

UNIVERSIDADE FEDERAL DE SÃO CARLOS  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA  
DEPARTAMENTO DE COMPUTAÇÃO

GUSTAVO DAVID LEITE BASTOS

UM ESTUDO SOBRE JOGOS ELETRÔNICOS PARA O  
ENSINO DE ENGENHARIA DE SOFTWARE

SÃO CARLOS -SP  
2024

GUSTAVO DAVID LEITE BASTOS

UM ESTUDO SOBRE JOGOS ELETRÔNICOS PARA O ENSINO DE ENGENHARIA DE  
SOFTWARE

Trabalho de conclusão de curso  
apresentado ao Curso de Graduação em  
ao Departamento de Computação da  
Universidade Federal de São Carlos,  
para obtenção do título de bacharel em  
Ciências da Computação.

Orientador: Delano Beder

São Carlos-SP  
2024

## RESUMO

Nos estudos levantados, o ensino de engenharia de software, especialmente nos estágios iniciais do ensino de computação, é apontado diversas vezes como altamente teórico e abstrato (CALDERÓN *et al.*, 2017; CLEGG *et al.*, 2017; MONTENEGRO *et al.*, 2017; PETRI *et al.*, 2017; PIEPER *et al.*, 2017; ALATRISTA-SALAS & NUNEZ-DEL-PRADO, 2018; LARENAS *et al.*, 2018; LÓPEZ-FERNÁNDEZ *et al.*, 2022), o que traz desafios para ensinar sobre de maneira intuitiva e acessível, o que, por sua vez, leva a uma desmotivação, em geral, por parte dos alunos em aprender seus conceitos e teorias, uma vez que não percebiam aplicações diretas nos momentos iniciais dos cursos de computação e engenharia de software. No entanto, a engenharia de software é essencial para reduzir custos e desperdícios de recursos (pessoal, tempo e financeiro), e torna-se necessário que durante o treinamento dos estudantes da área a exerçam de maneira prática das funções esperadas de um engenheiro de software (MONSALVE, 2018). Para contornar o problema, diferentes metodologias em diferentes áreas do ensino são estudadas como alternativas, e uma delas é a gamificação (utilização de elementos de jogos em contextos não-lúdicos) e o uso de jogos sérios (jogos com objetivos além do entretenimento, geralmente voltados ao ensino ou treinamento de técnicas ou habilidades em um contexto seguro). Essa pesquisa tem como objetivo fazer um levantamento de quais jogos digitais estão sendo utilizados no ensino de Engenharia de Software por meio de revisão sistemática da literatura e determinar quais áreas de ES são apoiadas por esses jogos, bem como os meios utilizados para avaliar os jogos e a sua aplicação no ensino.

**Palavras-chave:** Engenharia de Software. Jogos Eletrônicos. Ensino. Aprendizado.

## ABSTRACT

The reviewed work converges on the point about software engineering, particularly in initial stages of computer education, being highly theoretical and abstract (CALDERÓN *et al.*, 2017; CLEGG *et al.*, 2017; MONTENEGRO *et al.*, 2017; PETRI *et al.*, 2017; PIEPER *et al.*, 2017; ALATRISTA-SALAS & NUNEZ-DEL-PRADO, 2018; LARENAS *et al.*, 2018; LÓPEZ-FERNÁNDEZ *et al.*, 2022), which causes mass loss of interest and difficulty to learn its basic concepts and theories, amongst students, given the gap between in-class theory and real applications for that. However, software engineering is essential to reduce production and post-production costs and maintenance (people, time and money). To counter such a problem, different methods from different knowledge areas have been studied as complementary methods to motivate and facilitate the learning of SE. Two examples are gamification (applying game elements in a non-game context) and serious games (games with goals beyond entertainment - usually education or training of skills - in a safe environment). This work focuses on digital serious games and aims to survey which digital games are used to support SE learning/teaching and which areas of SE are supported by such games, as well as the means to evaluate and assess its impact on learning.

**Keywords:** Software Engineering. Electronic Games. Teaching. Learning.

## **AGRADECIMENTOS**

A minha família, que desde sempre me apoiou a seguir em frente e continuar com os estudos e nunca desistir, pois “o conhecimento é a única coisa que ninguém tira de nós”. A todos os meus professores, cada um que me apontou o caminho para que eu pudesse me guiar por eles e que diversas vezes me mostraram o caminho para vencer obstáculos aparentemente intransponíveis. Aos meus amigos mais próximos do coração, recentes ou de longa data, que por vezes foram a rocha que eu precisava para me levantar e seguir em frente. Por fim, mas não menos importante, a todos aqueles que em algum momento me estenderam a mão, que me ajudaram, foram muitos. A todos, minha eterna gratidão por mais essa conquista alcançada, e que lhes seja retribuído muitas vezes.

## LISTA DE FIGURAS E TABELAS

Figura 1 - Distribuição de jogos de acordo com a área de ES apoiada. ....	29
Figura 2 - Proporção de estudos por cada modalidade de avaliação da aplicação do jogo no processo de ensino. ....	34

## **LISTA DE TABELAS**

Tabela 1 - Estudos selecionados durante a execução da pesquisa e suas notas. ... 16

## SUMÁRIO

<b>1 - INTRODUÇÃO.....</b>	<b>09</b>
<b>2 - ESTUDOS RELACIONADOS.....</b>	<b>10</b>
<b>3 - PROTOCOLO DA REVISÃO SISTEMÁTICA DA LITERATURA.....</b>	<b>11</b>
<b>3.1 - Planejamento.....</b>	<b>12</b>
3.1.1 - Objetivo e Questões de Pesquisa.....	12
3.1.2 - Bases e Strings de Busca.....	12
3.1.3 - Critérios de Seleção de Estudos.....	12
3.1.4 - Verificação da Qualidade.....	13
3.1.5 - Formulário de Extração de Dados.....	13
<b>3.2 - Condução.....</b>	<b>14</b>
3.2.1 - Busca e Seleção de Estudos.....	14
3.2.2 - Extração de Dados.....	14
<b>3.3 - Relatório.....</b>	<b>15</b>
<b>4 - EXECUÇÃO DA RSL.....</b>	<b>15</b>
<b>5 - RESULTADOS E DISCUSSÕES.....</b>	<b>17</b>
5.1 - Quais jogos digitais utilizados para auxiliar o ensino de ES?.....	18
5.2 - Quais áreas da Engenharia de Software são apoiadas pelos jogos?.....	29
5.3 - De quais formas foram avaliados os jogos e a aplicação do jogo no ensino?.....	32
5.4 - Discussões finais.....	34
<b>6 - Ameaças À Validade.....</b>	<b>35</b>
<b>7 - Conclusão.....</b>	<b>36</b>
<b>REFERÊNCIAS.....</b>	<b>37</b>
<b>APENDICE A.....</b>	<b>42</b>

## 1 - INTRODUÇÃO

O problema do ensino de engenharia de software para alunos nos cursos de graduação é encontrado e discutido em diversos dos trabalhos levantados, e diversas causas são apontadas (CALDERÓN *et al.*, 2017; CLEGG *et al.*, 2017; MONTENEGRO *et al.*, 2017; PETRI *et al.*, 2017; PIEPER *et al.*, 2017; ALATRISTA-SALAS & NUNEZ-DEL-PRADO, 2018; LARENAS *et al.*, 2018; LÓPEZ-FERNÁNDEZ *et al.*, 2022): a natureza altamente teórica e abstrata dos conceitos de ES (engenharia de software); a falta de experiência em geral, por parte dos alunos, em projetos com demanda de técnicas ou conceitos da área; a incapacidade de simular um ambiente mais próximo do real no desempenho da função de um engenheiro de software, de maneira observável, controlada e segura; a ineficiência de aulas puramente expositivas e leitura de bibliografia como método de ensino eficaz preparar profissionais devidamente capazes.

Neste contexto, diferentes metodologias de ensino são experimentadas como alternativa à aula puramente expositiva combinada à realização de exercícios e seminários. Como exemplo de metodologias alternativas para o ensino, podemos citar a gamificação, que consiste na aplicação de elementos de jogos em contextos não-lúdicos (por exemplo, a atribuição de pontuação fictícia e/ou virtual, competição e cooperação entre membros de equipes, resolução de desafios com recompensas imediatas e mais perceptíveis - por exemplo troféus ou medalhas simbólicas). No ensino de ES, a aplicação direta de jogos, principalmente aqueles autorais de professores e educadores voltados a contextos específicos - sendo jogos físicos (*board games* ou jogos de cartas, ou ainda jogos de movimentação e comunicação) ou digitais (jogos eletrônicos para diversas plataformas, mas de maneira mais comum, para computadores), pode ser um aliado, principalmente quando utilizada em conjunto com os materiais e recursos tradicionais, para motivar e engajar os estudantes da área, principalmente aqueles sem experiência no mercado.

Este trabalho busca concentrar-se especificamente nos jogos digitais utilizados para o ensino de Engenharia de Software (ES), realizando uma Revisão Sistemática da Literatura (RSL). Um dos objetivos da RSL é investigar se os jogos utilizados de fato tornam o aprendizado mais fluido e engajante, e para isso, fazer um levantamento de quais jogos eletrônicos são utilizados para o ensino de ES, entender quais áreas da ES são apoiadas pelos jogos e como o impacto do uso de

jogos é avaliado no ensino.

As áreas identificadas pela RSL foram: conceitos básicos e fundamentais da ES; diagrama de estados; testes; requisitos; processos e normas; metodologias ágeis; decisões estratégicas; e um caso específico sobre segurança da informação no contexto de ES. Os métodos de avaliação levantados mais recorrentes foram os questionários de *feedback* não padronizados, ou seja, desenhados para cada estudo. Houve exemplos de dois modelos padronizados, *MEEGA* (Model for Evaluation of Educational Games) (WANGENHEIM & BORGATTO, 2011) e *UGALCO framework* (baseado em “*Usability*”, “*Motivation*”, “*Learning*”, “*User Experience*” & “*Communicability*”) (PEIXOTO *et al.*, 2014), alguns estudos mencionaram jogos cujas próprias métricas e desempenho dos estudantes serviu como base para avaliação e outros utilizaram notas de provas e exames teóricos como base de comparação entre grupos de alunos que utilizaram e grupos de alunos que não utilizaram jogos como material de apoio.

## **2 - ESTUDOS RELACIONADOS**

Esta seção tem como objetivo discutir brevemente trabalhos e pesquisas levantados, relacionados a este trabalho. Para fazer o levantamento referente a esta sessão, foi feita uma busca nas bases “*IEEE Xplore*” e “*Scopus Elsevier*” sobre RSL e MSL voltados ao uso de jogos eletrônicos no ensino das engenharia de software.

