

Teaching Software Engineering: An Overview of Current Approaches and Practices in the Last Decade of SBES

Ana Clementino
Universidade Federal Rural do
Semi-Árido (UFERSA)
ana.clementino@alunos.ufersa.edu.br

Erick Lima
Universidade Federal do Cariri
(UFCA)
erick.brito@aluno.ufca.edu.br

Luann Lima
Universidade Federal do Cariri
(UFCA)
luann.alves@aluno.ufca.edu.br

André Guedes
Instituto Federal de Mato Grosso
(IFMT)
andre.calisto@ifmt.edu.br

Dorgival Netto
Universidade Federal do Cariri
(UFCA)
dorgival.netto@ufca.edu.br

Jarbele Coutinho
Universidade Federal Rural do
Semi-Árido (UFERSA)
jarbele.coutinho@ufersa.edu.br

ABSTRACT

Software Engineering is constantly evolving, with new tools and programming languages fluctuating in popularity over time. However, the curriculum of Software Engineering disciplines cannot keep up with these innovations and remain current. On the other hand, if the courses promote constant adaptation of the teaching of the subject to keep it in line with updates, this may result in a misaligned curriculum. Therefore, it is necessary to have an in-depth understanding of successful practices, the challenges faced, and the opportunities that may arise in this process. Based on this understanding, this paper presents a bibliographic study of the Brazilian Symposium on Software Engineering (SBES) over the last ten years (2013 to 2023) with the main objective of investigating the approaches currently used to promote teaching areas that include Software Engineering. In this way, 48 papers were identified that aim to answer the questions investigated. Finally, the intention is, based on the best practices measured, to develop a strategic model that can serve as a guide for identifying existing approaches.

KEYWORDS

Software Engineering Education, Teaching, Bibliographic Study

1 INTRODUÇÃO

Com as rápidas mudanças no cenário tecnológico e a necessidade de adaptação dos profissionais do setor de Tecnologia da Informação (TI), o ensino de conceitos de Computação, e mais especificamente, de Engenharia de Software (ES) nos cursos de graduação na área de TI tem se tornado um desafio.

Acompanhar o avanço das tecnologias e dos processos adotados na indústria de software, e alinhá-los com os conteúdos programáticos não é uma tarefa trivial [23]. Em contrapartida, além da adoção de práticas comuns como aulas expositivas e leituras complementares na área de ES, percebe-se um esforço da academia em contemplar também a competência prática no desenvolvimento de pequenos projetos, para estimular nos estudantes habilidades técnicas e não-técnicas inerentes ao desenvolvimento de software[23].

Diante deste cenário, alguns estudos têm se dedicado a evidenciar os principais tópicos de discussão no ensino de ES no Brasil ([19], [12], [4], [21], [36]). Entretanto, apesar dos esforços para caracterizar experiências e abordagens inovadoras no processo de ensino e aprendizagem de Engenharia de Software, existe uma carência

de estudos que revelem um panorama sintetizado de abordagens e recursos disponíveis no contexto do ensino de ES nas Instituições de Ensino Superior (IES) brasileiras.

Neste contexto, este artigo investiga a seguinte questão de pesquisa: “Qual o panorama atual de abordagens adotadas para o ensino e aprendizagem de conceitos relacionados à área de Engenharia de Software no contexto do Simpósio Brasileiro de Engenharia de Software (SBES)?”. Assim, o objetivo é examinar as abordagens de ensino e aprendizagem de conceitos de ES e suas subáreas, por meio de artigos publicados nas diversas trilhas do SBES no período compreendido entre os anos de 2013 à 2023, de modo a contribuir com informações relevantes e atuais para a formação acadêmica e profissional de estudantes e profissionais de ES.

A fim de responder a questão de pesquisa, foi conduzido um estudo bibliográfico por meio da biblioteca digital mantida pela Sociedade Brasileira de Computação (SBC), SBC-OpenLib (SOL), destinada aos anais do *Simpósio Brasileiro de Engenharia de Software*. Este estudo identificou um conjunto de 48 artigos, abrangendo subáreas da ESe apresentando diversas abordagens como metodologias ativas, gamificação, atividades práticas, sala de aula invertida, entre outros. Esses artigos discutem a implementação de atividades lúdicas e dinâmicas, além do uso de jogos para facilitar a compreensão do estudantes que têm dificuldade em assimilar conteúdos avançados. Também destaca-se a utilização de projetos de software em sala de aula para proporcionar uma experiência prática, equilibrando conceitos teóricos e práticos.

Este trabalho está organizado da seguinte forma: A Seção 2 explora os conceitos relacionados ao processo de ensino e aprendizagem de Engenharia de Software e os trabalhos relacionados. Na Seção 3 consta a metodologia de pesquisa utilizada. Na Seção 4 são apresentados os resultados gerais obtidos na pesquisa. Na Seção 5 são feitas discussões acerca das análises realizadas. Na Seção 6 são discutidas algumas principais ameaças a validade deste estudo. Na Seção 7 são relatadas as considerações finais e propostas para trabalhos futuros.

2 ENSINO DE ENGENHARIA DE SOFTWARE

O ensino de Engenharia de Software nas Instituições Federais do Brasil enfrenta desafios específicos relacionados à estrutura curricular, à disponibilidade de recursos e à formação dos docentes [47]. A literatura especializada aponta para uma lacuna significativa entre o ensino tradicional, muitas vezes focado em conteúdos

denso e abstrato, e as demandas da indústria por profissionais com habilidades práticas e experiência em projetos reais [36][17].

Assim, se faz importante manter um currículo atualizado e alinhado com as demandas do mercado de trabalho [47]. Este currículo deve contemplar tanto os aspectos teóricos quanto os práticos da Engenharia de Software. No entanto, Santiago et al. [36] e Dos Santos et al. [17] corroboram de que o ensino, frequentemente atrelado a um modelo teórico, não prepara adequadamente os alunos para os desafios na carreira profissional de Engenharia de Software.

Nesse contexto, por exemplo, a Aprendizagem Baseada em Projetos (ABP) surge como uma solução promissora, defendida por Santiago et al. [36], permitindo que os alunos vivenciem o desenvolvimento de software de forma similar ao contexto profissional e minimizem a lacuna entre a academia e a indústria. Através de projetos práticos, os estudantes desenvolvem habilidades como trabalho em equipe, comunicação, planejamento e resolução de problemas, além de aprofundarem seus conhecimentos em áreas como requisitos de software, modelagem, prototipação e testes [16]. Essa interdisciplinaridade enriquece o aprendizado, preparando os alunos para atuarem em equipes multidisciplinares, comuns no mercado de trabalho.

Além disso, a formação dos docentes em Engenharia de Software também é um fator crítico para a qualidade do ensino [47]. É fundamental que os professores possuam sólida formação acadêmica e experiência prática na área, além de estarem atualizados com as novas tecnologias e tendências do mercado.

2.1 Estratégias para o Ensino de Engenharia de Software

Habitualmente, a abordagem tradicional de aulas expositivas e exercícios repetitivos pode resultar em desmotivação e dificuldade de aprendizagem. Para superar essa barreira, as metodologias ativas e a gamificação surgem como ferramentas importantes para promover um ensino mais engajador e eficaz.

Neste contexto, as metodologias ativas, buscam promover a participação ativa e a construção do conhecimento através de atividades práticas e interativas, de modo que, ao adotá-las, o ensino de ES se torna mais atraente e eficiente [25, 28]. Se destacam, por exemplo: a Aprendizagem Baseada em Problemas (ABP), no qual consiste em apresentar um problema real ou simulado aos alunos e desafiá-los a encontrar soluções, mobilizando seus conhecimentos prévios e buscando novos aprendizados [25]; e, a Aprendizagem Colaborativa que envolve a participação ativa dos alunos em atividades em grupo, como discussões, debates, resolução de problemas em conjunto e projetos colaborativos [13, 42].

Outra estratégia que se destaca como uma abordagem interessante para tornar o ensino de Engenharia de Software mais envolvente e motivador, promovendo a aprendizagem ativa e o desenvolvimento de habilidades essenciais para a formação de profissionais qualificados é a gamificação. No entanto, é fundamental analisar criticamente as potencialidades e os cuidados necessários para uma implementação eficaz e ética da gamificação, considerando tanto os benefícios quanto os potenciais efeitos negativos. Ferreira et al. [19] e Da Cunha et al. [12] apontam para a necessidade de inovar nas metodologias de ensino para atender às demandas do mercado de

trabalho e superar os desafios enfrentados pelas instituições de ensino superior no Brasil. A gamificação surge como uma alternativa viável, capaz de promover o engajamento dos alunos, a colaboração e o desenvolvimento de habilidades como pensamento crítico, resolução de problemas e trabalho em equipe. Mas, por outro lado, também se faz necessário compreender os efeitos negativos em se adotar estratégias de gamificação no ensino de ES, destaca Almeida et al. [1]. Portanto, ao se adotar a gamificação é fundamental ter um planejamento estratégico.

