ou ensino dos professores. A mesma linha de abordagem também está presente em Monsalve, Werneck e Leite (2011).

Em contrapartida, porém com o mesmo objetivo de melhoria da aprendizagem, outros estudos abordam soluções por meio de desafios e mudanças, sejam na dinâmica dos conteúdos ou no modo como estes são apresentados e exercitados, como relatado em Benabid e Al-Hudhud (2014), Georgas, Palmer e McCormick (2016), Delgado (2017), Gary, Sohoni e Lindquist (2017).

No que se refere a trabalhos correlatos, verificou-se que Santos *et al.* (2014) abordam uma pesquisa semelhante à temática explorada no presente trabalho. Observa-se como ressalvas à referida pesquisa, entretanto, o fato de não se ter parâmetros sistematicamente explícitos sobre como os citados autores procederam com o importante processo de inclusão e exclusão de artigo. Além disso, outra limitação identificada diz respeito ao reduzido número de materiais mencionados na amostra, apenas 26, algo que pode ter limitado o alcance dos resultados.

Por outro lado, em Cunha *et al.* (2018), vê-se uma abordagem temática similar ao que se verifica no presente trabalho, exceto no que se refere a aspectos metodológicos e aos objetivos da pesquisa, que, no caso citado, objetivou mapear a literatura disponível apenas no âmbito do Brasil. No trabalho mencionado, os autores extraíram resultados interessantes, por meio da realização de *snowballing*, algo que possibilitou aos autores traçarem um interessante panorama da literatura em educação em ES, no Brasil, no intervalo temporal de duas décadas.

Assim, “[...] pesquisadores e educadores têm dedicado esforços para desenvolver e aplicar novos métodos de ensino para apoiar o extenso currículo de engenharia de software” (SOUZA *et al.*, 2018, p. 2, tradução nossa). Esses esforços têm tornado rica a literatura na área de educação em ES. Por estas razões, no presente estudo, buscou-se analisar os artigos publicados em dois conhecidos canais focados na área de educação em ES, buscando extrair e sumarizar os dados das publicações encontradas traçando um recorte temporal.

3 Método da pesquisa

A presente pesquisa aborda um mapeamento sistemático da literatura a respeito das ferramentas (de software ou não) e das metodologias desenvolvidas e ou aplicadas mundialmente, com a finalidade de incrementar, maximizar ou aperfeiçoar o aprendizado e ou ensino de tópicos relacionados à Engenharia de Software.

A opção pela adoção desta modalidade de pesquisa se deu porque, de acordo com Kitchenham e Charters (2007), pesquisas desse tipo são mais apropriadas quando, “[...] durante o exame inicial de um domínio, antes do comissionamento de um sistema revisão, descobre-se que existe muito pouca evidência ou que o tópico é muito amplo” (KITCHENHAM; CHARTERS, 2007, p. 5, tradução nossa).

Ademais, julga-se pertinente explicitar a diferença existente entre estudos do tipo mapeamento e do tipo revisão sistemática. De acordo com Dermeval, Coelho e Bittencourt (2019), a diferença entre os dois tipos de estudos secundários reside no nível de aprofundamento das questões de pesquisa. Assim, revisões da literatura possuem um delineamento maior dessas questões enquanto os mapeamentos sistemáticos possuem questões um pouco mais abstraídas e que exigem menos aprofundamento na extração de dados, como corroboram os autores Kitchenham e Charters (2007).

Um estudo sistemático de mapeamento permite que as evidências em um domínio sejam plotadas em alto nível de granularidade. Isso permite a identificação de grupos de evidências e desertos de evidências para direcionar o foco de futuras revisões sistemáticas e identificar áreas para estudos mais primários a serem realizados (KITCHENHAM; CHARTERS, 2007, p. 5, tradução nossa).

3.1 Questões de pesquisa

As questões de pesquisa são os principais guias de revisões e mapeamentos sistemáticas da literatura. São elas que norteiam as atividades subsequentes da pesquisa, e suas definições são de extrema importância para o sucesso de sua execução, como afirmam Dermeval, Coelho e Bittencourt (2019).

Assim como em outros métodos de pesquisa, a questão de pesquisa de uma revisão ou mapeamento sistemático da literatura é norteadora de toda a condução da pesquisa. Todas as atividades posteriores da revisão derivam da questão de pesquisa, logo, defini-la de forma fidedigna ao tópico que se quer investigar é fundamental para o sucesso da execução de uma revisão da literatura. (DERMEVAL, COELHO, BITTENCOURT, 2019, p. 7).

