

UNIVERSIDADE FEDERAL DO MARANHÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

SEBASTIÃO HENRIQUE NASCIMENTO SANTOS

DESENVOLVIMENTO DE UM JOGO EDUCACIONAL PARA O ENSINO DE
RISCOS E MITIGAÇÕES EM ENGENHARIA DE SOFTWARE

São Luís

2019

SEBASTIÃO HENRIQUE NASCIMENTO SANTOS

**DESENVOLVIMENTO DE UM JOGO EDUCACIONAL PARA O ENSINO DE
RISCOS E MITIGAÇÕES EM ENGENHARIA DE SOFTWARE**

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Luis Jorge Enrique Rivero Cabrejos

São Luís

2019

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Núcleo Integrado de Bibliotecas/UFMA

Nascimento Santos, Sebastião Henrique.

Desenvolvimento de um Jogo Educacional para o Ensino de Riscos e Mitigações em Engenharia De Software / Sebastião Henrique Nascimento Santos. - 2019.

52 f.

Orientador(a): Luis Jorge Enrique Rivero Cabrejos.

Monografia (Graduação) - Curso de Ciência da Computação, Universidade Federal do Maranhão, São Luís, 2019.

1. Focus Group. 2. Gerenciamento de Risco. 3. Jogos Educacionais. 4. MEEGA+. I. Rivero Cabrejos, Luis Jorge Enrique. II. Título.

SEBASTIÃO HENRIQUE NASCIMENTO SANTOS

**DESENVOLVIMENTO DE UM JOGO EDUCACIONAL PARA O ENSINO DE
RISCOS E MITIGAÇÕES EM ENGENHARIA DE SOFTWARE**

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Luis Jorge Enrique Rivero Cabrejos

Aprovado em São Luís, 18 de dezembro de 2019.

Prof. Dr. Luis Jorge Enrique Rivero Cabrejos
Orientador
Universidade Federal do Maranhão (UFMA)

Prof. Dr. Anselmo Cardoso de Paiva
Examinador
Universidade Federal do Maranhão (UFMA)

Prof. Dr. Davi Viana dos Santos
Examinador
Universidade Federal do Maranhão (UFMA)

São Luís
2019

Agradecimentos

Agradeço aos meus orientadores de pesquisa Luis Rivero e Anselmo Paiva por terem me acolhido e orientado. Sou eternamente grato pelo entusiasmo em ensinar, motivar e, principalmente, ser uma inspiração como profissional e pesquisador.

Agradeço aos professores Davi Viana e Anselmo Paiva por compor a banca examinadora desse trabalho.

Aos meus companheiros de curso Arthur Serra, Eduardo Vieira, Layane Menezes e Pedro Carvalho, por primeiramente serem as melhores amizades, além de sempre estarem presente nos momentos de estudo, diversão e também nas dificuldades vividas.

Aos companheiros de pesquisa e laboratório, Yandson Costa, Flávia Carvalho, Jéssica Mendes e Kastney Frazão, que ajudaram na realização desse trabalho e estavam presentes nos mais diversos momentos. Ainda, aos meus antigos companheiros de laboratório Nélia Reis, Polyana Costa, Christyellen Lima e Pedro Vinicius Almeida por terem me acolhido no início dessa jornada. Ao amigo de longa data, Anis Settimi, pela companhia constante e as revisões realizadas. Também aos demais amigos que estiveram comigo durante essa trajetória.

Agradeço principalmente aos meus pais, sem eles nada disso seria possível. Seus cuidados, carinho e atenção foram fundamentais durante essa etapa. Sou extremamente grato pelo intenso incentivo para me aperfeiçoar e as condições que me deram para percorrer esse caminho.

RESUMO

O ensino de Engenharia de *Software* tradicionalmente é realizado a partir da apresentação de conceitos em nível formal com aulas expositivas. Embora essencial somente o conhecimento teórico não é suficiente para que o aluno esteja preparado para a tomada de decisões apresentadas no mercado. Assim é necessário capacitar os alunos com conteúdo teórico e prático, de forma a evoluir da educação tradicional para abordagens dinâmicas que envolvam aspectos de projetos reais. Jogos sérios ou jogos educacionais estão se tornando ferramentas cada vez mais utilizadas para estimular o aprendizado dos aspectos relacionados a engenharia de software. No entanto, poucos são os jogos aplicados em sala de aula, pois não há informações sintetizadas sobre os mesmos, impedindo que sejam adquiridos e replicados. Dessa forma, foi desenvolvido o jogo Arriscando, um jogo de cartas que motiva o aluno a relacionar riscos de projetos de *software* com possíveis soluções. Para o desenvolvimento do jogo foi necessário identificar o estado da arte dos jogos educacionais desenvolvidos para o ensino da Engenharia de *Software*. Para obter evidências empíricas sobre o aprendizado e a experiência evocados pelo jogo, o mesmo foi utilizado em três aulas de Engenharia de *Software*. Para avaliar o jogo, foi aplicado o questionário *MEEGA+* e uma dinâmica de *Focus Group*. Os resultados sugerem que o jogo foi bem aceito e que os alunos aprovaram a dinâmica utilizada, a interação social e o aprendizado por meio de uma atividade divertida.