2.2 Trabalhos Relacionados

Alguns estudos dedicaram-se à investigação sobre o ensino de ES através de estudos secundários [1, 25, 37]. Santos et al. realizou um mapeamento sistemático da literatura para identificar jogos sérios desenvolvidos no Brasil para o ensino de ES. O objetivo principal foi analisar o estado da arte dos jogos educacionais, focando em suas características, plataformas, tópicos de ES abordados e métodos de avaliação. A pesquisa identificou 25 jogos educacionais, revelando uma predominância de jogos digitais para *desktop* e uma maior concentração em tópicos como gerenciamento de projetos, processo de software e teste de software. Os autores identificaram a necessidade de jogos que abranjam novos tópicos e dinâmicas, além das tradicionais, para fortalecer o ensino de ES no Brasil.

Já Lima et al. [25] apresenta uma revisão sistemática da literatura sobre a aplicação de metodologias ativas no ensino de ES. O objetivo é analisar como essas metodologias contribuem para o ensino e aprendizagem da área, identificando os métodos mais utilizados, as dificuldades e os benefícios relatados. A pesquisa analisou 34 estudos, revelando que a aprendizagem baseada em problemas e a sala de aula invertida são os métodos mais utilizados, enquanto a gamificação aparece como uma metodologia promissora. Os resultados também evidenciaram dificuldades como a falta de conhecimento sobre metodologias ativas, a complexidade em implementar a teoria na prática e a necessidade de maior tempo para atividades.

Enquanto que Almeida et al. [1] se aprofunda na investigação dos efeitos negativos da gamificação em softwares educacionais. O objetivo é mapear os elementos de design de jogos que podem causar efeitos indesejáveis e analisar a percepção de desenvolvedores sobre esses efeitos. A pesquisa identificou 87 trabalhos que relatam efeitos negativos, destacando elementos como *badges*, *leaderboards*, competições e pontos como os que mais geram problemas. Os efeitos negativos mais comuns são a falta de impacto, o desempenho prejudicado, questões motivacionais, falta de compreensão e irrelevância. O estudo também destaca a importância de considerar o contexto e as necessidades dos usuários ao utilizar elementos de gamificação em softwares educacionais.

3 METODOLOGIA ADOTADA

De modo a atender o objetivo desta pesquisa, foi definido um método de pesquisa baseado em um estudo bibliográfico realizado nos anais do Simpósio Brasileiro de Engenharia de Software (SBES) publicados nos anos de 2013 à 2023. Para garantir o rigor metodológico da pesquisa, foram adotadas algumas das boas práticas recomendadas por Kitchenham e Charters [22], para protocolos de estudos bibliográficos mais específicos, como Revisão ou Mapeamento Sistemático.

3.1 Planejamento

Objetivo Geral de Pesquisa e Questões de Pesquisa. O objetivo geral desta pesquisa consistiu em compreender as principais abordagens adotadas no ensino de Engenharia de Software e suas subáreas, no contexto dos anais do SBES. Desse modo, foram definidas as seguintes Questões Específicas (QE) de pesquisa:

- **QE1:** Quais abordagens têm sido adotadas no ensino de conteúdos inerentes à ES? Esta questão buscou identificar uma lista de estratégias metodológicas utilizadas para promover o ensino de ES ou de suas subáreas.
- **QE2:** Como as abordagens encontradas para o ensino de conteúdos de ES podem ser categorizadas? Nesta questão de pesquisa, o objetivo foi fornecer uma categorização das estratégias encontradas, a partir das similaridades entre áreas de estudo.
- **QE3:** Como as abordagens encontradas foram validadas para promover o ensino de ES? Esta questão de pesquisa objetivou perceber aspectos de validação da abordagem no contexto de ensino.
- **QE4:** Quais as possibilidades de inovação no ensino de ES? Esta questão de pesquisa investigou as oportunidades de estudo indicadas nos artigos, de modo a perceber estratégias inovadoras neste contexto.

Processo de Busca. A busca de artigos para este estudo bibliográfico foi realizada na biblioteca digital mantida pela SBC, SBC-OpenLib (SOL), destinada aos anais do *Simpósio Brasileiro de Engenharia de Software* - disponível através do endereço *web*: <https://sol.sbc.org.br/index.php/sbes/issue/archive>. Em seguida, foi definida a *string* de busca para selecionar os artigos relevantes ao contexto de ensino de ES. Foram definidos termos na língua portuguesa e na língua inglesa, para abranger a coleta de mais estudos nesta pesquisa. Portanto, baseado no contexto da pesquisa foi definida a seguinte *string* de busca:

((“educação” OR “Educação”) AND (“ensino” OR “aprendizagem” OR “Ensino” OR “Aprendizagem”)) OR ((“education” OR “Education”) AND (“teaching” OR “learning” OR “Teaching” OR “Learning”)) AND ((“engenharia de software” OR “Engenharia de Software”) OR (“software engineering” OR “Software Engineering”))

Para abranger as subáreas da ES determinadas no *Software Engineering Body of Knowledge* (SWEBOK) [5] também foi especificada e adotada a seguinte *string* de busca mais específica. Neste caso, a *string* “<subárea>” foi substituída por *string(s)* correspondente(s) ao(s) termo(s) e sinônimo(s) de cada subárea no SWEBOK:

((“educação” OR “Educação”) AND (“ensino” OR “aprendizagem” OR “Ensino” OR “Aprendizagem”)) OR ((“education” OR “Education”) AND (“teaching” OR “learning” OR “Teaching” OR “Learning”)) AND <subárea>

(i) (“requisito” OR “requirement” OR “Requisito” OR “Requirement”); (ii) (“análise” OR “projeto” OR “modelagem” OR “Análise” OR “Projeto” OR “Modelagem”) OR (“analysis” OR “project” OR “modeling” OR “Analysis” OR “Project” OR “Modeling”); (iii) (“teste”

OR “test” OR “Teste” OR “Test”); (iv) (“manutenção” OR “manutenção” OR “Manutenção” OR “Maintenance”); (v) (“gerência de configuração” OR “gerenciamento de configuração” OR “Gerência de Configuração” OR “Gerenciamento de Configuração”) OR (“configuration management” OR “configuration management” OR “Configuration Management” OR “Configuration Management”); (vi) (“gerência de projeto” OR “gerenciamento de projetos” OR “Gerência de Projeto” OR “Gerenciamento de Projetos”) OR (“project management” OR “project management” OR “Project Management” OR “Project Management”); (vii) (“processo de desenvolvimento” OR “Processo de Desenvolvimento”) OR (“development process” OR “Development Process”); (viii) (“ferramenta” OR “método” OR “Ferramenta” OR “Método”) OR (“tool” OR “method” OR “Tool” OR “Method”); (ix) (“qualidade” OR “quality” OR “Qualidade” OR “Quality”).

Seleção e Avaliação dos artigos. Para selecionar os artigos encontrados, foram definidos Critérios de Inclusão (CI) e Critérios de Exclusão (CE), conforme recomendado por Kitchenham e Charters [22]. Assim, para um artigo ser incluído neste estudo, todos os critérios de inclusão definidos devem ter sido satisfeitos. Caso contrário, o artigo deve ser desconsiderado deste estudo. Portanto, foram considerados os seguintes CI:

- CI1: Estudos que tratem sobre Ensino de Engenharia de Software.
- CI2: Ter relevância com as *strings* de buscas.
- CI3: Estudos primários.
- CI4: Estudos publicados nos anais do SBES de 2013 à 2023.

Enquanto que, os CE definidos foram os seguintes:

- CE1: Estudos que não se referem à Engenharia de Software ou subáreas da ES, de acordo com SWEBOK, no contexto de educação.
- CE2: Não ter relevância com as *strings* de buscas.
- CE3: Estudos secundários ou terciários.
- CE4: Estudos publicados em um idioma diferente do inglês ou do português.
- CE5: Estudos sem acesso disponível;
- CE6: Estudos incompletos (menos de quatro páginas);

3.2 Execução

O período de seleção e avaliação dos artigos foi compreendido entre 05 de fevereiro e 10 de maio de 2024. Para selecionar os artigos na biblioteca digital SBC-OpenLib, foi implementado e adotado um *script* automatizado - através da linguagem de programação *Python* 7. Após a busca automatizada, o processo de seleção e avaliação dos artigos foi conduzido em duas etapas. Na primeira etapa foram analisados o título, o resumo e as palavras-chaves de cada artigo retornado, de acordo com todos os CE. Os artigos incluídos nessa etapa constituíram uma nova lista de artigos incluídos (LI1). Na segunda etapa foi analisado todo o texto dos artigos da LI1 a fim de

construir uma lista final de trabalhos selecionados, aplicando agora os CI. Para isso, foi feita a leitura completa dos artigos da LI1 para identificar os estudos que deveriam compor a lista final de artigos incluídos (LI).