A busca foi limitada aos últimos 5 anos, mas apenas 32 estudos eram do tipo RSL ou MSL. Destes, foram selecionados lendo título e resumo quais tratavam de jogos e engenharia de software. Três mapeamentos e revisões foram considerados pertinentes aos critérios de seleção para esta sessão.

Tumpa *et al.* (2024) colocam em evidência a discussão recorrente sobre o tema do preparo efetivo para o mercado de gerentes de projetos na educação superior. O trabalho levanta, como potencial auxílio para o preparo eficaz na gerência de projetos durante os cursos acadêmicos altamente teóricos, jogos do tipo simulação, que podem trazer aos estudantes da área uma prévia da prática do gerente de projetos, o que é esperado e o treinamento para fazê-lo em um ambiente seguro onde pode perceber e entender as consequências de tomadas de decisões. Ainda, o estudo levanta a necessidade de grande planejamento para que jogos de simulação possam ser eficazes em ensinar e treinar futuros gerentes de projetos.

Borges & Souza (2024) tinham como objetivo identificar as *soft skills* (habilidades não técnicas, ou ainda habilidades interpessoais) necessárias aos futuros engenheiros de software, bem como metodologias que auxiliam nesse ensino que, segundo os autores, deve ser desenvolvido a partir da graduação, mantendo os estudantes engajados. Como resposta, os autores propuseram um *framework* que objetiva cumprir esse papel, cobrindo as *soft skills* e mantendo o engajamento dos alunos de maneira dinâmica e interessante aos mesmos. O *framework* é baseado em gamificação e *role-playing* (interpretação de papéis), contando também com metodologias auxiliares como PBL (*Problem-Based Learning*). No estudo, não houve menção da validação da aplicação do *framework* como ferramenta eficaz no desenvolvimento de *soft skills*, embora como “trabalhos futuros” tenha sido mencionada a possibilidade de combiná-lo com alguma metodologia que também desenvolva *hard skills* (habilidades técnicas e teóricas).

Kharbouch (2025) oferece um contraponto interessante sobre como as metodologias gamificadas se concentram em grupos específicos de jogadores de acordo com a nomenclatura da ferramenta *Hexad* (ferramenta com o objetivo de entender a motivação que impulsiona os atos de cada jogador). De acordo com a ferramenta, os grupos mais comuns de tipos de jogadores são *Achievers* (48.64%) e *Players* (47.29%), embora os perfis de jogadores não sejam mutuamente exclusivos (um jogador pode ter mais de um perfil). A revisão comparou perfis de acordo com o *Big Five* (Modelo dos Cinco Grandes Fatores) (RAMMSTEDT & JOHN, 2007), e o perfil de jogador na ferramenta *Hexad*, encontrando correlações positivas, com a exceção do perfil *Philanthropist*, porém não foi detectado jogadores com esse perfil nos estudos sobre Jogos Sérios levantados pelos autores.

### **3 - PROTOCOLO DA REVISÃO SISTEMÁTICA DA LITERATURA**

A revisão foi feita utilizando a plataforma de Revisão e Mapeamento sistemático *PARSIFAL* (parsif.al), que divide o processo em três grandes etapas (Planejamento, Condução e Relatório), subdividindo cada etapa em outras menores. O protocolo da plataforma, exportado por meio da própria, foi anexado no “APÊNDICE A” deste trabalho. A escolha da metodologia enquanto Revisão Sistemática da Literatura ajudou em diversos momentos, desde o início, dando os primeiros passos da pesquisa, quanto durante a condução, uma vez determinado os

objetivos da pesquisa, a forma estruturada de coleta de dados por meio de formulários, ajudou a recuperar e sintetizar posteriormente os resultados, bem como classificar e avaliar a qualidade e adequação dos estudos levantados à pesquisa.

### **3.1 - Planejamento**

O planejamento consiste na definição do protocolo de pesquisa e levantamento dos estudos, verificação da qualidade por meio de um questionário graduado, e o formulário de extração de dados.

#### **3.1.1 - Objetivo e Questões de Pesquisa**

O objetivo primário da pesquisa (e também a principal questão de pesquisa) é levantar **(I)** quais jogos digitais eram utilizados para auxiliar o ensino de Engenharia de Software. Também aqui definimos como objetivos secundários (e questões de pesquisa secundárias) determinar, dos jogos levantados, **(II)** quais as áreas de ES apoiadas por esses jogos, **(III)** se os jogos ainda estavam disponíveis, de maneira gratuita online, **(IV)** quais as maneiras, se houvesse, de avaliação do jogo e da aplicação do jogo para o aprendizado.

#### **3.1.2 - Bases e String de Busca**

As bases utilizadas inicialmente seriam a *Scopus Elsevier*, *IEEE XPLORE* e *ACM Digital Library*. No entanto, ao decorrer da pesquisa, o acesso à base *ACM Digital Library* foi perdido por meio do Acesso CAFe da CAPES, então a base em questão foi removida, então os estudos relacionados a essa base exclusivamente foram desconsiderados. As *strings* de busca foram criadas utilizando combinações das palavras-chave na forma de “*wildcards*” (radicais seguidas de variações de sufixos) das palavras e expressões: “*Software Engineering*”, “*Game*”, “*Teach*” e “*Learn*”.

#### **3.1.3 - Critérios de Seleção dos Estudos**

Os critérios definidos para inclusão dos estudos recuperados foram dois: “Trata de métodos de avaliação de jogos digitais utilizados no ensino de ES” e “Trata de um ou mais jogos eletrônicos utilizados para auxiliar o ensino de ES”.

Para exclusão, os critérios foram:

- Estudo trata apenas de Gamificação
- Estudo trata de Jogos Não-Digitais
- Estudo não disponível publicamente (pelo acesso CAFE da CAPES)
- Estudo não trata de Engenharia de Software
- Trata apenas do Desenvolvimento de Jogos
- Trabalho não-pertinente (*Proceedings* de conferências, etc)
- Versões mais recentes do trabalho foram incluídas
- Apenas Trabalhos em Português ou Inglês

Os critérios foram aplicados após leitura do Título e Resumo dos estudos recuperados.

### 3.1.4 - Verificação da Qualidade

A verificação da qualidade dos estudos recuperados foi feita por meio de pontuação em 3 perguntas, cada uma com 3 possibilidades de resposta.

- 1ª Pergunta: O estudo menciona um ou mais jogos eletrônicos utilizados para ensino de ES?
- 2ª Pergunta: Realiza avaliação do processo (aplicação do jogo no ensino)?
- 3ª Pergunta: O(s) jogo(s) mencionado(s) está(ão) desenvolvido(s) e disponível(is)?

As respostas possíveis sendo: **Sim** (1.0), **Parcialmente** (0.5) e **Não** (0.0), sendo a pontuação total a soma das respostas a cada pergunta, que podem variar de 0.0 (caso a resposta fosse “**Não**” para todas as perguntas) até 3.0 (“**Sim**” para todas as perguntas). Foram considerados adequados à RSL aqueles estudos que atingiram pelo menos “2.0”, ou seja, pelo menos uma resposta “**Sim**” e “**Parcialmente**” nas outras duas perguntas, ou “**Sim**” para duas perguntas.

### 3.1.5 - Formulário de Extração de Dados

A extração de dados é realizada por meio de uma lista de questões a serem respondidas, bem como o tipo de resposta esperada (textual, booleana, numérica,

valorada, etc.). Essa extração é feita após a leitura completa de cada estudo e a resposta é guardada para futura análise e relatório.

### 3.2 - Condução

Nesta etapa, os trabalhos são recuperados, a verificação de qualidade é feita, em seguida a extração dos dados relevantes à revisão, e por fim uma análise de dados preliminar é feita de maneira automática. Essa análise reporta quantos estudos foram recuperados, quantos eram de quais bases, quantos foram aceitos e quantos rejeitados, duplicatas, quantidades de estudo por ano, entre outros dados.

#### 3.2.1 - Busca e Seleção de Estudos

A recuperação de estudos e trabalhos pertinentes à revisão foi feita utilizando as seguintes *strings* de busca nas bases mencionadas.

- A. **IEEE XPLORE:** ("Document Title":"software engineering" AND "Document Title":"gam\*") AND ("Document Title":"teach\*" OR "Document Title":"learn\*") AND ("Abstract":"software engineering" AND "Abstract":"gam\*") AND ("Abstract":"teach\*" OR "Abstract":"learn\*")
- B. **Scopus Elsevier:** Advanced query: TITLE-ABS-KEY ( {software engineering} AND game AND ( teach OR learn ) ) AND PUBYEAR > 2012 AND PUBYEAR < 2024 AND ( LIMIT-TO ( LANGUAGE , "English" ) OR LIMIT-TO ( LANGUAGE , "Portuguese" ) )

Em seguida, os estudos são importados à plataforma por meio de exportação de um arquivo “.bibtex” das bases e incluída na plataforma, que por sua vez, oferece uma maneira ágil de acessar “Título” e “Resumo” de cada estudo, marcá-los como aceito ou recusado e escolher o critério de exclusão ou aceitação, bem como fazer anotações referentes a cada decisão.

#### 3.2.2 - Extração de Dados

A extração é feita respondendo a cada questão do formulário de extração de dados após a leitura crítica de cada trabalho recuperado, dando ênfase às questões

de extração de dados. As questões, nesse caso, foram:

- Qual(is) o(s) jogo(s) mencionado(s)?
- O(s) jogo(s) mencionado(s) está(ão) disponível(is) para uso/teste?
- Qual o modo de avaliação do jogo?
- Qual o modo de avaliação do aprendizado?
- Qual(is) área(s) da ES são apoiadas pelo(s) jogo(s) mencionado(s)?

### **3.3 - Relatório**

Essa etapa oferece algumas análises e relatos automáticos convenientes a análises, como gráficos de anos dos estudos selecionados, respostas aos formulários de extração de dados, títulos e autores dos mesmos.

## **4 - Execução da RSL**

Nesta revisão foram recuperados um total de 225 trabalhos das bases mencionadas utilizando as *strings* definidas na seção “3.2.1 - Busca e Seleção de Estudos”, itens “A” e “B”. A partir dos quais foram removidos os estudos duplicados, e em seguida, 27 foram selecionados pelos critérios de seleção após a leitura do título e resumo, dos quais 20 atingiram pontuação igual ou superior a 2.0 na verificação de qualidade (7 estudos dos 28 atingiram 1.5 ou menos, apenas).

Os 20 estudos com a pontuação igual ou superior a 2.0 passaram pelo processo de extração de dados, no qual cada pergunta definida na seção “3.2.2 - Extração de Dados” é respondida para cada trabalho recuperado após sua leitura completa. Tendo em vista o principal objetivo desta revisão, levantar os jogos digitais utilizados para apoiar o ensino de Engenharia de Software, os jogos mencionados foram listados, em especial aqueles que estão disponíveis de maneira gratuita e online. Os estudos selecionados e a nota individual de cada um estão listados na “Tabela 1”.