De forma semelhante, definir boas questões de pesquisa constitui-se como uma fase de extrema importância de qualquer pesquisa secundária, conforme esclarecem estudiosos, como Kitchenham e Charters (2007). Isto posto, a presente pesquisa buscou responder aos questionamentos descritos no Quadro 1:

Quadro 1 – Questões de Pesquisa.

ID	Descrição
1	Quais os anos em que houve mais publicações de artigos com a temática proposta nos dois canais durante o período especificado?
2	Qual a origem geográfica da maioria dos estudos?
3	Quais as universidades e os pesquisadores que mais publicaram?
4	Quais as soluções que foram utilizadas para mitigar as dificuldades no ensino e na aprendizagem de Engenharia de Software?

Fonte: Elaboração própria.

3.2 Estratégia de Busca

A estratégia de busca adotada nesta pesquisa foi uma busca sistemática manual, em dois canais de publicação de artigos relacionados à educação em Engenharia de Software. Esta busca restringiu-se também aos anos de 2010 e de 2019 bem como aos canais *Transactions on Education*, um periódico e a *Conference on Software Engineering Education and Training*, um evento que também publica trabalhos relacionados à educação e treinamento em ES.

Dada a restrição da busca em dois canais específicos de publicação, não se fez necessária a formulação e utilização de *Strings* de busca ou palavras-chave na busca dos materiais de uso deste trabalho. Neste caso, os materiais da pesquisa foram extraídos na forma de metadados (título, nomes dos autores, data, nome do canal de publicação, palavras-chave e tipo de documento), em formato *BibTeX*, diretamente das páginas eletrônicas dos canais selecionados.

Posteriormente, esses dados foram convertidos, utilizando-se o sistema auxiliar de condução de revisões sistemáticas da literatura, o *Parsifal*, resultando numa planilha do tipo Excel contendo todos os dados dos estudos. Assim, de posse de todos os títulos, cada um dos documentos teve seu conteúdo avaliado, a fim de auxiliar o processo de classificação e, posteriormente, de extração de dados.

3.3 Critérios de exclusão

Trabalhos de revisão da literatura necessitam de critérios de exclusão consistentes dos artigos coletados durante o processo de busca. Na presente pesquisa, os autores estabeleceram alguns critérios de exclusão de materiais que não fossem de interesse da pesquisa, a saber: (i) materiais em desacordo com o que abrange a ISO 12207:2017, (ii) artigos duplicados, (iii) resumos estendidos, erratas ou workshops que eventualmente fossem recuperados junto aos materiais coletados em ambos os canais.

3.4 Procedimentos de seleção de estudos

Determinou-se, ainda, que os artigos passariam por um processo de leitura adaptativa. Conforme Ali, Petersen e Wohlin (2014), a leitura adaptativa é um processo no qual cada um dos artigos é avaliado por dois revisores de forma independente. Inicialmente, neste caso, cada artigo deveria ser avaliado pelo título e ou *abstract*. Ao final, o material era classificado

como RELEVANTE, IRRELEVANTE ou INCERTO para a pesquisa, por cada revisor. Deste modo, cada artigo passou por duas classificações, e a decisão de incluí-lo ou não dependia da avaliação das regras presente no Quadro 2:

Quadro 2 – Critérios de classificação de artigos

		Revisor 2		
		Relevante	Incerto	Irrelevante
Revisor 1	Relevante	A	B	D
	Incerto	B	C	E
	Irrelevante	D	E	F

Fonte: Adaptado de Ali, Petersen e Wohlin (2014).

A classificação seguia, ainda, algumas regras pré-estabelecidas para o processo de seleção: (1) Artigos classificados com A e B seriam incluídos como potenciais estudos primários; (2) Artigos classificados com F seriam automaticamente excluídos; (3) Artigos classificados como C passariam por uma segunda avaliação, utilizando-se o processo de leitura adaptativa; (4) Artigos classificados como D e E mostram um desacordo (D é pior). Os revisores deveriam discutir e reclassificá-los para A, C ou F, de modo que, ao final, todos estivessem classificados e que fosse possível aferir o nível de sintonia dos revisores quanto aos objetivos da pesquisa.