3.3 Extração e síntese de dados

Com o objetivo de compreender o contexto das pesquisas publicadas sobre o ensino de Engenharia de Software no SBES, foram extraídos alguns principais dados gerais do artigo, como: título; ano de publicação; autores; instituições de afiliação dos autores; unidades federativas dos autores, conforme instituição; subárea da ES conforme o SWEBOK [5]; e, acesso a um material suplementar ou artefato da abordagem. Além disso, para responder às questões de pesquisa específicas, apresentadas na Seção 3.1, foram extraídos trechos dos artigos. Por fim, para apoiar a análise e documentação dos artigos foram utilizadas as ferramentas: *Google Docs*¹, *Google Sheets*² e *Google Drive*³. É importante observar que pelo menos dois pesquisadores analisaram e classificaram cada artigo. De modo que, na existência de desacordos ou dúvidas na coleta da informação, um terceiro pesquisador interveio na análise, analisando as informações e contribuindo com decisões para chegar em um consenso.

4 RESULTADOS

A partir dos dados extraídos dos trabalhos primários, foram sintetizadas algumas informações relevantes sobre o ensino de Engenharia de Software, no contexto do SBES dos últimos 10 anos. Dessa forma, a Seção 4.1 discute as informações resultantes dos dados coletados dos artigos, conforme apresentado na Seção 3.3. Enquanto que, a Seção 4.2 discute os dados, de modo a fornecer respostas às questões de pesquisa, definidas na Seção 3.1.

4.1 Informações resultantes dos dados extraídos dos artigos

Inicialmente, a busca automatizada nos anais do SBES dos anos de 2013 à 2023 resultou em 65 trabalhos primários retornados. Assim, estes trabalhos foram analisados conforme a primeira etapa de seleção (definida na Seção 3.2), o que resultou em 52 trabalhos incluídos para a segunda etapa, e 13 trabalhos excluídos. Através da leitura completa dos textos desses trabalhos, na segunda etapa do estudo, foram incluídos 48 trabalhos primários, e excluídos 04 trabalhos. Por fim, compuseram a Lista final de trabalhos primários Incluídos (LI), um total de 48 artigos. Esta LI pode ser consultada por meio do Material Suplementar.

Em relação à distribuição de artigos publicados ao longo dos anos foi percebido que em 2020, o SBES teve seu maior percentual de trabalhos publicados no contexto de ensino de ES, com 11 artigos publicados. Em seguida, os anos de 2017 e 2018, também se destacaram como edições do evento que promoveram discussões sobre esta temática, por meio de 10 artigos publicados, em cada um destes anos. Percebe-se também que no ano de 2023 foram publicados 8 artigos no contexto de ensino de ES; no ano de 2021 houveram 6 artigos; no ano de 2022, apenas 2 artigos; e, no ano de 2014, apenas 1 artigo. Outros aspecto percebido se refere à ausência de artigos

publicados neste contexto, nas edições do evento nos anos de 2019, 2016, 2015 e 2013.

Também foi feito um levantamento das instituições de afiliações de cada autor dos artigos. Foi constatada a participação de 41 instituições distintas e de 169 pesquisadores envolvidos na autoria dos artigos. Dentre as instituições que se destacam com o envolvimento de pesquisadores e ações de pesquisa acerca do ensino de ES, foram identificadas a UFC (32 pesquisadores), a PUCRS (17 pesquisadores), a PUC-Rio e a Unipampa (11 pesquisadores, cada uma delas), a UFAM e a UFMA (9 pesquisadores, cada), a USP (8 pesquisadores), a UFBA (6 pesquisadores), a UFPE, UEM e UnB (5 pesquisadores, cada), a UFPA e UTFPR (4 pesquisadores, cada), e, a UFF e a UP (3 pesquisadores, cada). As demais instituições identificadas tiveram uma representação de 2 pesquisadores ou menos.

Outro aspecto percebido foi que 31 pesquisadores (equivalente a 20%) tiveram autoria em mais de um artigo. Dentre eles, 0,65% tiveram autoria em pelo menos 4 artigos; 6,45% em pelo menos 3 artigos; e, 12,90% em pelo menos 2 artigos. Os demais 80% dos pesquisadores publicaram apenas 1 artigo neste recorte temporal do SBES. A lista com a identificação destes pesquisadores, bem como o detalhamento do quantitativo de pesquisadores relacionados a suas instituições de afiliação são apresentados no Material Suplementar. É importante ressaltar que para esta análise foi necessário realizar uma limpeza nos dados referentes aos nomes dos pesquisadores, uma vez que alguns deles usaram grafias diferentes em seus nomes.

De acordo com cada instituição de afiliação dos pesquisadores foram extraídas suas regiões de origem. Portanto, foram identificados 19 estados brasileiros: o Ceará (CE), com 34 pesquisadores; seguido do estado Rio Grande do Sul (RS), com 31 pesquisadores, e do Rio de Janeiro (RJ), com 16 pesquisadores; e, do Paraná (PR) e do Amazonas (AM), com 13 pesquisadores. Os demais estados identificados apresentaram menos 10 de pesquisadores e podem ser visualizados no Material Suplementar.

Em relação à relevância dos artigos para a comunidade científica, foi realizada uma análise de *clusters* e percebeu-se que o artigo de Ferreira [19] fundamenta e direciona pesquisas subsequentes, pois há uma evidencia de centralização em torno deste artigo, como pode ser verificado nos artefatos do Material Suplementar. O estudo de Ferreira [19], conecta-se diretamente a outros estudos importantes, como Silva [38] e Costa [8]; e, à uma rede dispersa de estudos subsequentes, indicando diferentes níveis de influência e desenvolvimento de temas de pesquisa. Além disso, as conexões indiretas, como as observadas em Lima [24] e Olivindo [30], destacam a extensão do impacto de Ferreira [19] na comunidade científica.

É possível identificar uma ligação entre Lima [24] e Trinta [43], sugerindo uma continuidade ou colaboração significativa entre esses estudos que abordam metodologias ativas. Outros temas são abordados nos *clusters* como *Design Thinking* ([41], [27], [46]), ensino de Teste de Software ([45], [3]) e estudos sobre a ótica dos professores de Engenharia de Requisitos ([2], [35]).

Esta dispersão sugere uma diversidade de tópicos e subtemas dentro do campo de estudo, refletindo uma ampla gama de interesses e abordagens. Os nós isolados representam pesquisas emergentes que ainda não foram amplamente integradas na literatura principal, oferecendo oportunidades para identificar novas direções e tendências na pesquisa, como ensino de Teste de Software ([29], [10], [9], [33]), Manutenção de Software ([4]), Arquitetura de Software

¹ *Google Docs*: <https://docs.google.com/document/u/0/?tgif=c>

² *Google Sheets*: <https://docs.google.com/spreadsheets/u/0/?tgif=c>

³ *Google Drive*: <https://drive.google.com/>

([40]), UML ([39], [18]), Métodos Ágeis ([6], [20]), Gamificação ([7] [15]), Metodologias Ativas ([11], [34], [31]), utilização atividades lúdicas ([14]), estratégias pedagógicas inovadoras ([21]), simulações e exemplos práticos ([26], [32]).

Os artigos da LI também foram categorizados conforme as subáreas da Engenharia de Software, indicadas no SWEBOK [5]. A Tabela 1 destaca as subáreas de ES encontradas em cada artigo analisado. Desse modo, percebe-se que no contexto de ensino de ES, a subárea de Análise/Projeto/Modelagem foi abordada em 31% dos artigos; a subárea de Processo de Desenvolvimento foi investigada por 25% artigos; a subárea de Teste de Software foi explorada em 15% artigos; a subárea de Requisitos de Software foi estudada em 8% artigos; a subárea de Manutenção de Software foi examinada em 4% dos artigos; e, por fim, a subárea de Gerência de Projetos foi abordada em 2% dos artigos. É importante destacar que as subáreas de Qualidade de Software, Gerência de Configuração e Ferramentas/Métodos não apresentaram artigos relacionados. Outros 15% dos artigos apresentaram pesquisa abrangendo a área geral de Engenharia de Software ([A19,A21,A25,A26,A28,A36,A39]).

Em relação à disponibilidade dos artefatos usados nas abordagens apresentadas nos artigos ou de um material suplementar que apoie a replicação dessas abordagens, foi constatado que apenas 42% dos artigos fornecem informações sobre os artefatos usados nos estudos, através de endereços eletrônicos de repositórios de dados. Enquanto que 58% dos artigos não dispõem o acesso aos artefatos ou fornecem informações de acesso indisponíveis. As informações de acesso aos materiais suplementares das abordagens disponibilizadas pelos artigos da LI deste estudo são fornecidas no Material Suplementar.