Foi realizada uma busca simplificada por trabalhos relacionados, de caráter semelhante a este, nas bases especificadas, também por meio do acesso CAFe, no portal de periódicos da CAPES.

Tabela 1 - Estudos selecionados durante a execução da pesquisa e suas notas. (Fonte: autoria própria)

<b>Tabela de Estudos Aceitos</b>	
<b>Estudo</b>	<b>Pontuação</b>
PIEPER, J.; LUETH, O.; GOEDICKE, M.; FORBRIG, P. A case study of software engineering methods education supported by digital game-based learning: Applying the SEMAT Essence kernel in games and course projects. <i>In: EDUCATION CONFERENCE (EDUCON)</i> , 8, 2017, Atenas - Grécia. IEEE, 2017, p. 1689-1699. DOI: 10.1109/EDUCON.2017.7943076	3,0
GORDILLO, A.; LÓPEZ-FERNÁNDEZ, D.; TOVAR, E.; MAYOR, J. A Comparison of the Usefulness of Game-Based Learning and Video-Based Learning for Teaching Software Engineering in Online Environments. <i>In: FRONTIERS IN EDUCATION CONFERENCE (FIE)</i> , 53, 2023, College Station, Texas, Estados Unidos da America. IEEE, 2023, p. 1-7. DOI: 10.1109/FIE58773.2023.10343449	3,0
LØVGREN, D. E. H.; LI, J.; OYETOYAN, T. D. A Data-Driven Security Game to Facilitate Information Security Education. <i>In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING: COMPANION PROCEEDINGS (ICSE-COMPANION)</i> , 41, 2019, Montreal, QC, Canada. IEEE, 2019, p. 256-257, DOI: 0.1109/ICSE-Companion.2019.00102.	3,0
PETRI, G.; WANGENHEIM, C. G.; BORGATTO, A. F. A Large-Scale Evaluation of a Model for the Evaluation of Games for Teaching Software Engineering. <i>In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING: SOFTWARE ENGINEERING EDUCATION AND TRAINING TRACK (ICSE-SEET)</i> , 39, 2017, Buenos Aires, Argentina. IEEE, 2017, p. 180-189, DOI: 10.1109/ICSE-SEET.2017.11.	2,5
ROSALES, S. R.; REYES, S. V.; CISNEROS, J. L. V.; SIGG M. L. A serious game to promote object oriented programming and software engineering basic concepts learning. <i>In: INTERNATIONAL CONFERENCE IN SOFTWARE ENGINEERING RESEARCH AND INNOVATION (CONISOFT)</i> , 4, 2016, Puebla, México. IEEE, p. 97-103, DOI: 10.1109/CONISOFT.2016.23.	2,0
FURTADO, L. S.; OLIVEIRA, S. R. B. A teaching proposal for the software measurement process using gamification: an experimental study. <i>In: IEEE FRONTIERS IN EDUCATION CONFERENCE (FIE)</i> , 50, 2020, Uppsala, Sweden, IEEE, 2020, p. 1-8. DOI: 10.1109/FIE44824.2020.9274194.	2,0
MONTENEGRO, C. H.; ASTUDILLO, H.; ÁLVAREZ, M. C. G. ATAM-RPG: A role-playing game to teach Architecture Trade-off Analysis Method (ATAM). <i>In: LATIN AMERICAN COMPUTER CONFERENCE (CLEI)</i> , 43, 2017, Cordoba, Argentina. IEEE, 2017, p. 1-9. DOI: 10.1109/CLEI.2017.8226416.	2,0
LARENAS F.; MARÍN B.; GIACHETTI G. Classutopia: A Serious Game for Conceptual Modeling Design. <i>In: INTERNATIONAL JOURNAL OF SOFTWARE ENGINEERING AND KNOWLEDGE ENGINEERING</i> , 28:11n12 (edição especial), 2018, Santiago, Chile. World Scientific Publishing Company, 2018, p. 1679-1699. DOI: 10.1142/S0218194018400235.	2,5
SPETH S.; HOFMEISTER, L.; BECKER, S.; BREITENBÜCHER, U. Gamify-IT: A Web-Based Gaming Platform for Software Engineering Education. <i>In: INTERNATIONAL WORKSHOP ON SOFTWARE ENGINEERING EDUCATION FOR THE NEXT GENERATION (SEENG)</i> , 5, 2023, Melbourne, Australia. IEEE, 2023, p. 25-28. DOI: 10.1109/SEENG59157.2023.00009.	3,0
PRASETYA, W. <i>et al.</i> Having Fun in Learning Formal Specifications. <i>In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING: SOFTWARE ENGINEERING EDUCATION AND TRAINING (ICSE-SEET)</i> , 41, 2019, Montreal, QC, Canada. IEEE, 2019,	2,5

p. 192-196. DOI: 10.1109/ICSE-SEET.2019.00028.	
ANDRADE, L.; GRYNBERG E.; SCHOTS, M.; WERNECK, V. M. B. InspectorX 2.0: Developing a Multi-Device Game for Software Inspection Education. <i>In: CONFERENCE ON SOFTWARE ENGINEERING EDUCATION AND TRAINING (CSEE&amp;T)</i> , 32, 2020, Munich, Germany. IEEE, 2020, p. 1-4. DOI: 10.1109/CSEET49119.2020.9206210.	2,0
CALDERÓN, A.; RUIZ, M.; ORTA, E. Integrating Serious Games as Learning Resources in a Software Project Management Course: the Case of ProDec. <i>In: INTERNATIONAL WORKSHOP ON SOFTWARE ENGINEERING CURRICULA FOR MILLENNIALS (SECM)</i> , 1, 2017, Buenos Aires, Argentina. IEEE, 2017, p. 21-27. DOI: 10.1109/SECM.2017.3.	2,0
LÓPEZ-FERNÁNDEZ, D.; MAYOR, J.; PÉREZ, J.; GORDILLO, A. Learning and Motivational Impact of Using a Virtual Reality Serious Video Game to Learn Scrum. <b>IEEE Transactions on Games</b> , v. 15, n. 3, p. 430-439, Outubro, 2022. DOI: 10.1109/TG.2022.3213127.	2,0
LINO, J. E. N. <i>et al.</i> Project Management Game 2D (PMG-2D), A Serious Game to Assist Software Project Managers Training. <i>FRONTIERS IN EDUCATION CONFERENCE (FIE)</i> , 2015, El Paso, TX, USA. IEEE, 2015, p. 1-8. DOI: 10.1109/FIE.2015.7344168.	2,0
MONSALVE, E. S. <i>et al.</i> SimuES-W: A Collaborative Game to Improve Software Engineering Teaching. <b>Comp. y Sist.</b> , Ciudad de México, v. 22, n. 3, p. 953-983, jul./set. 2018. DOI: 10.13053/CyS-22-3-2711.	2,5
AYDANA, U.; YILMAZ, M.; CLARKE, P. M.; O'CONNOR, R. V. Teaching ISO/IEC 12207 software lifecycle processes: A serious game approach. <b>Computer Standards &amp; Interfaces</b> , v. 54, n. 3, p. 129-138, nov. 2017. DOI: 10.1016/j.csi.2016.11.014.	2,5
ALATRISTA-SALAS, H.; NUNEZ-DEL-PRADO, M. Teaching Software Engineering through Computer Games. <i>WORLD ENGINEERING EDUCATION CONFERENCE (EDUNINE)</i> , 2018, Buenos Aires, Argentina. IEEE, 2018, p. 1-4. DOI: 10.1109/EDUNINE.2018.8450996.	3,0
CLEGG, B. S.; ROJAS, J. M.; FRASER, G. Teaching Software Testing Concepts Using a Mutation Testing Game. <i>INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING: SOFTWARE ENGINEERING EDUCATION AND TRAINING TRACK (ICSE-SEET)</i> , 39, 2017, Buenos Aires, Argentina. IEEE, 2017, p. 33-36. DOI: 10.1109/ICSE-SEET.2017.1.	2,0
OZCELIK, E.; CAGILTAY, N. E., OZCELIK, N. S. The effect of uncertainty on learning in game-like environments. <b>Computers &amp; Education</b> , v. 62, p. 12-20, set. 2013. DOI: 10.1016/j.compedu.2013.02.009.	2,0
PEIXOTO, D. C. C.; RESENDE, R. F.; PÁDUA, C. I. P. S. The Issues of Adopting Simulation Games in Software Engineering Classes. <i>FRONTIERS IN EDUCATION CONFERENCE (FIE)</i> , 2015, El Paso, TX, USA. IEEE, p. 1-8. DOI: 10.1109/FIE.2015.7344071.	2,5

## 5 - Resultados e Discussões

As questões de pesquisa utilizadas para direcionar o objetivo deste estudo foram respondidas como segue. Cada uma delas teve como objetivo levantar o estado da arte sobre o uso de jogos eletrônicos no ensino de engenharia de software.

## 5.1 - Quais jogos digitais utilizados para auxiliar o ensino de ES?

Os jogos levantados durante a revisão foram:

### - **Essence Kernel Puzzler (PIEPER et al., 2017)**

O jogo tem como objetivo ajudar os estudantes a se familiarizar com os conceitos e vocabulários da *Software Engineering Method and Theory (SEMAT) Essence Kernel*, utilizando um simples diagrama de associação de “Nomes” e “Definições” para tornar mais interessante o processo de associação do que pela simples leitura de cada definição. O jogo está desenvolvido completamente e foi disponibilizado online para qualquer um que acesse o *link* e insira um apelido. A avaliação foi feita por meio de *feedback* coletado pelos ministrantes do curso dos estudantes que participaram do estudo e houve um nível de aprovação geral de 92% por parte dos alunos reportando que o jogo ajudou a compreender as relações entre termos e definições.

### - **Essence Simulation Game (PIEPER et al., 2017)**

O jogo simula um ambiente de engenharia de software que leva os jogadores a tomarem decisões com base no *Essence Kernel*. Diferente do “*Essence Puzzler*”, o “*Essence Simulation Game*” busca um maior nível de entretenimento, buscando um engajamento maior, trazendo oportunidades de aprendizado tanto individual quanto interações em equipe. Foi proposto como reforço para fixação dos conceitos adquiridos durante o *Puzzler*, bem como a introdução de novos conceitos e trazer um primeiro contato com a experiência de tomada de decisões na prática. Também como o “*Puzzler*”, o “*Simulation Game*” foi avaliado por meio de *feedback* dos alunos participantes, que avaliaram o jogo como uma maneira mais divertida de aprendizado, ainda que o nível de ganho de conhecimento medido entre as turmas avaliadas tenha sido considerado equivalente.