4 Resultados da pesquisa

A busca nas bases de dados resultou em um escopo de 791 materiais. Após o processo de exclusão de materiais não relevantes, de acordo com os critérios de exclusão metodologicamente definidos, resultou-se no quantitativo de 744 artigos como amostra da presente pesquisa. A descrição dos resultados da aplicação do processo de exclusão é apresentada no Quadro 3.

Quadro 3 – Classificação dos materiais após aplicação dos critérios de exclusão

Etapa: análise em bases pré-definidas		
Fonte	Retornados	Avaliados
IEEE Transactions	481	478
CSEET	310	266

Fonte: Elaboração própria.

Nota-se que, com a aplicação dos critérios de exclusão, 47 documentos foram removidos do conjunto de artigos a serem avaliados nas etapas seguintes. Desta forma, subsequentemente, iniciou-se a seleção dos artigos a serem inclusos, a partir da classificação de dois revisores, de forma independente. Com a conclusão desta etapa, aferiu-se o percentual de concordância entre os revisores, por meio do coeficiente de Fleiss, como preconizam Gwet (2010) e Randolph (2005).

A escolha da versão do Fleiss se deu por não haver restrição para a quantidade de casos que precisam ser categorizados para cada categoria (Irrelevante, Relevante, Incerto), conforme defendem Brennan e Prediger (1981). Assim sendo, após verificação, constatou-se que o percentual de concordância foi de 72,88%. Sabendo-se que o coeficiente foi calculado

baseado apenas na avaliação de elementos como o Título e o Resumo e que o processo de avaliação adaptativa prevê a discussão entre os juízes no caso de discordâncias, considera-se como satisfatório o grau de concordância obtido.

Finalmente, após os processos de exclusão e classificação dos estudos, a quantidade de artigos resultante foi de 131, sendo 18 do *Transactions on Education*, um periódico e 113 da *Conference on Software Engineering Education and Training*. Deste conjunto, extraíram-se as informações de interesse da pesquisa. O primeiro dado extraído refere-se aos países que mais publicaram pesquisas no âmbito do ensino-aprendizagem em ES nos canais pesquisados e no período de tempo delimitado. Neste sentido, é possível visualizar que a temática sofreu variações constantes no decorrer dos anos como demonstra a Figura 1.

Figura 1 – Quantidade de artigos que abordam a melhoria do ensino-aprendizagem em ES

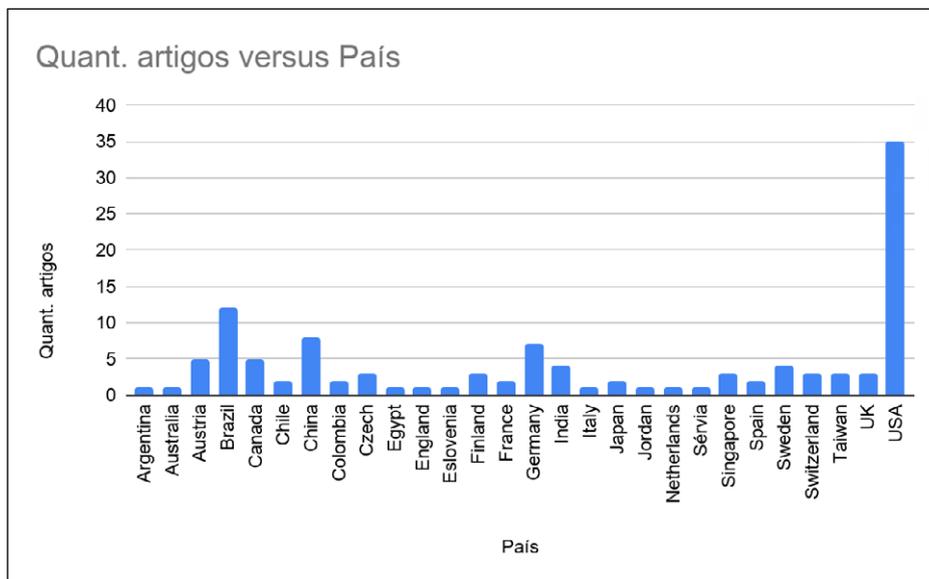


Fonte: Elaboração própria.