4.2 Respostas às Questões de Pesquisa

4.2.1 QE1: Quais abordagens têm sido adotadas no ensino de conteúdos inerentes à ES? Dentre as abordagens, utilizadas no ensino de ES no contexto das publicações do SBES, na última década, destacam-se: jogos educacionais, *hackathons*, metodologias ativas, gamificação, *design thinking*, estudos de caso e projetos práticos. Diversos jogos educacionais têm sido incorporados ao ensino de ES a fim de tornar o processo de ensino e aprendizagem mais interativo e prático. Pode-se citar o artigo (A1) que apresenta a adaptação do jogo Lego4Scrum para ser utilizado um quadro branco *online* (utilizando o Miro⁴) com o auxílio da ferramenta de comunicação Discord⁵ para o ensino de Métodos Ágeis. Outros jogos, como o jogo de tabuleiro Scrum (A29), o Testing Game (A30) e o GreaTest (A45), também foram utilizados para ensinar Métodos Ágeis e Teste de Software, respectivamente. Além desses, jogos para o ensino de Arquitetura de Software (A17) e a integração de jogos educativos com conceitos de proveniência para análise de fluxo de jogo e tomada de decisões (A28) foram apresentados.

Hackathons educacionais (A9 e A14), projetos FLOSS (*Free/Libre and Open Source Software*) (A7 e A10) e projetos OSS (*Open Source Software*) (A24), foram implementados para engajar os alunos e desenvolver habilidades de trabalho em equipe e comunicação. Estas iniciativas proporcionaram um ambiente de aprendizagem dinâmico, além de estimular a criatividade e a colaboração entre os participantes.

⁴Miro:<https://miro.com/>

⁵Discord:<https://discord.com/>

Metodologias ativas, como a Aprendizagem Baseada em Problemas (PBL) têm sido amplamente utilizadas para aplicação prática de conceitos teóricos, por exemplo, no ensino de Teste de Software (A34) e sistemas móveis colaborativos são incentivados a aplicar esses conceitos na prática, desenvolvendo projetos de aplicativos móveis colaborativos na plataforma *Android* (A38). A aplicação de metodologias como PBL e Métodos Ágeis, integração entre disciplinas específicas, definição de processos e contextos para condução da fábrica de software (A39).

Estudos mostram o uso de 15 técnicas de *Design Thinking* (DT) para apoiar a eliciação de requisitos (A20) e estimular as *soft skills* (A22). Integração da utilização entre PBL e mapas mentais (A31). Outras práticas incluem a integração entre disciplinas, metodologias ativas como sala de aula invertida (*Flipped Classroom*), e uso de ferramentas para simulação de projetos reais (A32). Paschoal et al. [33] realizou um experimento comparando os requisitos de teste derivados por alunos que estudaram com o modelo de sala de aula invertida, com os requisitos de teste derivados por alunos que estudaram com o modelo de ensino tradicional (A12).

Abordagens como a Aprendizagem Baseada em Desafios (*CBL - Challenge Based Learning*) (A16) e o Aprendizagem Baseada em Equipes (*TBL - Team-Based Learning*) integrado ao *framework* IDEAL (A44) também têm tido resultados significativos no maior engajamento dos alunos e percepções positivas tanto dos estudantes quanto dos professores. Há a junção de metodologias ativas como em Coutinho et al. [9] que utilizam PBL e o Ensino *Just-in-Time* (JiTT) para o ensino de Teste de Software em equipes ágeis de Desenvolvimento de Software Distribuído, especialmente enfocando o Teste Exploratório (A8). Metodologias ativas, como aprendizagem baseada em problemas e sala de aula invertida, também foram utilizadas durante o Ensino Remoto Emergencial (ERE) (A13). Devido essa diversidade de abordagens, Lima et al. [24] propõem um guia que apoia o docente na escolha do tipo de metodologia ativa a ser introduzida no ensino de ES, a partir da compatibilidade do perfil, interesse e do estilo de aprendizagem dos discentes (A6).

A aplicação de elementos de jogos, como pontuações e recompensas, foi utilizada para aumentar a motivação e o engajamento dos alunos. Ferramentas como *Classcraft* permitiram a criação de atividades gamificadas, mesmo em um ambiente assíncrono (A18). Dinâmicas práticas, trabalho em grupo gamificados (A47) também foram outras estratégias implementadas. Alguns artigos apresentaram as percepções dos estudantes adquiridas durante as experiências de aulas invertidas com roteiros de estudo gamificados (A4). A gamificação também foi utilizada durante o ERE (A5).

Estudos de caso reais e simulações de ambientes de trabalho foram utilizados para ajudar os alunos a entenderem a aplicação prática dos conceitos de Engenharia de Software. Combinação de aulas teóricas e projetos práticos em ES para proporcionar um aprendizado mais uniforme (A36). Abordagens como a simulação da interação cliente-desenvolvedor (A21), envolvimento dos alunos em projetos de código aberto (A24 e A41), e a participação em equipe envolvidas em projetos reais com o uso de metodologias ágeis e outras práticas pedagógicas para promover a aprendizagem baseada em projetos (A37). A utilização de exemplos práticos para demonstrar o ensino de modelagem de software (A23).

Os alunos também participaram de projetos de software com metodologias ágeis e *design* centrado no usuário, culminando em

Subárea de Engenharia de Software	ID dos artigos analisados
Análise/Projeto/Modelagem	[A4, A5, A10, A13, A15, A17, A23, A31, A32, A37, A38, A43, A46, A47, A48]
Processo de Desenvolvimento	[A1, A2, A3, A6, A9, A7, A14, A16, A18, A29, A35, A41]
Teste de Software	[A8, A11, A12, A22, A30, A34, A45]
Requisitos de Software	[A20, A33, A40, A44]
Manutenção de Software	[A27, A42]
Gerência de Projetos	[A24]

Tabela 1: Categorização dos artigos por área de conhecimento da Engenharia de Software, conforme SWEBOK [5].

competições de empreendedorismo (A43). Abordagens como *Think-Pair-Square* (A46) também foram implementadas para comparar os efeitos do aprendizado em grupo e individualmente. De forma diferente, foram identificados artigos que exploraram o uso atual de exemplos de projetos reais empregados por docentes no ensino de ES (A3).

Ferramentas específicas também têm sido adotadas para finalidades como treinamento de testadores de localização e internacionalização com falhas realistas em aplicativos Android (A27); e, apoio ao ensino de Orientação a Objetos, a partir de dados gerados por alunos para identificar dificuldades (A48).

A aplicação de *Design Thinking* no ensino de ES visa promover experiências práticas e apoiar o desenvolvimento de *soft skills* (A15). A organização do curso em *Sprints*, criação de Mínimos Produtos Viáveis (MVPs), uso de materiais *premium* e integração com a indústria por meio de vídeos *React* e reuniões síncronas com profissionais de referência são exemplos dessas práticas (A25).

Além dessas abordagens, alguns artigos trazem a perspectiva dos professores em relação ao processo de ensino e aprendizagem, integrando diferentes metodologias para potencializar o aprendizado dos estudantes em Engenharia de Requisitos (A26, A33 e A42). Outros trabalhos fornecem uma visão geral do ensino de Teste de Software a partir da perspectiva dos educadores (A11). A Arquitetura Pedagógica adotada integra aprendizagem baseada em projetos, discussões em classe e dinâmicas de aprendizagem em grupo (A26). Outros trabalhos buscam promover um programa de ensino para o Processo de Software, com base no uso de abordagens centradas no aluno (A2).

Costa et al. [8] destaca a integração de estudos de caso, considerando perspectivas de alunos, instituições de ensino superior e empresas de TI no Maranhão, e propõe reflexões sobre a educação alinhada às demandas locais (A19). Por fim, Trinta e Santos [44] discutem a integração entre disciplinas e uso de ferramentas no ensino de conceitos fundamentais sobre computação em nuvem, virtualização, analisando *feedbacks* dos alunos sobre os conteúdos apresentados (A35).

4.2.2 QE2: Como as abordagens encontradas para o ensino de conteúdos de ES podem ser categorizadas? Conforme análise, as abordagens mais usadas têm sido: metodologias de ensino para diferentes finalidades, jogos, ferramentas específicas, gamificação, e *Design Thinking*. Apenas em três artigos não foi possível identificar as abordagens adotadas (A11, A19, A33). Para apresentar esses resultados foi utilizado uma abreviatura (de uso específico deste estudo) para as subáreas da ES, conforme descrito a seguir: Análise/Projeto/Modelagem (APM); Processo de Desenvolvimento (PD);

Teste de Software (TS); Requisitos de Software (RS); Manutenção de Software (MS); Gerência de Projetos (GP); e, para identificar a área geral de Engenharia de Software, adotamos sigla ES.

Desse modo, a Tabela 2 categoriza os artigos que adotam abordagens relacionadas às metodologias de ensino adotadas nas pesquisas, relacionando-as às áreas da ES, conforme SWEBOK [5]. Em seguida, na Tabela 3 são categorizados os artigos que adotam abordagens relacionadas à gamificação e jogos, conforme subáreas da ES. E, por fim, na Tabela 4 são categorizados os artigos que adotam abordagens relacionadas ao uso de ferramentas, conforme subáreas da ES.