### - **Jogo educacional criado com a plataforma SGAME (GORDILLO et al., 2023)**

O jogo não possui um nome propriamente dito, mas trata sobre conceitos de ES como: modularidade, acoplamento, coesão, abstração e ocultação de informações. Ele utiliza um ambiente semelhante ao jogo “*Flappy Bird*”, e cada vez que o jogador colide com os obstáculos, um recurso educacional emerge na tela (com questões relacionadas a ES). Em caso de resposta correta, segue o jogo; caso contrário, o jogador deve reiniciar o jogo. Foi desenvolvido utilizando a plataforma “*SGAME*” (GORDILLO *et al.*, 2011), utilizada para o desenvolvimento de jogos sérios sem a necessidade de conhecimento em programação. Aqui foi observado a competição entre pontos como um possível motivador para os jogadores se superarem.

- **Data-Driven Security Game (DdSG) (LØVGREEN *et al.*, 2019)**

Trata-se de um jogo no estilo Defesa de Torres (*Tower Defense*) voltado ao ensino de segurança de software, mais especificamente sobre padrões de ataques e estratégias de mitigação dos mesmos. O jogo utiliza um sistema que o alimenta em tempo real conectado ao sistema *CAPEC* (*Common Attack Pattern Enumeration Classification*) para atualizar seus conteúdos sobre ameaças e técnicas de mitigação. Foi desenvolvido um protótipo funcional para o teste de aplicação e avaliado por meio de *feedback* dos estudantes sobre a experiência, que relataram mais facilidade no aprendizado dos conceitos envolvidos por meio do jogo. Ainda que trate sobre segurança e proteção de software, o autor levanta o ponto de que a segurança de software deve ser pensada desde os primeiros estágios do ciclo de vida do software.

- **SimSE (PETRI *et al.*, 2017)**

O jogo simula processos de engenharia de software para que os alunos possam praticar os conceitos aprendidos em aula, por meio da tomada de decisões. Foi validado pelo modelo *MEEGA* (*Model for the Evaluation of Educational Games*) (WANGENHEIM & BORGATTO, 2011) de coleta de *feedback* dos estudantes participantes do teste com o jogo.

- **SCRUM'ed (PETRI *et al.*, 2017)**

Jogo focado em aprimorar o entendimento de metodologias ágeis, simulando as fases e eventos diários da metodologia ágil SCRUM. O jogo oferece um ambiente no qual o jogador percorre de maneira mais acelerada cada etapa da metodologia a fim de ter um panorama prático e amplo sobre seu funcionamento. Validado por meio do modelo *MEEGA* (WANGENHEIM & BORGATTO, 2011).

- **EAREqGame (PETRI *et al.*, 2017)**

Focado no ensino de requisitos de software, oferecendo alguns exemplos para que os alunos colem e analisem requisitos de software a fim de pôr em prática essa habilidade, visando melhor compreensão sobre como elicitar, documentar e validar os mesmos. Validado por meio do modelo *MEEGA* (WANGENHEIM & BORGATTO, 2011).

- **ThatPMGame (PETRI *et al.*, 2017)**

Focado no gerenciamento de projetos de software, o jogo busca simular situações reais enfrentadas por gerentes de projetos, exigindo tomadas de decisões estratégicas por parte dos jogadores, levando em consideração os conceitos aprendidos em aula sobre riscos, custos, gestão e gerenciamento de projetos. Validado por meio do modelo *MEEGA* (WANGENHEIM & BORGATTO, 2011).

- **Software KIDS (ROSALES *et al.*, 2016)**

O jogo se propõe a introduzir os conceitos básicos de Programação Orientada a Objetos (POO) e Engenharia de Software (ES) a crianças, e para isso o jogo foi desenvolvido para o sistema operacional *Android*, o qual o estudo afirma ser preferido aos mais jovens. O jogo tem uma estrutura simples de resolução de problemas curtos, organizados em tempos de 60 a 90 segundos, e os problemas abrangem conceitos de Classes, Objetos, Herança, Análise de Requisitos, Testes, Documentação, entre outros. Os testes com os jogos foram feitos com estudantes da Universidade Autônoma

de Zacatecas, no México, e por grupos de crianças de 6 a 12 anos participantes de programas educacionais. Os testes foram avaliados por meio de *feedback* direto dos participantes do estudo, por meio de questionários, e as respostas apontaram para um grande interesse (90%) em tecnologia, mesmo apesar das dificuldades dos níveis mais altos, também por conta do idioma diferente (o jogo estava em inglês, enquanto o idioma nativo dos participantes é o espanhol, o que levou os autores a planejarem para as próximas versões a tradução completa dos níveis do jogo).

- **X-Med (FURTADO & OLIVEIRA, 2020)**

O jogo foi desenvolvido de maneira relativamente simplista, sem ambientação sonora, interface gráfica simplificada e funcional, o que trouxe um *feedback* de não imersão do jogo. O objetivo inicial do jogo era trabalhar o processo de medição do software, pelo paradigma *Goal-Question-Metrics* (GQM), simulando as etapas do processo: coleta, armazenamento, análise e geração de relatórios sobre as métricas do próprio “X-Med”, como objeto de estudo. Durante os testes, os estudantes da disciplina utilizaram o padrão ISO 25010 para realizar a medição. O jogo foi avaliado por meio de *feedback* dos participantes, e o retorno obtido foi de que o jogo não era muito envolvente, era simplista demais, o que não ajudou o engajamento dos participantes. Os autores propuseram como alternativa o desenvolvimento de uma nova versão do jogo com mais elementos imersivos e maior qualidade para tornar a experiência mais eficaz no seu objetivo.

- **ATAM-RPG (MONTENEGRO *et al.*, 2017)**

Focado em ensinar a análise de *trade-offs* arquiteturais, no formato de um *Role-Playing Game* (RPG), onde os participantes assumem o papel dos *stakeholders* (por exemplo, os arquitetos, desenvolvedores, usuários finais, etc). O cenário é um caso realista baseado no *Sistema Nacional de Alarma de Maremoto (SNAM)* do Chile, ou o sistema de alertas contra tsunamis, onde os jogadores precisam lidar com uma situação e decidir, sem a pressão do tempo real, mas com tempo simulado, negociando entre si sobre tempo disponível

para realização do projeto como um todo, levando em conta aquilo que tem maior prioridade equilibrando o tempo necessário de implementação. A análise dos testes com a ferramenta mostraram, por meio de feedback com questionários feitos aos alunos participantes, que os jogadores sentiram maior facilidade em compreender sobre os *trade-offs* arquiteturais.

- **Classutopia (LARENAS et al., 2018)**

Um jogo no estilo RPG com o objetivo de ensinar sobre diagramação de classes, em cenários de criação, análise e correção de diagramas de classe, com diferentes níveis de complexidade. O jogador entra na pele de um robô herói que percorre diagramas de classes completando desafios e derrotando vilões, a fim de corrigir erros e interpretar diagramas para resolver problemas. O feedback coletado mostrou que os jogadores reportaram um maior nível de compreensão e retenção dos conceitos de modelagem de software, e que o jogo torna o aprendizado mais engajante.

- **Gamify-IT (SPETH et al., 2023)**

Outro macro-jogo, no estilo RPG com exploração de 5 mundos e 5 *dungeons* (masmorras comuns aos jogos de exploração e RPG), onde os jogadores interagem com *NPCs* (*Non-Playable Characters*, ou personagens não jogáveis), e realizam mini-jogos para avançar na história e no jogo. Os mini-jogos, por sua vez, são: *Bugfinder*, sobre identificação e correção de erros de código; *ChickenShock*, tiro ao alvo de perguntas e respostas, onde o jogador deve atirar na resposta correta para a pergunta de cada nível; *CrosswordPuzzle*, palavras cruzadas onde as pistas são definições de conceitos com o tema de engenharia de software; e *FiniteQuiz*, jogo simples de perguntas e respostas. Um diferencial desse jogo é que é altamente customizável e, como os jogos tem uma lógica mais simples e genérica, os conteúdos abordados podem ser adaptados a outras disciplinas e outros contextos. Os testes foram realizados com alunos do curso de introdução à engenharia de software na Universidade de Stuttgart, na Alemanha, e a avaliação foi feita por meio de *feedback* dos alunos, que avaliaram a

experiência como motivadora e divertida para o aprendizado em contraposição com as aulas regulares.

- **FormalZ (PRASETYA et al., 2019)**

Um jogo de *browser* (roda diretamente no navegador), no estilo *Tower Defense*, no qual o objetivo é proteger uma CPU de “ataques”. O foco do jogo é ensinar sobre requisitos formais e pré- e pós-condições que bloqueiam ataques, permitindo a passagem de dados corretos esperados pela CPU. Os escudos são constituídos por “blocos de código”, os quais o jogador deve conectar e utilizar corretamente para criar as condições corretas para cumprir os desafios de cada nível. O jogo foi testado em turmas de estudantes de graduação na Universidade de Utrecht, dos Países Baixos, que responderam por meio de feedback que o jogo é divertido e engajante para o ensino do conteúdo planejado. No entanto, sugeriram melhorias como uma interface gráfica melhor e expansão para outros tipos de especificações formais.

- **InspectorX 2.0 (ANDRADE et al., 2020)**

Jogo voltado ao ensino e treino de inspeção de software, encontrando e classificando defeitos em artefatos de software. O jogo simula 3 papéis para os jogadores; Inspetor, Autor e Moderador; e possui dois modos de jogo; *Full Inspection Process* onde o jogo simula todos os processos de inspeção, incluindo planejamento, descoberta de defeitos e reuniões de equipe; e *Defect Crawler*, um modo de jogador único, onde os jogadores inspecionam os artefatos e ganham pontos ao detectar defeitos. O jogo possui um *ranking* para avaliação do desempenho dos alunos, e um sistema de *chat* entre os alunos para discutirem livremente as soluções para os problemas. O jogo foi testado por alunos de graduação no Brasil, que relataram que o jogo ajudou com o aprendizado, mas a interface poderia ser mais intuitiva.

- **ProDec (CALDERÓN et al., 2017)**

Voltado a gerenciamento de projetos por meio de um ambiente simulado para o jogador treinar tomadas de decisões estratégicas durante a realização de um projeto. Os jogadores assumem o papel de gerentes de software e podem jogar dois modos distintos: *Full Play*, no qual os jogadores criam um projeto do zero e gerenciam todas as fases do ciclo de vida de um software; e *Quick Play*, no qual os jogadores resolvem problemas específicos, pontuais, previamente definidos pelo professor. O jogo se propõe a criar um ambiente seguro para tomadas de decisões nas quais os jogadores podem observar as consequências de cada decisão tomada. Foi desenvolvido e testado com alunos da Universidade de Cádiz, na Espanha, e a avaliação foi feita por meio de coleta de feedback dos alunos que demonstraram alto nível de engajamento e aprendizado, destacando como realista e envolvente a experiência. Algumas sugestões também foram trazidas como visualização de diagramas PERT (*Program Evaluation and Review Technique*) para melhorar a avaliação do projeto por parte dos jogadores e melhorias na interface gráfica.