Ao analisar a Figura 1, nota-se que a quantidade de publicações oscilou consideravelmente, visto que, em determinados anos, observou-se alta no desenvolvimento de estudos envolvendo a melhoria do ensino-aprendizagem em ES, enquanto, em outros, verificaram-se quedas consideráveis no número de publicações. Complementarmente, considerando as duas bases, a média anual do tipo pesquisado girou em torno de 14 de publicações.

Outra informação extraída diz respeito aos países de origem dessas publicações. Neste sentido, buscou-se visualizar os países e continentes de onde mais surgiram pesquisas com a referida temática. Neste quesito, verificou-se que os Estados Unidos da América aparecem como o maior publicador de pesquisas que abordaram a melhoria do ensino-aprendizagem em Engenharia de Software, como demonstra a Figura 2.

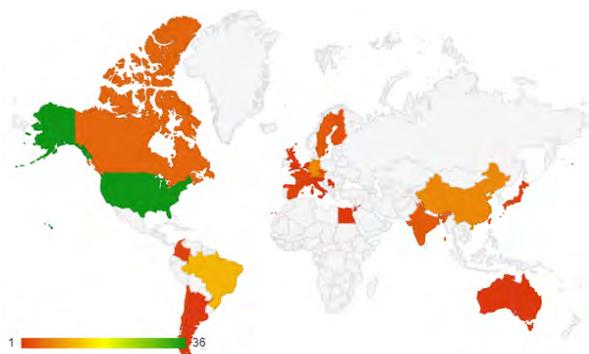
Figura 2 – Quantidade de artigos que abordam a temática da melhoria do ensino-aprendizagem em Engenharia de Software



Fonte: Elaboração própria.

Ainda no que diz respeito aos países que mais publicaram, é possível destacar outros três países que se sobressaíram na produção desses artigos, em especial, o Brasil com 12, a China com 8 e a Alemanha com 7 como detalha a Figura 3. Ainda nesse aspecto, buscou-se avaliar qual a distribuição geográfica das publicações encontradas. Os dados referentes a esta questão são detalhadas na Figura 3.

Figura 3 – Distribuição dos países por região geográfica dos artigos dos canais pesquisados



Fonte: Elaboração própria.

Posto isso, de acordo com a figura 3, fica nítido que países da África, da América Central, Oriente Médio, Norte da Ásia, Sul e Sudoeste Asiático não tiveram publicações na área de ensino em ES nos canais pesquisados. O baixo número de publicações nesses

canais pode ser justificado por diversos problemas tais como: questões geo e socioeconômicas, baixo interesse na referida linha de pesquisa ou, ainda, a própria natureza do presente estudo, pela baixa aleatoriedade da amostra produzida e busca restrita a apenas dois canais de publicação.

Sobre a classificação dos trabalhos quanto ao conteúdo abordado, identificou-se que 26 deles abordaram o desenvolvimento e ou aplicação de ferramentas (computacionais ou não), para melhoria do ensino. Por outro lado, a expressa maioria dos trabalhos abordou o desenvolvimento e ou implementação de metodologias ou abordagens para melhorar a aprendizagem dos estudantes. O Quadro 4 detalha os dados relativos a este quesito, segmentado por canal de publicação.

Quadro 4 – Tipo de abordagem por canal de publicação

Tipo de conteúdo abordado	Canal	
	Periódico	Conferência
Metodologias/abordagens	15	90
Ferramentas	3	23

Fonte: Elaboração própria.

De acordo com o Quadro 4, é possível verificar uma maior riqueza de materiais com a temática pesquisada na *Conference* que no periódico. Além disso, visualiza-se que parece haver um maior interesse dos pesquisadores por trabalhos que apoiam com mais ênfase, o desenvolvimento/implementação de abordagens para melhoria do ensino-aprendizagem que o desenvolvimento/implementação propriamente dita de ferramentas, sejam elas computacionais ou não.

Outra questão de interesse da pesquisa diz respeito às universidades e aos autores que mais publicaram artigos no período e nos periódicos pesquisados. Neste sentido, verificou-se que a instituição de ensino mais vinculada aos artigos dos canais pesquisados foi a *Carnegie Mellon University*, uma instituição conhecida internacionalmente por sua excelência no ensino de Engenharia de Software, referida 7 vezes aos trabalhos pesquisados.