Portanto, percebe-se o uso frequente de metodologias ativas e a aplicação prática dos conhecimentos de ES, visando envolver os alunos de maneira mais prática e colaborativa. Isso sugere uma movimentação para além do ensino tradicional, baseado em palestras, para abordagens que incentivam a participação ativa dos estudantes. Além disso, percebe-se a integração de ferramentas tecnológicas no processo de ensino. Destacam-se o uso de softwares como *Miro*, *Trello*, *Discord*, *Google Classroom*, *Padlet*, *GitHub*, dentre outros, considerados fundamentais para proporcionar uma experiência de aprendizado mais dinâmica e realista.

Um outro aspecto interessante é a prevalência de jogos e gamificação, que reflete um esforço em tornar o aprendizado mais envolvente e motivador para os alunos, utilizando elementos de ludificação para facilitar a compreensão de conceitos complexos. A gamificação emergiu como uma técnica recorrente (A4, A6, A34, A47), assim como a sala de aula invertida (A4, A6, A12, A13, A32).

Outras metodologias específicas incluem abordagens centradas no aluno (A2) e a aprendizagem baseada em exemplos (A3, A23). Estudos de caso e exemplos práticos são frequentemente utilizados para ilustrar conceitos complexos (A9, A14, A17, A18, A35, A36, A39, A40, A46). Bem como, o uso do *Design Thinking* (A15, A20). Outras categorias incluem ensino baseado em equipes cruzadas (A21), atividades práticas e interação com a comunidade de código aberto (A24, A41), *sprints* e MVPs com integração à indústria (A25), e a integração entre disciplinas e metodologias ágeis (A37, A42, A43).

4.2.3 QE3: Como as abordagens encontradas foram validadas para promover o ensino de ES? Com o objetivo de perceber aspectos de validação das abordagens utilizadas em cada estudo, os artigos foram categorizados conforme o tipo de pesquisa adotada para validação da abordagem. Desse modo, foi identificado que o tipo de pesquisa mais adotada para validação das abordagens foram *Surveys*, percebidos em 18 artigos (A3, A4, A5, A9, A7, A10, A11, A15, A16, A23, A31, A32, A34, A35, A36, A39, A40, A42). Outro

Metodologias de Ensino	APM	PD	TS	RS	MS	GP	ES
Metodologias ativas (PBL, CBL, TBL, PjBL)	[A13, A17, A31, A32, A38, A47]	[A1, A6, A16, A18]	[A8, A22]	[A40, A44]	-	-	[A26, A39]
Abordagem centrada no aluno	-	-	-	-	[A2]	-	-
Aprendizagem Baseada em Exemplos	[A23]	[A3]	-	-	-	-	-
Ensino baseado em equipes cruzadas	-	-	-	-	-	-	[A21]
Atividades práticas, interação com a comunidade de código aberto	-	[A41]	-	-	-	[A24]	-
<i>Sprints</i> , MVPs, integração com a indústria	-	-	-	-	-	-	[A25]
Integração entre disciplinas, metodologias ágeis	[A37, A43]	-	-	-	[A42]	-	-
Sala de Aula Invertida	[A4, A13, A32]	[A6]	[A12]	-	-	-	-

Tabela 2: Artigos que adotaram abordagens relacionadas à metodologias de ensino, de acordo com as subareas da ES.

Gamificação/Jogos	APM	PD	TS	RS	MS	GP	ES
Jogos	[A5, A17, A31]	[A1, A18, A29]	[A30, A34, A45]	[A40]	-	-	[A28, A36]
Gamificação	[A4, A47]	[A6]	[A34]	-	-	-	-

Tabela 3: Artigos que adotaram abordagens relacionadas à gamificação e jogos, de acordo com as subareas da ES.

Abordagens Gerais	APM	PD	TS	RS	MS	GP	ES
Uso de Ferramentas	[A10, A17, A32, A38, A48]	[A1, A7, A18]	-	[A20, A40]	[A27]	-	-
Exemplos Práticos	[A17, A46]	[A9, A14, A18, A35]	-	[A40]	-	-	[A36, A39]
<i>Design Thinking</i>	[A15]	-	-	[A20]	-	-	-

Tabela 4: Artigos que adotaram abordagens gerais, de acordo com as subareas da ES.

tipo de pesquisa comum entre os estudos foi o **Relato de Experiência**, apresentado em 10 artigos (A8, A13, A18, A21, A30, A37, A43, A44, A45, A47) e a execução de **Experimentos**, descritos em 7 artigos (A12, A20, A27, A28, A29, A46, A48). Seguidos do relato de **Estudos de Caso** (A17, A22, A24, A26), **Entrevistas** (A1, A14, A41) e **Estudos bibliográficos** (A19, A33). Outros artigos destacaram tipos de pesquisa, como pesquisa-ação, grupo focal, painel de especialistas, e estudo piloto (A2, A6, A25, A38). Em sua maioria, os estudos analisados dedicaram-se a praticar a validação das abordagens com profissionais da área de ES, educadores de ES e/ou estudantes de cursos de graduação na área de Computação. A Tabela 5 apresenta a categorização por tipo de pesquisa realizada em cada artigo, para validação das abordagens propostas, indicando os respectivos artigos.

4.2.4 QE4: Quais as possibilidades de inovação no ensino de ES? As possibilidades de inovação no ensino de ES também foram investigadas e listadas a partir das propostas de trabalhos futuros indicados em cada artigo analisado. De modo geral, a Tabela 6 apresenta uma síntese das propostas inovadoras indicadas para a continuidade de estudos e aplicações experimentais relacionadas ao ensino de ES.

Os artigos sobre jogos (A1, A5) destacam o estímulo ao engajamento entre os estudantes e adoção de práticas de aprendizagem ativa. Outros artigos (A28, A29, A30) indicam o uso de colaboração

online e *Design Centrado no Usuário* (DCU), sugerindo investigações sobre o uso de jogos para diferentes conteúdos de ES, avaliação da aprendizagem dos estudantes e novos estudos promovendo as comparações com grupos que não utilizam gamificação. É importante destacar que alguns artigos indicam o uso de jogos para o ensino de conceitos específicos de ES (A31, A34, A36), como *Design de Interface de usuário*, *Métodos Formais* e *Abordagens Baseadas em Componentes*; bem como apontam para a expansão do uso de *Jogos Educativos online*.

As Metodologias Ativas de Ensino (PBL, CBL, TBL, PjBL) são indicadas em diferentes artigos. Estes, também sugerem a exploração e expansão de *Jogos Educativos* para ensinar conceitos de ES, integrar tecnologias digitais para ampliar a interatividade, aplicar gamificação em diferentes contextos educacionais e avaliar a eficácia dessas abordagens na aprendizagem dos estudantes. Outros artigos (A6, A8, A16, A26, A31 e A44) enfatizam a importância de metodologias ativas e colaborativas no ensino de ES. Estas metodologias incluem o uso de ciclos de *design* para aprimorar abordagens educativas, a implementação de estratégias colaborativas em contextos de desenvolvimento ágil, e a aplicação de PBL e TBL.

Os artigos A7, A20 e A27 enfatizam a aplicação de metodologias ativas no ensino de ES. Por exemplo, A7 sugere o uso de PBL para ensinar conceitos complexos de ES através de cenários reais e

Tipos de validação	ID dos artigos
Entrevistas	[A1, A14, A41]
Estudo bibliográfico/teórico	[A19, A33]
Estudo de Caso	[A17, A22, A24, A26]
Experimento	[A12, A20, A27, A28, A29, A46, A48]
Relato de Experiência	[A8, A13, A18, A21, A30, A37, A43, A44, A45, A47]
Survey	[A3, A4, A5, A9, A7, A10, A11, A15, A16, A23, A31, A32, A34, A35, A36, A39, A40, A42]
Outro	[A2, A6, A25, A38]

Tabela 5: Tipos de pesquisa usadas para validação das abordagens.

Inovação no Ensino de ES	ID dos artigos
Gamificação no Ensino	[A5, A17, A18, A19, A31, A36]
Ferramentas Colaborativas Online	[A1, A9]
Ensino Lúdico	[A1, A4, A5, A10, A11, A14, A16, A29]
Desenvolvimento de Jogos Educacionais	[A17, A27, A28, A30]
Metodologias Ativas de Ensino	[A1, A5, A8, A10, A15, A21, A22, A23, A24, A25, A26, A29, A32, A34, A35, A39, A40, A42, A44]
Uso de Exemplos em Aula	[A3, A4, A9, A13, A14, A23]
Estratégias de Ensino à Distância	[A5, A8, A17, A18, A20]
Adaptação de Materiais Multimídia	[A4, A14]
Análise de Impacto Pedagógico	[A7, A11, A12, A14, A15, A19, A21, A23, A24, A25, A26, A28, A30, A32, A33, A34, A35, A36, A37, A38, A39, A40, A41, A42, A43, A44, A45, A46, A47, A48]

Tabela 6: Possibilidades de inovação no ensino de Engenharia de Software.

desafios práticos, enquanto A27 recomenda a expansão do uso de metodologias ativas para incluir uma maior variedade de técnicas pedagógicas adaptadas às necessidades específicas dos estudantes.