- **ScrumVR (LÓPEZ-FERNÁNDEZ et al., 2022)**

Um jogo de Realidade Virtual para o ensino de maneira imersiva e interativa dos conceitos fundamentais da metodologia Scrum. O jogador interpreta um novo integrante de uma equipe Scrum e é guiado por um Scrum Master, passando pelas etapas de *Dailies*, *Sprint Planning*, *Sprint Review* e retrospectiva. Os jogadores também passam por quatro mini jogos e interações para aprender sobre a metodologia *Kanban*; o artefato *Burndown Charts*; a técnica do *Planning Poker*; e o método *MoSCoW* (*Must have/Should have/Could have/Won't have* - técnica de priorização de tarefas). A jogabilidade se dá por meio de interação com os *Menus* do jogo, sem necessitar de controles externos de grande complexidade. O jogo foi desenvolvido e testado na Universidade Politécnica de Madrid, na Espanha, e a avaliação do jogo foi por meio de testes sobre a metodologia, aplicados aos alunos antes e depois das sessões de jogos. Os relatos sobre o jogo em si foram positivos também, afirmando que o jogo não causou tonturas ou

estranhamento (em função do uso de realidade virtual), foi relativamente fácil de aprender e utilizar, e foi considerado envolvente e inovador pelos alunos participantes do estudo. O jogo pode ser jogado utilizando em dispositivos Android, por meio de um *Google Cardboard*, o que foi relatado como acesso de baixo custo à tecnologia.

- **PMG-2D (Project Management Game 2D) (LINO, 2015)**

Voltado ao ensino de gestão de projetos, com o objetivo de treinar os jogadores em gestão de custo, tempo, risco e gestão de pessoas. O jogador assume o papel de um gerente de projeto e deve gerir todas as fases do ciclo de vida do software. Cada membro da equipe possui uma personalidade única, baseada no modelo dos *Big Five Personality Traits* (Cinco Grandes Fatores - OCEAN - *Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism*), o que dita a dinâmica do time. O jogo simula questões reais como conflitos interpessoais, decisões sobre prazos e riscos inesperados. O jogador recebe uma avaliação no final, com feedback sobre erros e sugestões de melhorias na conduta. O jogo foi desenvolvido e testado com 35 profissionais da TI que participaram de uma avaliação prática do jogo, que relataram sucesso do jogo em motivar o aprendizado, aumentar a retenção do conhecimento e interesse em aprender mais sobre o tema. As sugestões de melhorias foram na interface gráfica e na curva de aprendizado do jogo.

- **SimuIES-W (MONSALVE, 2018)**

Um jogo baseado em web, do tipo simulação de projetos reais de ES, voltado ao ensino dos conceitos fundamentais por meio da simulação de desafios do cotidiano de projetos. Os jogadores assumem diferentes papéis, como engenheiro de software, coordenador técnico, gerente de projeto e controlador de qualidade; e devem lidar com orçamento, contratação e demissão de engenheiros, construção de artefatos de software e controle de qualidade. O jogo possui duas modalidades: Competitiva, na qual os jogadores podem lançar desafios para dificultar o progresso dos adversários;

e Colaborativa, na qual os jogadores devem argumentar e convencer seus companheiros sobre as melhores soluções para os desafios propostos. O jogo ainda pode ser customizado pelos educadores, adaptando o conteúdo a seus contextos mais específicos. O jogo foi desenvolvido completamente, e testado com alunos de graduação e pós-graduação em Engenharia de Software na PUC-Rio e na UERJ, no Brasil. A avaliação do jogo foi feita por meio de *feedback* coletado dos alunos participantes do estudo, os quais relataram que o jogo representa bem a dinâmica de um projeto de software real, consideraram o jogo interessante, motivador e educativo, e alguns professores que utilizaram o jogo em suas aulas relataram que sentiram melhoras na compreensão dos alunos sobre gestão de software ao utilizar o jogo.

- **Floors (AYDANA, 2017)**

Jogo voltado ao ensino da norma *ISO/IEC 12207* sobre padrão dos processos do ciclo de vida de software. Os jogadores interagem com *NPCs* que explicam conceitos e apresentam desafios baseados na norma. O ambiente do jogo é disposto em 9 andares virtuais 3D (por isso o nome do jogo), cada um representando um dos processos do ciclo de vida de software descrito na norma, e também dispõe de diálogos (com os *NPCs*) mais técnicos e instrumentais. O jogo foi testado com base em um grupo de 40 alunos de graduação no curso de Engenharia de Software, separados aleatoriamente em dois grupos, um permitido a jogar o jogo (*Virtual Learning* - aprendizado gamificado) e, ao outro (*Paper-Based Learning* - material impresso), foi fornecido a documentação oficial da *ISO/IEC 12207*. O desempenho foi medido por um exame escrito sobre as regras e normas, e o desempenho observado foi maior no grupo *Virtual Learning*, que considerou o jogo motivador e envolvente em comparação com materiais tradicionais de estudo e aprendizado. Os autores sugerem que o jogo pode ser utilizado complementando cursos formais sobre normas e processos de ciclo de vida de software.

- **CodeCombat (ALATRISTA-SALAS & NUNEZ-DEL-PRADO, 2018)**

Jogo no estilo RPG, com o objetivo de ensinar POO (Programação Orientada a Objetos) e conceitos básicos de desenvolvimento de software, e algumas linguagens (*Python, JavaScript, Lua e CoffeeScript*). O personagem do jogador deve escrever códigos para superar os desafios de cada nível, que vão incrementando e adicionando novos conceitos e dificuldades (variáveis, *loops*, condicionais, etc). O jogo se propõe a ensinar de forma dinâmica, o desenvolvimento de software já pensando em boas práticas de design e programação, desde o início do aprendizado. O progresso do aluno é monitorado pelos professores, que recebem relatórios individuais e coletivos sobre o desempenho daquele. Na avaliação da aplicação do jogo, foi comparado o desempenho em exame escrito de um grupo de alunos que utilizou o jogo com outro grupo que utilizou apenas materiais tradicionais. Aqueles que utilizaram o jogo tiveram um maior desempenho no teste pós-sessões e relataram que o jogo tornou o aprendizado mais divertido e envolvente, o que levou a um maior engajamento observado do que no grupo daqueles que estudaram apenas por métodos tradicionais.

- **Code Defenders (CLEGG et al., 2017)**

Jogo voltado a ensinar “*Mutation Testing*”, cobertura de código, mutação e qualidade de testes. O jogo possui três modalidades de jogo, baseadas no número de jogadores: Multijogador, no qual um time de atacantes e um time de defensores se enfrentam; Duelo 1x1, no qual há um jogador atacante e outro defensor; e *Single Player*, no qual um único jogador enfrenta um oponente controlado por IA. Os embates são definidos pelos atacantes criando mutantes do código original, enquanto os defensores devem criar testes para capturar esses mutantes. O jogo também avalia a qualidade dos testes baseando-se nos conceitos de *mutation testing*, onde são inseridos pequenos erros no código, para verificar se os testes conseguem identificá-los. Code Defenders foi desenvolvido e disponibilizado publicamente por meio do domínio “<http://code-defenders.org>”. A avaliação da experiência foi feita por meio de *feedback* dos participantes do estudo sobre a

percepção dos alunos da participação do jogo como ferramenta de aprendizado. Os relatos foram de que o jogo tornou o aprendizado de testes mais acessível de forma prática e motivador, além de ajudar a compreender a importância da qualidade de testes e das técnicas de *mutation testing*.

- **Mission 336 (OZCELIK et al., 2013)**

O jogo é voltado ao ensino de banco de dados, focado em diagramas de entidade-relacionamento (DER), no qual o jogador deve defender um centro de dados nacional com 7 portas, cada uma representando um conceito fundamental de banco de dados, como chave primária, chave secundária, atributo, cardinalidade, etc. Neste jogo, duas modalidades foram utilizadas para testar o impacto da incerteza na jogabilidade de jogos sérios; na modalidade com incerteza, há um fator de aleatoriedade que pode alterar os riscos de cada nível e também as recompensas, enquanto na outra, não há o fator (pontuações e riscos são constantes). O jogo foi testado com 140 alunos de graduação em engenharia de software, divididos entre as duas versões, e os jogadores no grupo com incertezas tiveram pontuações notadamente superiores nos testes finais sobre o conteúdo aprendido. O jogo, apesar de não tratar diretamente de engenharia de software por si, trata de banco de dados, e ainda menciona um importante ponto de engajamento que é a presença de incerteza nos jogos sérios e obteve resultados positivos de engajamento, segundo feedback dos alunos participantes do estudo.

- **SPIAL (Software Process Improvement Animated Learning Environment) (PEIXOTO et al., 2015)**

Jogo do tipo simulação, focado em melhoria do processo de software, segundo o modelo *CMMI-DEV* versão 1.3 (*Capability Maturity Model Integration For Development*), com o objetivo de trazer às salas de aula um modelo mais prático e realista do cotidiano de engenheiros de software e oferece dois caminhos: *Continuous* (definido baseado em níveis de Capabilidade) permite a seleção de áreas de processos customizadas de acordo com os objetivos da organização; *Staged* (definido baseado em níveis de Maturidade) permite a evolução de um cenário pré-definido, trazendo áreas

de processos sucessivamente incrementais à medida que o jogo avança. O jogador entra no papel de um gerente de um grupo de SPI (*Software Process Improvement*) e deve interagir com os *NPCs*, que representam cada um dos *stakeholders* do projeto, e cumprir desafios propostos por eles para avançar no jogo, cumprindo tarefas de melhorias de processos de software: melhorar custo, remover e prevenir defeitos, reduzir riscos, etc. Ao final, o jogador recebe uma pontuação baseada em quão próximo a solução do jogador chegou perto da solução proposta pelo criador do cenário. A avaliação do jogo foi feita com base em um grupo piloto para determinar pontos fortes e fracos do *SPIAL*, e depois com um segundo grupo, ambos do curso de graduação da UFMG, no Brasil. A avaliação seguiu o *framework* de avaliação de jogos *UGALCO* (PEIXOTO *et al.*, 2014) (questionário graduado com notas entre 0 e 1; no qual 0.0 é um negativo forte, 0.25 é um negativo fraco, 0.5 é neutro, 0.75 positivo fraco e 1.0 é positivo forte, as questões são macro-organizadas em Experiência de Jogo, Adaptatividade, Experiência de Aprendizado e Usabilidade). Os relatos dos estudantes são positivos e consideraram o jogo divertido, com nível de aprofundamento adequado e fácil de jogar. No entanto, o feedback também relatou o jogo como mais eficaz em fixar e reforçar conceitos do que introduzir novos conhecimentos