Palavras-Chave: Jogos Educacionais. Gerenciamento de Riscos. *MEEGA+*. *Focus Group*.

Abstract

Teaching in Software Engineering is traditionally based on the presentation of concepts at the formal level with lectures. Theoretical knowledge is essential but only it is not enough for the student to be prepared to make decisions in the market. Thus, it is necessary to train students with theoretical and practical content in order to evolve traditional education to dynamic approaches that involve aspects of real projects. Serious games or educational games are becoming increasingly used tools to stimulate the learning of aspects related to Software Engineering. However, there are few games used in the classroom, as there is no synthesized information about them, avoiding them from being replicated. Thus, the game Risking was developed, a card game that motivate the student to relate risks of software projects with possible solutions. For the development of the game, it was necessary to identify the state of the art of educational games developed for teaching Software Engineering. To obtain empirical information about the learning and experience evoked by the game, it was used in three Software Engineering classes. To evaluate the game, the MEEGA+ questionnaire and a Focus Group dynamic was applied. The suggested results about the game were well accepted and the students approved about the use, social interaction and learning through a fun activity.

Keywords: Educational Game. Risk Management. MEEGA+. Focus Group.

Lista De Ilustrações

Figura 1: Processo do Design Thinking d.school.....	15
Figura 2: Elementos do jogo Arriscando.....	33
Figura 3: Exemplo de processo de ataque e defesa do ponto de vista do atacante e a defesa..	35
Figura 4: Alunos jogando o Arriscando em sala de aula.....	39
Figura 5: Resultados do MEEGA+ Experiência do Jogador.....	42
Figura 6: Resultados do MEEGA+ usabilidade.	43

Lista de Tabelas

Tabela 1. Categorias de riscos, com exemplos de riscos e suas respectivas mitigações.	17
Tabela 2: Strings de busca utilizadas na mapeamento sistemático da literatura.	20
Tabela 3: Bases utilizadas no <i>mapeamento</i> sistemático da literatura.	21
Tabela 4: Bases quanto as suas características de busca.	22
Tabela 5: Formulário de extração de dados.	23
Tabela 6: Sumarização do processo de busca e inclusão dos trabalhos.	25
Tabela 7: Jogos identificados no mapeamento sistemático.	27

Lista de Abreviaturas

ES	Engenharia de Software
MSL	Mapeamento Sistemático da Literatura
JE	Jogos Educacionais

SUMÁRIO

1	INTRODUÇÃO	11
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	Jogos educacionais	14
2.2	<i>Design thinking</i>	15
2.3	Gerenciamento de riscos	16
2.4	Jogos educacionais para gerenciamento de riscos	18
3	MAPEAMENTO SISTEMÁTICO DA LITERATURA	19
3.1	Questões de pesquisa	19
3.2	<i>String</i> de busca.....	20
3.3	Veículos de publicação e bases digitais	21
3.4	Critérios de inclusão e exclusão	22
3.5	Estratégia para extração de dados	22
3.6	Resultados e discussão	24
4	ARRISCANDO: UM JOGO PARA O ENSINO DE GERENCIAMENTO DE RISCOS EM PROJETOS DE SOFTWARE	29
4.1	Processo de desenvolvimento	29
4.1.1	Empatia	29
4.1.2	Definição e ideação	30
4.1.3	Prototipação e testes	31
4.2	Elementos e dinâmica do jogo	32
5	AVALIAÇÃO	36
5.1	MEEGA+	36
5.2	<i>Focus group: Lovers vs Haters</i>	36
5.3	Aplicação do jogo	37
5.4	Resultados da aplicação do MEEGA+ e questões abertas	40
5.5	Resultados da aplicação do <i>focus group – Lovers VS Haters</i>	43
5.6	Ameaças à validade	44
6	CONCLUSÃO	46
	REFERÊNCIAS	48

1 INTRODUÇÃO

A Engenharia de *Software* (ES) é uma área da computação que se preocupa com todos os aspectos relacionados com a produção de *software*. Nesse contexto, a ES lida com as etapas do processo de desenvolvimento, visando a especificação do *software*, definição de arquitetura, metodologias de desenvolvimento, execução de verificação e validação, assim como a evolução do *software* e o gerenciamento de todo o processo. Essas atividades são realizadas de acordo com restrições organizacionais e financeiras, exigindo que os profissionais da área busquem soluções utilizando teorias, métodos e ferramentas eficientes (SOMMERVILLE, 2011).