Os artigos A2 e A3 convergem na importância de adotar abordagens pedagógicas centradas no aluno. Ambos sugerem a implementação de metodologias ativas, como a ABP e PBL, para desenvolver habilidades de resolução de problemas, pensamento crítico e trabalho em equipe, alinhadas com as demandas do mercado de trabalho na área de ES. Já, o artigo A18 propõe investigar a eficácia de ferramentas assíncronas, como fóruns, em contextos de gamificação; e, o artigo A39 sugere continuar o desenvolvimento de ferramentas de recomendação de técnicas de *Design Thinking*.

Os artigos A15 e A20 indicam a utilização do *Design Thinking* para melhorar a educação em ES. E, o artigo A15 foca no desenvolvimento de habilidades interpessoais (soft skills). De modo semelhante, os artigos A13, A22, A32 e A40 também reforçam a importância de desenvolver tanto habilidades interpessoais e habilidades técnicas nos alunos de ES. Estes estudos incluem a investigação do impacto de habilidades não técnicas na empregabilidade dos profissionais, a integração de habilidades de comunicação e trabalho em equipe, e a adaptação de metodologias para melhorar essas competências.

Os artigos A3 e A23 indicam necessidade de desenvolver habilidades práticas nos alunos para prepará-los adequadamente para o mercado de trabalho, através da integração de projetos práticos, estágios supervisionados e experiências de trabalho em equipe no currículo de ensino. Além disso, enfatizam a importância de fornecer oportunidades para os alunos aplicarem os conceitos teóricos aprendidos em sala de aula em situações do mundo real.

O artigo A45 também destaca a importância de desenvolver competências específicas nos alunos de Engenharia de Software como programação, *design* de software e gerenciamento de projetos; além de habilidades interpessoais, como trabalho em equipe, comunicação eficaz e resolução de problemas. A proposta é integrar essas competências ao currículo de ensino para promover uma formação completa e alinhada com as necessidades da indústria de software.

A integração de tecnologias digitais surge como uma tendência comum em artigos como A38, A40 e A48. Esses trabalhos exploram e indicam o uso de plataformas *online*, ferramentas de visualização e a adaptação de conteúdos para ambientes de ERE e híbrido.

Artigos como A6, A16, A22, A38 e A40 discutem a necessidade de adaptar o ensino de ES a diferentes contextos e necessidades. Isso pode envolver a personalização de abordagens educativas, a consideração de diferentes ambientes de aprendizagem e a adaptação de metodologias para atender a variados perfis de estudantes.

Os artigos A24 e A41 discutem metodologias de ensino em ES com foco em atividades práticas e interação com a comunidade de código aberto, proporcionando uma experiência relevante aos alunos. Trabalhos futuros sugerem investigar o impacto de longo prazo dessas metodologias nas carreiras dos alunos e nas contribuições para projetos de código aberto, além de explorar estratégias de ensino e colaboração com comunidades de software livre.

Além disso, ferramentas colaborativas *online* têm sido apontadas como recursos importantes para promover a interação entre alunos e professores. Esses recursos foram utilizados durante o ERE, mas continuaram sendo utilizados. Através dessas ferramentas, é possível realizar atividades em grupo, discussões e compartilhamento de recursos, enriquecendo a experiência de aprendizagem em ES.

O ensino lúdico também é visto como uma alternativa para tornar o aprendizado mais prático e interativo. O uso de abordagens lúdicas, como jogos e simulações, permite aos alunos assimilarem os princípios e práticas dos métodos ágeis de desenvolvimento de software de forma mais dinâmica. Paralelamente, o desenvolvimento de jogos educativos direcionados especificamente para o ensino de ES tem demonstrado ser uma estratégia eficaz e promissora.

Outra tendência que permanece é o uso de metodologias ativas de ensino, que envolvem os alunos de forma mais participativa na construção do conhecimento. Isso pode incluir estudos de caso, projetos práticos e aprendizado baseado em problemas, estimulando o pensamento crítico e a aplicação prática dos conceitos aprendidos. Em linhas gerais, a utilização de exemplos práticos e casos reais em sala de aula tendem a tornar o aprendizado mais contextualizado e aplicável ao contexto da Engenharia de Software.

A adaptação de materiais multimídia também foram apresentados como promissores no processo de ensino-aprendizagem em Engenharia de Software. Recursos como vídeos, animações e infográficos podem ser utilizados para tornar o conteúdo mais dinâmico e acessível, atendendo às diferentes necessidades e estilos de aprendizagem dos alunos.

Por fim, também foram indicados a análise do impacto pedagógico das propostas, estudos ou sugestões relacionadas à avaliação da eficácia das diferentes abordagens apresentadas por meio da coleta de *feedback*, da realização de estudos comparativos e da identificação das melhores práticas no ensino de Engenharia de Software. Tais indicadores apresentam propostas futuras referentes à realização de estudos experimentais, análises de dados, comparações entre diferentes abordagens de ensino, entre outros métodos, de modo a identificar práticas educacionais eficazes para melhorias no processo de ensino e aprendizagem de Engenharia de Software.

5 DISCUSSÃO DOS RESULTADOS

A partir dos resultados obtidos percebe-se algumas tendências em abordagens de ensino de ES e possibilidades de inovação, considerando diferentes estratégias metodológicas. Por enquanto, no estudo, não foram identificadas adoções de recursos de Inteligência Artificial (IA), embora o uso de IA seja uma tendência promissora nessa área. A exemplo, é possível incorporar ferramentas como o GitHub Copilot, que utiliza IA para ajudar na escrita de código, com Aprendizagem Baseada em Problemas (do inglês, Problem-Based Learning - PBL). Utilizando o GitHub Copilot para ensinar práticas de codificação eficientes, promover a aprendizagem por meio de exemplos práticos e feedback em tempo real.

De modo adicional, percebe-se que integrar iniciativas que envolvam o estímulo de soft e hard skills necessárias ao desenvolvimento do profissional de ES é um caminho para enfrentar os desafios identificados no ensino de Engenharia de Software. Neste estudo, foram identificadas algumas abordagens que têm sido comumente adotadas. Entretanto, é importante perceber detalhes de sua implementação, para afirmar a sua efetividade na minimização de desafios como: defasagem curricular, atualização das demandas da indústria, promoção da diversidade e inclusão. Por isso, existe a necessidade de aprofundamento dos resultados alcançados e de estudos complementares a estes resultados, como ampliação do

estudo bibliográfico em outras bases de dados da área de ES e execução de estudos qualitativos que contribuam com a validação desses resultados em ambientes acadêmicos.

Por outro lado, integrar projetos no currículo pode ajudar a minimizar a lacuna entre teoria e a prática no ensino de ES, preparando os alunos para desafios reais no desenvolvimento de software, inclusive o desenvolvimento das habilidades interpessoais. Bem como, estimular a formação contínua dos docentes para garantir que estejam atualizados com as novas tecnologias e metodologias de ensino, para melhor adaptar o currículo e as práticas às necessidades dos alunos e da indústria.

Por fim, percebe-se, comumente, que adotar abordagens diversificadas como apoio ao ensino de ES surge como estratégia para cobrir a lacuna existente entre o currículo de ES e as constantes atualizações da indústria de software. Entretanto, faz-se necessário a execução de estudos qualitativos com alunos, professores e profissionais da indústria para discutir as abordagens educacionais identificadas, perceber pontos fortes e fracos, e sugerir melhorias.

6 LIMITAÇÕES E AMEAÇAS A VALIDADE

A seguir são discutidas algumas principais limitações e ameaças à validade da pesquisa, através da categorização apresentada por Wohlin et al. (2012). Desse modo, a fim de mitigar a validade de construto, a string de busca utilizada pode não incluir todos os sinônimos existentes para a área de ES e pode ser insuficiente para capturar todos os estudos na área. Assim, para minimizar essas potenciais ameaças, foram utilizados sinônimos para os construtos-chave, e foram feitos testes de simulação da execução da string, até se obter a string de busca apresentada na seção 3.1.

Considerando que o estudo se baseia exclusivamente nos artigos publicados no SBES entre 2013 e 2023. Embora essa base de dados seja altamente relevante para o contexto brasileiro, estudos relevantes de outras fontes não foram considerados, o que pode resultar em uma visão parcial do estado da arte. Além disso, o contexto brasileiro, sob a ótica dos SBES, pode não ser representativo de práticas em outros países ou contextos internacionais. A fim de mitigar essa ameaça, pretende-se incluir artigos de outras conferências nacionais e periódicos para obter uma visão mais abrangente das práticas educacionais em Engenharia de Software no Brasil.