## 5.2 - Quais áreas da Engenharia de Software são apoiadas pelos jogos?

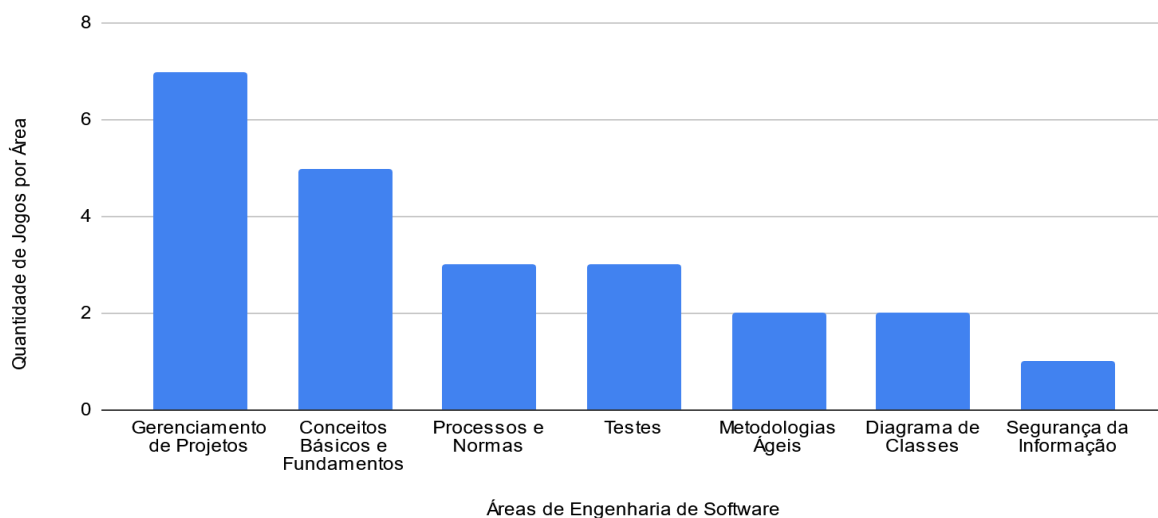


Figura 1 - Distribuição de jogos de acordo com a área de ES apoiada (Fonte: autoria própria)

As áreas apoiadas pelos jogos estão distribuídas como mostra o gráfico da “Figura 1”. Nesta revisão, notamos claramente uma concentração nos temas de “Conceitos Básicos e Fundamentos” e “Gerenciamento de Projetos”.

**a) Conceitos Básicos e Fundamentos:**

Jogos que abordam os fundamentos e conceitos básicos de engenharia de software, introdução de conceitos fundamentais. “*Essence Kernel Puzzler*” (PIEPER et al., 2017) e “*Essence Simulation Game*” (PIEPER et al., 2017), são dois jogos que abordam o *Essence Kernel*, da SEMAT. “Jogo educacional criado com a plataforma SGAME” (GORDILLO et al., 2023) aborda conceitos iniciais básicos da engenharia de software, tais como modularidade, acoplamento, coesão, abstração e ocultação de informações. “*Software KIDS*” (ROSALES et al., 2016) aborda tanto conceitos básicos de POO quanto de ES, introduzindo-os de maneira incremental, tais como Classes, objetos, atributos, Análise de problemas, requisitos, design, implementação, testes, etc. “*CodeCombat*” (ALATRISTA-SALAS & NUNEZ-DEL-PRADO, 2018) se propõe a tratar desde o ensino de linguagens do paradigma POO, mas voltado a ensinar boas práticas e códigos estruturados desde o início do ensino.

**b) Processos E Normas:**

Jogos voltados ao ensino e treinamento de processos e normas formais definidos dentro da Engenharia de Software. “*X-Med*” (FURTADO & OLIVEIRA, 2020) trata de processos de medição de software baseada no paradigma GQM (*Goal-Question-Metrics*). “*Floors*” (AYDANA, 2017) trata do ensino da *ISO/IEC 12207*, sobre padrões do ciclo de vida de software. “*SPIAL (Software Process Improvement Animated Learning Environment)*” (PEIXOTO et al., 2015) trata de SPI (*Software Process Improvement*), trazendo caminhos possíveis para melhoria do processo de desenvolvimento, análise e evolução de software, a partir do zero ou a partir de um cenário pré-definido.

**c) Metodologias Ágeis:**

Jogos que introduzem e oferecem treino nas metodologias ágeis, como *Scrum*, *Kanban*, etc. “*SCRUM’ed*” (PETRI *et al.*, 2017) trata das fases e eventos da metodologia *Scrum* como *dailies* e *sprints*. “*ScrumVR*” (LÓPEZ-FERNÁNDEZ *et al.*, 2022) também trata dos conceitos básicos da metodologia *Scrum*, colocando o jogador no papel de um iniciante na metodologia guiado por um *NPC Scrum Master*.

**d) Requisitos:**

Jogos voltados ao ensino de requisitos formais, desde coleta e análise até sua validação. “*EAREqGame*” (PETRI *et al.*, 2017) traz um projeto para os jogadores coletarem, especificarem e validarem seus requisitos. “*FormalZ*” (PRASETYA *et al.*, 2019) põe em prática a capacidade dos jogadores de elaborar e elicitar pré e pós-condições que servirão de escudo para filtrar dados incorretos.

**e) Gerenciamento de Projetos:**

Jogos que simulam problemas e situações do cotidiano de um engenheiro de software, trazendo um ambiente mais prático ao ensino de ES. “*SimSE*” (PETRI *et al.*, 2017) simula ações e decisões que um gerente de projeto deve tomar durante o ciclo de vida de software e as consequências das decisões. “*ThatPMGame*” (PETRI *et al.*, 2017), assim como “*SimSE*” segue a mesma analogia. “*ATAM-RPG*” (MONTENEGRO *et al.*, 2017) foca nos *trade offs* arquiteturais, utilizando como inspiração um sistema real. “*ProDec*” (CALDERÓN *et al.*, 2017) oferece opções para o jogador analisar um projeto inteiro ou situações pontuais. “*PMG-2D (Project Management Game 2D)*” (LINO, 2015) se concentra em treinar engenheiros de software inexperientes nas áreas de custo, tempo, risco e gestão de pessoas. “*SimULES-W*” (MONSALVE, 2018) traz como diferencial a simulação de um ambiente colaborativo onde as tomadas de decisões são tomadas em conjunto.

**f) Diagramas de Classes:**

Jogos voltados ao ensino e fixação de diagramas de classes. “*Classutopia*” (LARENAS *et al.*, 2018) possui níveis de dificuldades que aumentam de acordo com a progressão do jogador, e permite uma visão dinâmica de diagramas, bem como, de maneira interativa, o leva a navegar por um diagrama. “*Mission 336*” (OZCELIK *et al.*, 2013) Também trata de diagramas de classes, mas no contexto de banco de dados, ajudando a fixar os conceitos.

**g) Testes:**

Jogos voltados ao ensino e treino de metodologias, técnicas e conceitos de testes e garantia de qualidade. “*Gamify-IT*” (SPETH *et al.*, 2023) possui quatro mini jogos voltados a conceitos básicos e identificação de erros. “*InspectorX 2.0*” (ANDRADE *et al.*, 2020) trata especificamente sobre inspeção de software, no qual os jogadores aumentam sua pontuação baseado nos defeitos e erros que encontram. “*Code Defenders*” (CLEGG *et al.*, 2017) trata de cobertura de código, qualidade de testes e técnicas de *mutation testing*.

**h) Segurança de Informação:**

Ainda que não seja especificamente um tema contido dentro da Engenharia de Software, “*Data-Driven Security Game (DdSG)*” (LØVGREEN *et al.*, 2019) insere o pensamento da segurança da informação no contexto do ensino de planejamento e desenvolvimento de softwares como parte inerente ao processo.

**5.3 - De quais formas foram avaliados os jogos e a aplicação do jogo no ensino?**

Nos estudos levantados nesta revisão, em grande parte deles, o jogo é aplicado no início do semestre como material de apoio ou após uma breve explicação sobre seu funcionamento, e é disponibilizado aos estudantes durante todo o semestre, sendo a avaliação do jogo feita utilizando ferramentas e técnicas de coleta de *feedback* direto dos aprendizes participantes dos estudos e testes com

jogos eletrônicos. Entre os modelos de avaliação, foram citados *MEEGA/MEEGA+* (PETRI *et al.*, 2017), *UGALCO Framework* (PEIXOTO *et al.*, 2014), como modelos padronizados de questionários de *feedback* sobre jogos sérios. (LINO, 2015; CLEGG *et al.*, 2017), foram utilizadas métricas coletadas pelo próprio jogo (“*PMG-2D*” e “*Code Defenders*”, respectivamente) para avaliar o desempenho dos alunos. Em (AYDANA, 2017); (ALATRISTA-SALAS & NUNEZ-DEL-PRADO, 2018; OZCELIK *et al.*, 2013), foi comparado o desempenho em testes teóricos daqueles que utilizaram apenas o material tradicional do grupo que utilizou os jogos mencionados (“*Floors*”, “*Code Combat*” e “*Mission 336*”, respectivamente) como material de apoio. Em (PEIXOTO *et al.*, 2015), houveram dois momentos de teste do jogo: no estudo piloto, um questionário sobre o conteúdo a ser ministrado foi aplicado antes e outro após a sessão de jogo, e os resultados comparados para medir o impacto do jogo como ferramenta de ensino; no segundo estudo, foi utilizado o *framework UGALCO* (PEIXOTO *et al.*, 2014). No restante dos estudos, o jogo foi avaliado por coleta de relatos ou questionários personalizados sobre a percepção dos alunos sobre o jogo e a experiência de aprender por meio do jogo, a qual foi positiva, relatando como engajadora e divertida a maioria dos jogos como metodologia alternativa de ensino. Os jogos mencionados em (PETRI *et al.*, 2017) foram todos avaliados utilizando o modelo padronizado MEEGA+. Ainda, as sugestões de melhorias dos jogos por parte dos participantes dos estudos se concentram em melhorias gráficas, mais níveis de jogo, melhorias na curva de dificuldade e integração de comunicação dentro do jogo.

No gráfico da “Figura 2” a seguir, podemos verificar a proporção de estudos por modalidade de avaliação do impacto do jogo no processo de ensino e aprendizado.