Entre os trabalhos pesquisados, verificou-se que o nome mais frequente nas publicações foi o do chinês Guoping Rong, que apareceu como autor principal de 5 artigos, seguido do austríaco Andreas Bollin, que apareceu como autor principal em outros 4 artigos. Ainda neste sentido, destaca-se o pesquisador chinês Dong Shao com 5 artigos, considerando-se os artigos em que aparece como autor e coautor.

5 Considerações finais

Frequentemente, os trabalhos com esta temática implementam melhorias para subáreas da ES, como nas áreas de design de software, modelos de processos de desenvolvimento, requisitos de software, entre outros. No que se refere às propostas de melhoria do processo ensino-aprendizagem, pode-se dizer, porém, que existem duas vertentes principais: trabalhos voltados ao desenvolvimento e ou aplicação de ferramentas e trabalhos voltados ao desenvolvimento e ou aplicação de metodologias.

Apesar da variação observada na quantidade de publicações nos canais estudados, verifica-se que a educação em Engenharia de Software é uma temática de constante interesse dos pesquisadores. Constata-se ainda que existe uma distribuição consideravelmente desigual das publicações ao redor do mundo, fato que exige estudos complementares para avaliar quais fatores colaboram para números tão baixos.

Na perspectiva das abordagens utilizadas, nota-se que existiu um foco maior dos autores por melhorias metodológicas, desde a forma como

os conteúdos devem ser lecionados, até como disponibilizar *feedback* aos estudantes na tarefa de auxiliá-los em seu aprendizado. Uma possível lacuna do presente estudo, diz respeito à identificação, classificação e, eventualmente, avaliação dessas metodologias e ferramentas de acordo com requisitos pré-estabelecidos, algo que pode ser justificado por inviabilidade no cronograma e da quantidade de documentos a serem avaliados.

REFERÊNCIAS

- ALI, N. B.; PETERSEN, K.; WOHLIN, C. A systematic literature review on the industrial use of software process simulation. **Journal of Systems and Software**. v. 97, p. 65-85, 2014.
- ARDIS, M. *et al.*, **SE 2014**: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering in Computer, v. 48, n. 11, p. 106-109, 2015. DOI: 10.1109/MC.2015.345. Disponível em: <https://www.acm.org/binaries/content/assets/education/se2014.pdf>. Acesso em: 22 jan. 2020.
- BENABID, A.; AL-HUDHUD, G. The synergy of engineering and research aspects in Software Engineering Education. *In*: Conference on Software Engineering Education and Training (CSE&T). n. 27, 2014, Klagenfurt, Austria. **Proceedings[...]**. IEEE. DOI: 10.1109/CSE&T.2014.6816780. Disponível em: <https://ieeexplore.ieee.org/document/6816780>. Acesso em: 21 jan. 2020.
- BRENNAN, R. L.; PREDIGER, D. J. Coefficient Kappa: Some uses, misuses and alternatives. **Educational and Psychological Measurement**. n. 41, p. 687-699, 1981.
- CUNHA, J. A. O. *et al.* Software engineering education in Brazil: a mapping study. *In*: Brazilian Symposium on Software Engineering. n. 32, p. 348 – 356, 2018. **Proceedings[...]**. IEEE. DOI: 10.1145/3266237.3266259. Disponível em: <https://www.researchgate.net/publication/327798050>. Acesso em: 24 fev. 2020.
- DELGADO, D. *et al.* Evolving a Project-Based Software Engineering Course: A Case Study. *In*: Conference on Software Engineering Education and Training (CSE&T). n. 30, 2017, Savannah, GA, USA. **Proceedings[...]**. IEEE. DOI: 10.1109/CSE&T.2017.22. Disponível em: <https://ieeexplore.ieee.org/document/8166686>. Acesso em: 21 jan. 2020.
- DERMEVAL, D.; COELHO, J. A. P. M.; BITTENCOURT, I. I. Mapeamento Sistemático e Revisão Sistemática da Literatura em Informática

na Educação. *In*: JAQUES, P. A.; PIMENTEL, M.; SIQUEIRA, S.; BITTENCOURT, I. (Org.)

Metodologia de Pesquisa em Informática na Educação: Abordagem Quantitativa de Pesquisa.

Porto Alegre: SBC, 2019. (Série Metodologia de Pesquisa em Informática na Educação, v. 2) Disponível em: <https://metodologia.ceie-br.org/livro-2>. Acesso em: 22 fev. 2020.