Outra ameaça à validade interna percebida foi relacionada à inclusão de artigos, que a depender dos critérios de inclusão e exclusão aplicados, pode estar sujeita a viés de seleção. Para mitigar o viés pessoal no estudo, três estudantes de Graduação conduziram o estudo bibliográfico. Três professores orientadores validaram as etapas e, em caso de divergência, realizaram uma revisão por pares, considerando o conteúdo apresentado no texto dos próprios estudos.

Algumas outras limitações também foram percebidas, como a indisponibilidade de acesso aos artefatos gerados ou utilizados nos artigos analisados nos estudos. A ausência de acesso a materiais suplementares com a descrição destes artefatos dificulta a análise detalhada da abordagem, a verificação dos resultados apresentados e a replicação do estudo.

7 CONSIDERAÇÕES FINAIS

O Simpósio Brasileiro de Engenharia de Software (SBES) é o principal evento de Engenharia de Software da América Latina, integrado ao Congresso Brasileiro de Software: Teoria e Prática (CBSOFT). Este evento vem sendo construído de modo a estimular a participação de profissionais, acadêmicos e estudantes em discussões acerca de resultados, técnicas e perspectivas de pesquisas sobre Engenharia de Software.

Neste sentido, a comunidade do SBES também se empenha a discutir os desafios, boas práticas e inovações no contexto de educação e treinamento em Engenharia de Software. Considerando essa perspectiva, este estudo investigou as principais abordagens adotadas no ensino de Engenharia de Software e suas subáreas, a partir dos artigos oriundos dos anais do SBES, publicados no período de 2013 a 2023. Para isso, foi conduzido um estudo bibliográfico entre 05 de fevereiro e 10 de maio de 2024. Após o processo de seleção e avaliação dos artigos retornados, foram considerados 48 artigos para análise.

Os resultados gerais indicam que o ensino de Engenharia de Software tem se mantido ao longo dos últimos 10 anos, como tópico de investigação relevante, na comunidade de ES e tem sido objeto de estudo por diferentes autores e instituições de ensino, e em diferentes contextos - acadêmicos, para ensino, e de indústria, para treinamento. No contexto de ensino de ES, as subáreas de Processo de Desenvolvimento, Análise/Projeto/Modelagem e Teste de Software são as mais investigadas. Embora, alguns artigos disponibilizam o acesso à materiais suplementares da pesquisa, boa parte deles não dispõem de acesso aos artefatos usados nos estudos. Isto dificulta a reprodutibilidade e uso dessas abordagens, em outros contextos de ensino. A categorização das abordagens disponibilizadas nos artigos são apresentadas nas Tabelas 2, 3 e 4.

Em linhas gerais, as abordagens identificadas para o ensino de ES e suas subáreas tem sido aplicadas através do uso de Jogos, Metodologias de Ensino específicas, Utilização de Ferramentas, Gamificação, Sala de Aula Invertida, *Design Thinking*, dentre outros. Para isso, alguns tipos de pesquisa são comumente usadas para validação dessas abordagens, como o uso de *Survey*, relatos de experiência, experimentos, estudos de caso, entrevistas, estudos bibliográficos/teóricos, dentre outros.

Como trabalhos futuros, é pretendido (1) ampliar a busca por estudos relacionados ao ensino de Engenharia de Software e suas subáreas, de modo a identificar abordagens complementares ou inovadoras referentes à este contexto. Para isso, serão consultadas bibliotecas digitais de conferências e periódicos da área de Computação. Além disso, é necessário (2) compreender como tais abordagens são categorizadas e como contribuem com aspectos de inovação no ensino de ES, de fato. Para, finalmente (3) estruturar um ambiente que promova o fácil acesso às abordagens, de forma ordenada.

DISPONIBILIDADE DE ARTEFATOS

O Material Suplementar desta pesquisa pode ser acessado no seguinte repositório: <https://zenodo.org/records/13768253>.

REFERÊNCIAS

- [1] Cláuvyn Almeida, Marcos Kalinowski, Anderson Uchôa, and Bruno Feijó. 2023. Negative effects of gamification in education software: Systematic mapping and practitioner perceptions. *Information and Software Technology* 156 (2023), 107142.
- [2] Fabiane Barreto Vavassori Benitti. 2017. As a teacher, I want to know what to teach in requirements engineering so that professionals can be better prepared. In *Proceedings of the XXXI Brazilian Symposium on Software Engineering* (Fortaleza, CE, Brazil) (SBES '17). Association for Computing Machinery, New York, NY, USA, 318–327. <https://doi.org/10.1145/3131151.3131185>
- [3] Thiago A. Beppe, Ítalo Linhares de Araújo, Bruno Sabóia Aragão, Ismayle de Sousa Santos, Davi Ximenes, and Rossana M. Castro Andrade. 2018. GreaTest: a card game to motivate the software testing learning. In *Proceedings of the XXXII Brazilian Symposium on Software Engineering* (Sao Carlos, Brazil) (SBES '18). Association for Computing Machinery, New York, NY, USA, 298–307. <https://doi.org/10.1145/3266237.3266254>
- [4] Andréa Sabedra Bordin, Lorenzo Mendes Rodrigues, and Tarcisio Casagrande. 2023. Ensino, Pesquisa e Extensão no Ensino de Engenharia de Software: Um Relato de Experiência. In *Anais do XXXI Workshop sobre Educação em Computação*. SBC, 30–40.
- [5] Pierre Bourque, Robert Dupuis, Alain Abran, James W Moore, and Leonard Tripp. 1999. The guide to the software engineering body of knowledge. *IEEE software* 16, 6 (1999), 35–44.
- [6] A Brito and J Vieira. 2017. '2TScrum' A Board Game to Teach Scrum. In *Proceedings of the XXXI Brazilian Symposium on Software Engineering*. 279–288.
- [7] Vitor Castro and Adam Santos. 2021. Evaluation of the application of gamification in the discipline of Software Engineering in times of pandemic. In *Proceedings of the XXXV Brazilian Symposium on Software Engineering*. 163–169.
- [8] Yandson Costa, Dayanne Gomes, Sebastião Santos, Nathasha Pinto, Davi Viana, Geraldo Braz, and Luis Rivero. 2020. Identifying improvement opportunities in software engineering education at the maranhao state: Listening to voices from academy and industry. In *Proceedings of the XXXIV Brazilian Symposium on Software Engineering*. 860–869.
- [9] Jarbele Coutinho, Wilkerson Andrade, and Patricia Machado. 2021. Teaching exploratory tests through pbl and jitt: An experience report in a context of distributed teams. In *Proceedings of the XXXV Brazilian Symposium on Software Engineering*. 205–214.
- [10] Maria Couto and Breno Miranda. 2023. I10n-trainer: a Tool to Assist in the Training of Localization (I10n) and Internationalization (i18n) Testers. In *Proceedings of the XXXVII Brazilian Symposium on Software Engineering*. 277–282.
- [11] Andrew Diniz da Costa, Hendi Lemos Coelho, Ricardo Almeida Venieris, Carlos José Pereira de Lucena, Gustavo Robichez Carvalho, and Marcelo Fernandes Pereira. 2020. Assessing a Multidisciplinary Group of Undergraduate Students Applying the Challenge Based Learning Methodology to Learn Mobile Development. In *Proceedings of the XXXIV Brazilian Symposium on Software Engineering*. 700–709.
- [12] José Adson OG da Cunha, Gabriel Araújo Marques, Wellington Lourenço Lemos, Uélio Dornelas Câmara Jr, and Francisco JS Vasconcellos. 2018. Software engineering education in Brazil: a mapping study. In *Proceedings of the XXXII Brazilian Symposium on Software Engineering*. 348–356.
- [13] Maria Elizabeth Bianconcini de Almeida and Maria Elisabete Brito Prado. 2003. Criando situações de aprendizagem colaborativa. *Workshop em Informática na Educação - WIE (2003)*, 53–60.
- [14] Lis Ângela De Bortoli. 2018. Non-conventional dynamics in a Software Engineering Course: practical and ludic activities. In *Proceedings of the XXXII Brazilian Symposium on Software Engineering*. 328–337.
- [15] Fabrício de Sousa Pinto and Paulo Caetano Silva. 2017. Gamification applied for software engineering teaching-learning process. In *Proceedings of the XXXI Brazilian Symposium on Software Engineering*. 299–307.
- [16] Luciana Mara Freitas Diniz, Fischer Jônatas Ferreira, and Joao Paulo Diniz. 2021. Interdisciplinaridade no ensino de engenharia de software e interação humano-computador com a utilização de tecnologias digitais: um relato de experiência. In *Anais do XXVII Workshop de Informática na Escola*. SBC, 116–127.
- [17] Maria Elaine Silva dos Santos, Tamires Siqueira Rocha, and Mirko Barbosa Perkusich. 2021. O ensino de engenharia de software no nível superior: um mapeamento sistemático. *Revista Principia-Divulgação Científica e Tecnológica do IFPB* 56 (2021), 116–125.
- [18] Cláudio Marcio Felisbino, Adolfo Gustavo Serra Seca Neto, and Laudelino Cordeiro Bastos. 2018. Supporting to the teaching and learning process in object orientation during the construction of class diagrams. In *Proceedings of the XXXII Brazilian Symposium on Software Engineering*. 338–347.
- [19] Thais Ferreira, Davi Viana, Juliana Fernandes, and Rodrigo Santos. 2018. Identifying emerging topics and difficulties in software engineering education in Brazil. In *Proceedings of the XXXII Brazilian Symposium on Software Engineering*. 230–239.
- [20] Kiev Gama and Higor Oliveira. 2022. An experience report on teaching Scrum principles in a playful way through distant collaboration with online whiteboards. In *Proceedings of the XXXVI Brazilian Symposium on Software Engineering*. 143–152.
- [21] Marcos Kalinowski, Tatiana Escovedo, Fernanda Pina, Adriana Vidal, Ariane Pereira Da Silva, Ricardo Ponsirenas, and Daiana Garibaldi Da Rocha. 2023. Training the Professionals that Industry Needs: The Digital Software Engineering Education Program at PUC-Rio. In *Proceedings of the XXXVII Brazilian Symposium on*