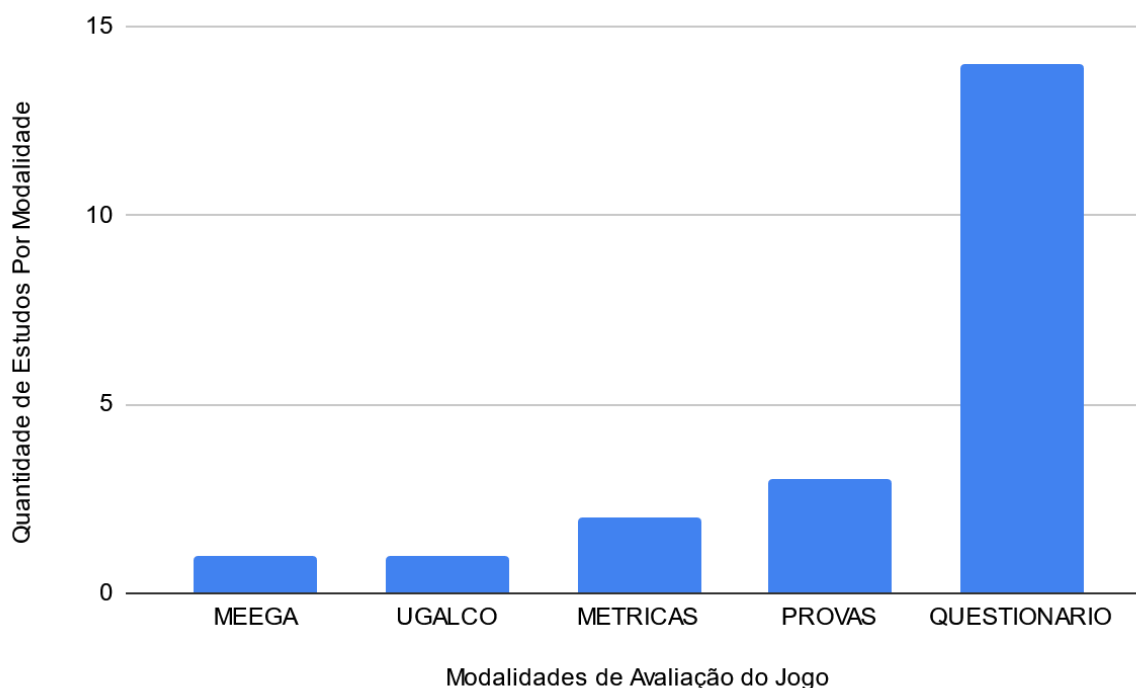


Figura 2: Proporção de estudos por cada modalidade de avaliação da aplicação do jogo no processo de ensino. (Fonte: autoria própria)

#### 5.4 - Discussões finais

No contexto dos jogos eletrônicos utilizados para o ensino de engenharia de software levantados nesta revisão, foi entendido como principal motivação pela busca de metodologias alternativas à tradicional; baseada em aulas expositivas, seminários em grupo ou individuais, exames teóricos e material escrito; a percepção por parte dos educadores de que o engajamento nas aulas de ES é baixo e dificultoso. A justificativa mais recorrente para essa percepção é de que o ensino de ES, principalmente no início da formação acadêmica dos alunos, os quais não tem vasta experiência em cargos de engenheiros de software, torna-se muito teórico e abstrato.

Como menciona Peixoto *et al.* (2015), a própria indústria percebe como ineficaz a formação do engenheiro de software. Também no mesmo estudo, a metodologia do uso de jogos é sugerida como complementar à tradicional, ou seja, não substitui o paradigma tradicional de aulas expositivas, material didático, seminários em grupo, etc.

Quanto ao engajamento, todos os estudos levantados tiveram *feedback* positivo por parte dos alunos participantes dos testes em maioria estatística (no

mínimo os relatos foram de 60% a 70% de avaliações positivas da experiência), e no geral descrevem o uso de jogos, enquanto auxílio ao material tradicional de ensino, como motivadoras e divertidas. Também o uso de “incerteza” (OZCELIK *et al.*, 2013) e “competição” (ALATRISTA-SALAS & NUNEZ-DEL-PRADO, 2018) e (FURTADO & OLIVEIRA, 2020), se mostraram eficientes como engajadores e reforçadores positivos.

Também é recorrente nos estudos a conclusão de que o jogo educacional deve equilibrar *user acceptance* e *teaching efficiency/efficacy*. Ou seja, um jogo educacional eficaz deve ser ao mesmo tempo divertido e engajador, e relevante ao ensino. Também é recorrente a opinião de que os estudantes e aprendizes são favoráveis a metodologias de ensino diferentes das tradicionais, como a gamificação ou o modelo híbrido de vídeo-aulas e jogos (digitais ou físicos).

No contexto dos jogos eletrônicos para o ensino de engenharia de software, apenas alguns poucos jogos mencionados foram encontrados disponíveis, uma vez que a maioria dos jogos são desenvolvidos no formato de protótipos a serem testados e aplicados em estudo, e nem sempre há divulgação e manutenção dos mesmos disponíveis. Os que foram encontrados estavam disponíveis em repositórios como github ou domínios particulares.

## **6 - Ameaças À Validade**

Durante o planejamento da RSL, foram identificados algumas possíveis ameaças à validade da pesquisa:

- Viés causado pela limitação das bases: quantidade de bases de dados.
- Viés causado por procedimentos manuais: a possibilidade de ter excluído ou não ter encontrado algum estudo relevante.
- Viés dos critérios de seleção e exclusão: seleção de estudos que seguissem todos os critérios (seleção de estudos primários disponíveis pelo acesso CAFe).

A utilização da Rede Nacional de Ensino e Pesquisa (RNP) possibilitou o acesso a estudos não gratuitos por meio da Comunidade Acadêmica Federada (CAFe), a qual a Universidade Federal de São Carlos (UFSCar) participa, no Portal de Periódicos CAPES. Inicialmente foram determinadas três bases reconhecidas na área de Tecnologia, no entanto uma delas teve seu acesso perdido pelo acesso

CAFe durante a execução da pesquisa e não foi possível recuperar os estudos encontrados nela. Em virtude dos prazos, se tornou inviável voltar e reiniciar a pesquisa selecionando outras bases.

## **7 - Conclusão**

Esta revisão foi motivada pela ideia de identificar e analisar o uso de jogos eletrônicos no ensino de engenharia de software enquanto metodologia alternativa à tradicional, identificando os jogos utilizados, a motivação de seu uso, como são avaliados e medidos os impactos e a eficiência da aplicação dos mesmos.

A revisão identificou como principais formas de avaliação o feedback dos participantes do estudo, muitas vezes de maneira subjetiva, outras por meio de ferramentas de coleta de feedback, como *MEEGA* (WANGENHEIM & BORGATTO, 2011) ou *UGALCO* (PEIXOTO *et al.*, 2014), e que os próprios testes são feitos, normalmente, de maneira limitada e pontuais, visto que utilizam como base de coleta de dados grupos de alunos pequenos (de 12 a 60) em uma única turma ou semestre, e a maioria dos protótipos são aplicados em estudos apenas uma única vez.

Baseados nas análises dos estudos levantados, pode-se concluir que o uso de jogos eletrônicos traz benefícios, principalmente quando aliados às metodologias tradicionais de ensino ao invés de substituí-las. Como trabalhos futuros, se poderia expandir a busca a outras bases de dados, comparar as conclusões levantadas e verificar se o mesmo ocorre, podendo servir de base para uma nova proposta de ensino multimídia, apoiadas também por outras áreas da ciência como a psicologia e a educação especial, caso o mesmo possa ser feito levando em conta jogos que cubram necessidades especiais.

## REFERÊNCIAS

ALATRISTA-SALAS, H.; NUNEZ-DEL-PRADO, M. Teaching Software Engineering through Computer Games. WORLD ENGINEERING EDUCATION CONFERENCE (EDUNINE), 2018, Buenos Aires, Argentina. IEEE, 2018, p. 1-4. DOI: 10.1109/EDUNINE.2018.8450996.

ANDRADE, L.; GRYNBERG E.; SCHOTS, M.; WERNECK, V. M. B. InspectorX 2.0: Developing a Multi-Device Game for Software Inspection Education. In: CONFERENCE ON SOFTWARE ENGINEERING EDUCATION AND TRAINING (CSEET), 32, 2020, Munich, Germany. IEEE, 2020, p. 1-4. DOI: 10.1109/CSEET49119.2020.9206210.

AYDANA, U.; YILMAZ, M.; CLARKE, P. M.; O'CONNOR, R. V. Teaching ISO/IEC 12207 software lifecycle processes: A serious game approach. Computer Standards & Interfaces, v. 54, n. 3, p. 129-138, nov. 2017. DOI: 10.1016/j.csi.2016.11.014.

BORGES, G. G.; SOUZA, R. C. G. Skills Development For Software Engineers: Systematic Literature Review. INFORMATION AND SOFTWARE TECHNOLOGY. [s. l.]. v. 168, abr. 2024. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0950584923002501>. Acesso em: 07/02/2025. DOI: 10.1016/j.infsof.2023.107395.

CALDERÓN, A.; RUIZ, M.; ORTA, E. Integrating Serious Games as Learning Resources in a Software Project Management Course: the Case of ProDec. In: INTERNATIONAL WORKSHOP ON SOFTWARE ENGINEERING CURRICULA FOR MILLENNIALS (SECM), 1, 2017, Buenos Aires, Argentina. IEEE, 2017, p. 21-27. DOI: 10.1109/SECM.2017.3.

CLEGG, B. S.; ROJAS, J. M.; FRASER, G. Teaching Software Testing Concepts Using a Mutation Testing Game. INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING: SOFTWARE ENGINEERING EDUCATION AND TRAINING TRACK (ICSE-SEET), 39, 2017, Buenos Aires, Argentina. IEEE, 2017, p. 33-36. DOI: 10.1109/ICSE-SEET.2017.1.

FURTADO, L. S.; OLIVEIRA, S. R. B. A teaching proposal for the software measurement process using gamification: an experimental study. In: IEEE FRONTIERS IN EDUCATION CONFERENCE (FIE), 50, 2020, Uppsala, Sweden, IEEE, 2020, p. 1-8. DOI: 10.1109/FIE44824.2020.9274194.

GORDILLO, A.; BARRA, E.; QUEMADA, J. SGAME: An Authoring Tool to Easily Create Educational Video Games by Integrating SCORM-Compliant Learning Objects. *IEEE Access*, v. 9, p. 126414-126430, 2021. DOI: 10.1109/ACCESS.2021.3111513.