DESAI, P.; JOSHI, G. H. Activity Based Teaching Learning in software Engineering. *In*: IEEE International Conference on Engineering Education: Innovative Practices and Future Trends (AICERA). p. 1-6. 2012. **Proceedings [...]** AICERA. DOI: 10.1109/AICERA.2012.6306729. Disponível em: <https://ieeexplore.ieee.org/document/6306729>. Acesso em: 22 jan. 2020.

GARY, K.; SOHONI, S.; LINDQUIST, T. It's Not What You Think: Lessons Learned Developing an Online Software Engineering Program. *In*: 30th Conference on Software Engineering Education and Training (CSEE&T), n. 30, 2017, Savannah, GA, USA. **Proceedings [...]**. IEEE. DOI: 10.1109/CSEET.2017.45. Disponível em: <https://asu.pure.elsevier.com/en/publications/its-not-what-you-think-lessons-learned-developing-an-online-softw>. Acesso em: 23 jan. 2020.

GEORGAS, J.; PALMER, J. D.; McCORMICK, M. J. Supporting software architecture learning using runtime visualization. *In*: 29th Conference on Software Engineering Education and Training (CSEE&T), n. 29, 2016, Dallas, USA. **Proceedings [...]**. IEEE. Disponível em: <https://nau.pure.elsevier.com/en/publications/supporting-software-architecture-learning-using-runtime-visualiza>. Acesso em: 23 fev. 2020.

GWET, K. L. (2010). **Handbook of interrater reliability** (2nd ed.). Gaithersburg, MD: Advanced Analytics.

HEROUT, P.; BRADA, P. UML-Test Application for Automated Validation of Students' UML Class Diagram. *In*: Conference on Software Engineering Education and Training (CSEE&T). n. 29, 2016, Dallas, TX, USA. **Proceedings [...]**. IEEE. DOI: 10.1109/CSEET.2016.33. Disponível em: <https://ieeexplore.ieee.org/document/7474487>. Acesso em: 21 jan. 2019.

ISO / IEC / IEEE 12207: 2017 (E) First edition 2017-11: ISO / IEC / IEEE International Standard - Systems and software engineering - Software life cycle processes.

KITCHENHAM, B. A., CHARTERS, S. **Guidelines for performing Systematic Literature Reviews in Software Engineering**. 2007. Disponível em: <https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.117.471>. Acesso em: 23 jan. 2020.

LEMOS, Wellington; CUNHA, José; SARAIVA, Juliana. Ensino de Engenharia de Software em um Curso de Sistemas de Informação: Uma Análise dos Problemas e Soluções na Perspectiva de Professores e Alunos. *In*: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI), 27., 2019, Belém. **Anais [...]** do XXVII Workshop sobre Educação em Computação. Porto Alegre: Sociedade Brasileira de Computação, July 2019. p. 305-318. ISSN 2595-6175. DOI: <https://doi.org/10.5753/wei.2019.6638>. Acesso em:

MARQUES, D.; DA SILVA, A. C.; DA SILVA, L. F. A Survey to Evaluate Educational Games Designed to Teach Software Engineering. European Conference on Games Based Learning. n. 2, p. 821-823, 2014. Disponível em: <https://search.proquest.com/openview/3b643d8c64d4c2d0f663ded1e374a9f0/>. Acesso em: 21 jan. 2020.

MEIRELES, M.; BONIFÁCIO, B. Uso de Métodos Ágeis e Aprendizagem Baseada em Problema no Ensino de Engenharia de Software: Um Relato de Experiência. *In*: Simpósio Brasileiro de Informática na Educação. **Anais [...]** do XXVI Simpósio Brasileiro de Informática na Educação. p. 180-189, 2015, Maceió. DOI: <http://dx.doi.org/10.5753/cbie.sbie.2018.1798>. Disponível em: <https://br-ie.org/pub/index.php/sbie/article/view/8155>. Acesso em: 22 jan. 2020.

MONSALVE, E. S.; WERNECK, V.; LEITE, J. C. S. P. Teaching software engineering with SimulES-W. *In*: Conference on Software Engineering Education and Training (CSEE&T). n. 24, 2011, Honolulu, HI, USA. **Proceedings [...]**. IEEE. DOI: 10.1109/CSEET.2011.5876102. Disponível em: <https://www.researchgate.net/publication/260739789>. Acesso em: 21 jan. 2020.