- Software Engineering*. 37–46.
- [22] Barbara Kitchenham and Stuart Charters. 2007. Guidelines for performing systematic literature reviews in software engineering. (2007).
- [23] Felipe Torres Leite, Jarbele CS Coutinho, and Reudismam Rolim de Sousa. 2020. An experience report about challenges of software engineering as a second cycle course. In *Proceedings of the XXXIV Brazilian Symposium on Software Engineering*. 824–833.
- [24] José Lima, Fernanda Alencar, and Wylliams Santos. 2021. A preliminary guide for assertive selection of active methodologies in software engineering education. In *Proceedings of the XXXV Brazilian Symposium on Software Engineering*. 170–179.
- [25] José Vinícius Lima, Mozart de Melo Alves Júnior, Andres Moya, Ricardo Almeida, Patricia Anjos, Maria Lencastre, Roberta Andrade de Araújo Fagundes Fagundes, and Fernanda Alencar. 2019. As Metodologias Ativas e o Ensino em Engenharia de Software: uma revisão sistemática da literatura. In *Anais do XXV Workshop de Informática na Escola*. SBC, 1014–1023.
- [26] Marcelo Maia. 2023. Crossed Teams: Leveraging Student Interaction in Software Engineering Practice. In *Proceedings of the XXXVII Brazilian Symposium on Software Engineering*. 524–533.
- [27] Anna B Marques, Bruna Ferreira, Adriana Lopes, and Williamson Silva. 2020. Stimulating the development of soft skills in software engineering education through design thinking. In *Proceedings of the XXXIV Brazilian Symposium on Software Engineering*. 690–699.
- [28] Andressa de Souza Silva Medeiros and Jorge Marques Prates. 2020. Metodologias Ativas no Ensino de Teste de Software para Alunos com Dedicção Parcial. *Anais do Computer on the Beach* 11, 1 (2020), 548–555.
- [29] Silvana M Melo, Veronica XS Moreira, Leo Natan Paschoal, and Simone RS Souza. 2020. Testing education: a survey on a global scale. In *Proceedings of the XXXIV Brazilian Symposium on Software Engineering*. 554–563.
- [30] Mayara Olivindo, Necio Veras, Windson Viana, Mariela Cortés, and Lincoln Rocha. 2021. Gamifying flipped classes: An experience report in software engineering remote teaching. In *Proceedings of the XXXV Brazilian Symposium on Software Engineering*. 143–152.
- [31] Ana Carolina Oran, Rayfran Rocha Lima, Bruno Gadelha, Nayane Maia, Williamson Silva, and Luis Rivero. 2023. Empowering Technical Skills and Soft Skills in Software Engineering Students through Problem-Based Learning. In *Proceedings of the XXXVII Brazilian Symposium on Software Engineering*. 348–357.
- [32] Sofia Costa Paiva and Dárlinton Barbosa Feres Carvalho. 2018. Software CREATION WORKSHOP: A capstone course for business-oriented software engineering teaching. In *Proceedings of the XXXII Brazilian Symposium on Software Engineering*. 280–288.
- [33] Leo Natan Paschoal, Myke M Oliveira, Silvana M Melo, Ellen F Barbosa, and Simone RS Souza. 2020. Evaluating the impact of software testing education through the flipped classroom model in deriving test requirements. In *Proceedings of the XXXIV Brazilian Symposium on Software Engineering*. 570–579.
- [34] Cristiane Soares Ramos, Ricardo Ajax Dias Kosloski, Elaine Venson, Rejane M da Costa Figueiredo, and Victor Hugo A Deon. 2018. TBL as an active learning-teaching methodology for software engineering courses. In *Proceedings of the XXXII Brazilian Symposium on Software Engineering*. 289–297.
- [35] Thalia S. Santana, Taciana N. Kudo, and Renato F. Bulcao-Neto. 2023. Undergraduates' perspective on a pedagogical architecture to requirements engineering education. In *Proceedings of the XXXVII Brazilian Symposium on Software Engineering (SBES '23)*. Association for Computing Machinery, New York, NY, USA, 422–431. <https://doi.org/10.1145/3611372.3614194>
- [36] Cynthia Pinheiro Santiago, José Wally Mendonça Menezes, and Francisco José Alves de Aquino. 2023. Proposta e Avaliação de uma Metodologia de Aprendizagem Baseada em Projetos em Disciplinas de Engenharia de Software através de uma Sequência Didática. *Revista Brasileira de Informática na Educação* 31, 1 (2023), 31–59.
- [37] Sebastião Henrique Nascimento Santos, Yandson de Jesus Saraiva Costa, Davi Viana dos Santos, Alex Oliveira Barradas Filho, João Batista Bottentuit Junior, and Luis Jorge Enrique Rivero Cabrejos. 2020. Identificando jogos sérios para o ensino de engenharia de software no brasil através de um mapeamento sistemático. *Research, Society and Development* 9, 7 (2020), e329973702–e329973702.
- [38] Fernanda Gomes Silva, Moara Sousa Brito Lessa, Nádia da Luz Lopes, and Christina von Flach G. Chavez. 2020. Teaching UML Models with FLOSS Projects: A study carried out during the period of social isolation imposed by the COVID-19 pandemic. In *Proceedings of the XXXIV Brazilian Symposium on Software Engineering*. 483–492.
- [39] Williamson Silva, Bruno Gadelha, Igor Steinmacher, and Tayana Conte. 2018. What are the differences between group and individual modeling when learning UML?. In *Proceedings of the XXXII Brazilian Symposium on Software Engineering*. 308–317.
- [40] Tamires AS Sousa and Anna BS Marques. 2020. LEARN Board Game: A game for teaching Software Architecture created through Design Science Research. In *Proceedings of the XXXIV Brazilian Symposium on Software Engineering*. 834–843.
- [41] Anderson Felipe Souza, Bruna Ferreira, Natasha Valentim, and Tayana Conte. 2018. An experience report on teaching multiple design thinking techniques to software engineering students. In *Proceedings of the XXXII Brazilian Symposium on Software Engineering*. 220–229.
- [42] Patricia Torres, Paulo Alcantara, and Esrom Adriano Freitas Irala. 2017. Grupos de consenso: Uma proposta de aprendizagem colaborativa para o processo de ensino-aprendizagem. *Revista Diálogo Educacional* 4, 13 (2017), 129–143.
- [43] Fernando Trinta, Paulo AL Rego, and Windson Viana. 2020. Teaching development of distributed software during covid-19: An experience report in brazil. In *Proceedings of the XXXIV Brazilian Symposium on Software Engineering*. 616–625.
- [44] Fernando Antonio Mota Trinta and Emanuele Santos. 2017. Teaching Software Development for the Cloud: An Experience Report. In *Proceedings of the XXXI Brazilian Symposium on Software Engineering*. 338–347.
- [45] Pedro Henrique Dias Valle, Rafaela Vilela Rocha, and José Carlos Maldonado. 2017. Testing Game: An Educational Game to Support Software Testing Education. In *Proceedings of the XXXI Brazilian Symposium on Software Engineering (Fortaleza, CE, Brazil) (SBES '17)*. Association for Computing Machinery, New York, NY, USA, 289–298. <https://doi.org/10.1145/3131151.3131182>
- [46] Jéssyka Vilela and Carla Silva. 2023. An Experience Report on the use of Problem-based learning and Design Thinking in a Requirements Engineering Postgraduate Course. In *Proceedings of the XXXVII Brazilian Symposium on Software Engineering*. 432–441.
- [47] Avelino F Zorzo, Daltro Nunes, Ecivaldo S Matos, Igor Steinmacher, Jair C Leite, Renata Araujo, Ronaldo CM Correia, and Simone Martins. 2017. Referenciais de Formação para os Cursos de Graduação em Computação 2017. (2017).