GORDILLO, A.; LÓPEZ-FERNÁNDEZ, D.; TOVAR, E.; MAYOR, J. A Comparison of the Usefulness of Game-Based Learning and Video-Based Learning for Teaching Software Engineering in Online Environments. In: FRONTIERS IN EDUCATION CONFERENCE (FIE), 53, 2023, College Station, Texas, Estados Unidos da America. IEEE, 2023, p. 1-7. DOI: 10.1109/FIE58773.2023.10343449

KHARBOUCH, M., et al. Uncharted Dimensions, Gaps, And Future Trends Of Serious Games In Software Engineering. *COMPUTER STANDARDS & INTERFACES*. vol. 92, mar. 2025. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0920548924000849>. Acesso em: 07/02/2025. DOI: 10.1016/j.csi.2024.103915

LARENAS F.; MARÍN B.; GIACHETTI G. Classutopia: A Serious Game for Conceptual Modeling Design. In: *INTERNATIONAL JOURNAL OF SOFTWARE ENGINEERING AND KNOWLEDGE ENGINEERING*, 28:11n12 (edição especial), 2018, Santiago, Chile. World Scientific Publishing Company, 2018, p. 1679-1699. DOI: 10.1142/S0218194018400235.

LINO, J. E. N. et al. Project Management Game 2D (PMG-2D), A Serious Game to Assist Software Project Managers Training. *FRONTIERS IN EDUCATION CONFERENCE (FIE)*, 2015, El Paso, TX, USA. IEEE, 2015, p. 1-8. DOI: 10.1109/FIE.2015.7344168.

LÓPEZ-FERNÁNDEZ, D.; MAYOR, J.; PÉREZ, J.; GORDILLO, A. Learning and Motivational Impact of Using a Virtual Reality Serious Video Game to Learn Scrum. *IEEE Transactions on Games*, v. 15, n. 3, p. 430-439, Outubro, 2022. DOI: 10.1109/TG.2022.3213127.

LØVGREN, D. E. H.; LI, J.; OYETOYAN, T. D. A Data-Driven Security Game to Facilitate Information Security Education. In: *INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING: COMPANION PROCEEDINGS (ICSE-COMPANION)*, 41, 2019, Montreal, QC, Canada. IEEE, 2019, p. 256-257, DOI: 0.1109/ICSE-Companion.2019.00102.

MONSALVE, E. S. et al. SimulES-W: A Collaborative Game to Improve Software Engineering Teaching. *Comp. y Sist, Ciudad de México*, v. 22, n. 3, p. 953-983, jul./set. 2018. DOI: 10.13053/CyS-22-3-2711.

MONTENEGRO, C. H.; ASTUDILLO, H.; ÁLVAREZ, M. C. G. ATAM-RPG: A role-playing game to teach Architecture Trade-off Analysis Method (ATAM). In: *LATIN AMERICAN COMPUTER CONFERENCE (CLEI)*, 43, 2017, Cordoba, Argentina. IEEE, 2017, p. 1-9. DOI: 10.1109/CLEI.2017.8226416.

OZCELIK, E.; CAGILTAY, N. E., OZCELIK, N. S. The effect of uncertainty on learning in game-like environments. *Computers & Education*, v. 62, p. 12-20, set. 2013. DOI: 10.1016/j.compedu.2013.02.009.

PEIXOTO, D. C. C.; RESENDE, R. F.; PÁDUA, C. I. P. S. Evaluating Software Engineering Simulation Games: The UGALCO Framework. In: *ASEE/IEEE FRONTIERS IN EDUCATION CONFERENCE*, 44, p. 1-9, Madrid, Spain, 22-25 October, 2014. Disponível em: [https://homepages.dcc.ufmg.br/~cascini/TRFIE\\_14.pdf](https://homepages.dcc.ufmg.br/~cascini/TRFIE_14.pdf). Acesso em: 09/02/2025. DOI: 10.1109/FIE.2014.7044204.

PEIXOTO, D. C. C.; RESENDE, R. F.; PÁDUA, C. I. P. S. The Issues of Adopting Simulation Games in Software Engineering Classes. *FRONTIERS IN EDUCATION*

CONFERENCE (FIE), 2015, El Paso, TX, USA. IEEE, p. 1-8. DOI: 10.1109/FIE.2015.7344071.

PETRI, G.; WANGENHEIM, C. G.; BORGATTO, A. F. A Large-Scale Evaluation of a Model for the Evaluation of Games for Teaching Software Engineering. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING: SOFTWARE ENGINEERING EDUCATION AND TRAINING TRACK (ICSE-SEET), 39, 2017, Buenos Aires, Argentina. IEEE, 2017, p. 180-189, DOI: 10.1109/ICSE-SEET.2017.11.

PIEPER, J.; LUETH, O.; GOEDICKE, M.; FORBRIG, P. A case study of software engineering methods education supported by digital game-based learning: Applying the SEMAT Essence kernel in games and course projects. In: EDUCATION CONFERENCE (EDUCON), 8, 2017, Atenas - Grécia. IEEE, 2017, p. 1689-1699. DOI: 10.1109/EDUCON.2017.7943076

PRASETYA, W. et al. Having Fun in Learning Formal Specifications. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING: SOFTWARE ENGINEERING EDUCATION AND TRAINING (ICSE-SEET), 41, 2019, Montreal, QC, Canada. IEEE, 2019, p. 192-196. DOI: 10.1109/ICSE-SEET.2019.00028.

RAMMSTEDT, B.; JOHN, O. P. Measuring personality in one minute or less: A 10-item short version of the Big Five Inventory in English and German. *Journal of Research in Personality*, v. 41, n. 1, p. 203-212, fev. 2007. DOI: <https://doi.org/10.1016/j.jrp.2006.02.001>. Acesso em: 26/02/2025.

ROSALES, S. R.; REYES, S. V.; CISNEROS, J. L. V.; SIGG M. L. A serious game to promote object oriented programming and software engineering basic concepts learning. In: INTERNATIONAL CONFERENCE IN SOFTWARE ENGINEERING RESEARCH AND INNOVATION (CONISOFT), 4, 2016, Puebla, México. IEEE, p. 97-103, DOI: 10.1109/CONISOFT.2016.23.

SPETH S.; HOFMEISTER, L.; BECKER, S.; BREITENBÜCHER, U. Gamify-IT: A Web-Based Gaming Platform for Software Engineering Education. In: INTERNATIONAL WORKSHOP ON SOFTWARE ENGINEERING EDUCATION FOR

THE NEXT GENERATION (SEENG), 5, 2023, Melbourne, Australia. IEEE, 2023, p. 25-28. DOI: 10.1109/SEENG59157.2023.00009.

TUMPA, R. J.; AHMAD, T.; NAENI, L. M.; KUJALA, J. Computer-based Games in Project Management Education: A Review. PROJECT LEADERSHIP AND SOCIETY, [s. l.], 5, dez. 2024, disponível em: <https://www.sciencedirect.com/science/article/pii/S2666721524000152?via%3Dihub>. Acesso em: 07/02/2025. DOI: 10.1016/j.plas.2024.100130.

WANGENHEIM, C. G.; BORGATTO, A. F. A Model for the Evaluation of Educational Games for Teaching Software Engineering. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 25, 2011, São Paulo, Brasil, set. 2011, p. 194-203. DOI: 10.1109/SBES.2011.27. Acesso em 09/02/2025.

## **APÉNDICE A:** Protocolo de RSL do Parsif.al

### **TCC - Um estudo sobre jogos eletrônicos para o ensino de engenharia de software**

Gustavo Bastos, Delano Beder, Joice Otsuka

#### **PLANEJAMENTO**

##### **Descrição**

Encontrar quais jogos digitais utilizados para auxiliar o ensino de engenharia de software, levantar estudos secundários sobre o tema, e compreender como é feita a avaliação do jogo e quais as etapas/fases da engenharia de software são apoiadas pelos jogos.

##### **PICOC**

- **Population:**
- **Intervention:**
- **Comparison:**
- **Outcome:**
- **Context:**

##### **Questões de Pesquisa**

1. Quais jogos são utilizados para auxiliar o aprendizado sobre Engenharia de Software?
2. Qual a forma de avaliação do aprendizado?
3. Quais etapas/fases/estágios da engenharia de Software são treinadas por meio de jogos?
4. Qual a forma de avaliação dos jogos?

##### **Palavras Chaves e Sinônimos**

Keyword

Synonyms

Learning

Serious Games

Software Engineering

Teaching

### **String de Busca**

(learning OR teaching) AND "software engineering" AND games

### **Fontes de Dados**

- ACM Digital Library (<https://dl.acm.org/>)
- IEEE Xplore (<https://ieeexplore.ieee.org/Xplore/home.jsp>)
- Scopus Elsevier (<https://www.scopus.com/home.uri>)

### **Critérios de Seleção**

#### **Critérios de Inclusão:**

- Trata de métodos de avaliação de jogos para o ensino de ES
- Trata de um jogo para ensino de Eng. de Software

#### **Critérios de Exclusão:**

- Apenas gamificação
- Jogos não digitais
- Não disponível publicamente
- Não trata de Eng. de Software
- Não trata de Jogos
- Trabalhos acadêmicos não pertinentes (proceedings, etc)
- Trata apenas do desenvolvimento de jogos
- Versão mais recente do trabalho incluída

### **Checklist de Verificação da Qualidade**

#### **Perguntas:**

- Menciona um ou mais jogos utilizados para ensinar Engenharia de Software?
- Realiza avaliação do processo (aplicação do jogo no ensino)?

- O(s) jogo(s) mencionado(s) estão desenvolvidos e disponíveis?

### **Respostas:**

- Sim
- Parcialmente
- Não

### **Formulário de Extração de Dados**

- Qual o jogo ou os jogos mencionados?
- Jogos disponíveis para uso/teste?
- Modo de avaliação do jogo?
- Quais áreas da Engenharia de Software são exploradas pelo(s) jogo(s)?
- Modo de avaliação do aprendizado?

### **CONDUÇÃO**

#### **Strings de Buscas das Bibliotecas Digitais**

##### **ACM Digital Library:**

Title:("software engineering" AND game\* AND (teach\* OR learn\*)) OR

Abstract:("software engineering" AND game\* AND (teach\* OR learn\*)) OR

Keyword:("software engineering" AND game\* AND (teach\* OR learn\*))

##### **IEEE Xplore:**

("Document Title": "software engineering" AND "Document Title": "gam\*") AND

("Document Title": "teach\*" OR "Document Title": "learn\*") AND

("Abstract": "software engineering" AND "Abstract": "gam\*") AND ("Abstract": "teach\*" OR "Abstract": "learn\*")

##### **Scopus Elsevier:**

Advanced query

```
TITLE-ABS-KEY ( {software engineering} AND game AND ( teach OR learn ) ) AND  
PUBYEAR > 2012 AND PUBYEAR < 2024 AND ( LIMIT-TO ( LANGUAGE , "English"  
) OR LIMIT-TO ( LANGUAGE , "Portuguese" ) )
```

### **Estudos Importados**

- **ACM Digital Library:** 0
- **IEEE Xplore:** 18
- **Scopus Elsevier:** 200