NYINKEU, N. D ; NGATCHU, H. Work and Play in Software Engineering Training: Experiences from the Silicon Mountain. . *In*: 30th Conference on Software Engineering Education and Training (CSEE&T). n. 30, 2017, Savannah, GA, USA. **Proceedings [...]**. IEEE. DOI: 10.1109/CSEET.2017.26. Disponível em: <https://ieeexplore.ieee.org/document/8166690>. Acesso em: 21 jan. 2020.

PAIVA, S. R. ESCollab: uma metodologia colaborativa voltada para o ensino de engenharia de software. 2012. Dissertação (Mestrado em Informática) - Centro de Informática da Universidade Federal da Paraíba, UFPB. João Pessoa, 2012. Disponível em: <https://repositorio.ufpb.br/jspui/handle/tede/6073>. Acesso em: 24 fev. 2020.

PINTO, G. H. L. *et al.* Training Software Engineers Using Open-Source Software: The Professors' Perspective.

In: Conference on Software Engineering Education and Training (CSEE&T). n. 30, 2017, Savannah, GA, USA. **Proceedings [...]**. IEEE. DOI: 10.1109/CSEET.2017.27. Disponível em: <https://ieeexplore.ieee.org/document/8166691>. Acesso em: 21 jan. 2020.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**: uma abordagem profissional. 8. ed. Porto Alegre: AMGH, 2016.

RANDOLPH, J. J. **Free-marginal multirater kappa**: an alternative to Fleiss' fixed-marginal multirater kappa. Joensuu University Learning and Instruction Symposium. 2005.

SHABALINA, O. ; SADOVNIKOVA, N.; KRAVETS, A. Methodology of Teaching Software Engineering: Game-Based Learning Cycle. *In*: 3rd Eastern European Regional Conference on the Engineering of Computer Based Systems. n. 3, 2013, Budapest, Hungria. **Proceedings [...]** do 3rd Eastern European Regional Conference on the Engineering of Computer Based Systems. DOI: 10.1109/ECBS-EERC.2013.22. Disponível em: <https://ieeexplore.ieee.org/abstract/document/6664517>. Acesso em: 22 jan. 2020.

SANTOS, R. *et al.* Ferramentas, Métodos e Experiências no Ensino de Engenharia de Software: um Mapeamento Sistemático. **Anais [...]** do III Congresso Brasileiro de Informática Na Educação. 2014. p. 544-548.

SIEVI-KORTE, O.; SYSTÄ, K.; HJELSVOLD, R. Global vs. local – Experiences from a distributed software project course using agile methodologies. *In*: IEEE Frontiers in Education Conference (FIE). 2014, El Paso, TX, USA. **Proceedings [...]** do FIE. DOI: 10.1109/FIE.2015.7344101. Disponível em: <https://ieeexplore.ieee.org/document/7344101>. Acesso em: 22 jan. 2020.

STEPPE, K. CHIN, S.; TUCK, W. W. Improving the Teaching of Software Design with Automated Modelling of Syntactic Dependencies. *In*: 30th Conference on Software Engineering Education and Training (CSEE&T), n. 30, 2017, Savannah, GA, USA. **Proceedings [...]**. IEEE. DOI: 10.1109/CSEET.2017.31. Disponível em: https://www.researchgate.net/publication/321657461_Improving_the_Teaching_of_Software_Design_with_Automated_Modelling_of_Syntactic_Dependencies. Acesso em: 21 jan. 2020.

TALIB, C. A. *et al.* Enhancing Students' Reasoning Skills in Engineering and Technology through Game-based learning. **International Journal of Emerging Technologies in Learning (IJET)**. v 14, n. 24, p. 69-80, 2019. Disponível em: <https://onlinejour>.

journals.publicknowledgeproject.org/index.php/i-jet/article/view/12117. Acesso em: 23 jan. 2020.

UNIVERSIDADE DA CALIFÓRNIA. **Software Engineering Curriculum Technology Transfer**: Lessons learned from MOOCs and SPOCs. Berkeley, UC, 2014. Disponível em: <http://www2.eecs.berkeley.edu/Pubs/TechRpts/2014/EECS-2014-17.html>. Acesso em: 22 jan. 2020.

SOUZA, M. R. A. *et al.* A systematic mapping study on game-related methods for software engineering education. **Information and Software Technology**, n. 95, p. 201–218.