No ensino de ES é necessário capacitar os alunos com conteúdo teórico e prático relacionados com o processo de desenvolvimento do *software*, métodos e ferramentas de resolução de problemas e gerenciamento das distintas etapas do processo. O processo tradicional para o ensino de ES consiste na apresentação dos conceitos no nível formal, com aulas expositivas, aulas de laboratório ou discussão de casos práticos (VON WANGENHEIM; SHULL, 2009). Em contrapartida, os resultados mostrados no estudo de Mendes *et al.* (2019) evidenciaram que os estudantes que cursaram disciplinas de ES estão insatisfeitos com metodologias de ensino extremamente expositivas e teóricas. Adicionalmente, apenas o conhecimento teórico, embora essencial, não é suficiente para que o aluno esteja preparado para a realidade do mercado, principalmente com relação à tomada de decisões em ambientes dinâmicos (DAMIAN; HADWIN; AL-ANI, 2006).

Dessa forma, é necessário evoluir da educação tradicional, que foca em apresentar as metodologias e técnicas da ES, para abordagens dinâmicas que envolvam aspectos de projetos reais (HUANG; DISTANTE, 2006). Nesse contexto, jogos educacionais (JE), também conhecidos como jogos sérios, podem se tornar ferramentas para estimular e despertar o interesse do aluno para aprender o conteúdo exigido (MOURA; SANTOS, 2018). Seu diferencial reside no fato do jogo ser projetado com o objetivo de aprendizagem sobre determinados conceitos, informações ou habilidades (SILVA *et al.*, 2016). Assim, os jogos sérios podem se tornar ferramentas para estimular e despertar o interesse do aluno para aprender o conteúdo exigido (MOURA; SANTOS, 2018).

Apesar de vários jogos terem sido produzidos para o ensino de ES, ainda é baixo o número de jogos utilizados no processo de ensino/aprendizagem (BATTISTELLA, 2016). Um dos problemas para a utilização destes jogos é que não há informações sintetizadas sobre as áreas de ensino que estes permitem apoiar dentro da ES, ou os requisitos para sua aplicação em sala de aula. Assim, foi identificada a necessidade de realizar um estudo secundário para

apresentar o estado da arte dos JE desenvolvidos para o ensino de ES no Brasil. Dessa forma, foi realizado um Mapeamento Sistemático da Literatura (MSL), um tipo de revisão da literatura que permite classificar os tópicos estudados dentro de uma área de pesquisa utilizando procedimentos metodológicos para diminuir as ameaças à validade dos resultados. A busca foi realizada nas bases dos principais veículos de publicações de trabalhos relacionados a ES e a informática na educação no Brasil. Ao todo, 25 jogos foram incluídos e tiveram suas informações extraídas e analisadas.

Como resultado da MSL percebeu-se que grande parte dos JE identificados na pesquisa estão direcionados para o ensino de gerenciamento de projetos, processo de *software* e testes de *software*. Dessa forma, identifica-se a oportunidade ainda pouco explorada de desenvolver jogos que apoiem o ensino de outros tópicos da ES, como Gerenciamento de Riscos, Engenharia de Requisitos ou Modelagem de Sistemas. Aliado com as expectativas dos alunos por uma disciplina voltada para a prática e com uso de materiais diversos (MENDES *et al.*, 2019), foi desenvolvido o jogo Arriscando, um jogo para o ensino de gerenciamento de riscos em projetos de desenvolvimento de *software*. O gerenciamento de riscos em projetos de desenvolvimento de *software* visa identificar e analisar os riscos que podem acontecer durante o projeto e planejar estratégias que diminuam a probabilidade de impactos negativos, decorrentes dos riscos, com o intuito de alcançar sucesso no projeto.

O jogo Arriscando tem o intuito de ensinar o gerenciamento de riscos em projetos de desenvolvimento de *software* usando uma dinâmica interativa e estimulante. O jogo foi concebido em formato de baralho para ser jogado em equipe, estimulando a interação entre os jogadores e a troca de conhecimento. Nele, são apresentadas categorias de riscos, riscos e mitigações que foram identificados nas publicações científicas de Boehm (1991), Lyytinen, Mathiassen e Ropponen (1998) e Shrivastava e Rathod (2015). O jogo foi aplicado em 3 turmas, de dois cursos de computação da Universidade Federal do Maranhão, e foi avaliado utilizando o *MEEGA+* (*Model for the Evaluation of Educational Games*) para avaliar a qualidade percebida; e a dinâmica *Lovers VS Haters* para captar pontos positivos e negativos do jogo, além de sugestões.

O objetivo principal deste trabalho foi produzir um jogo educacional que auxilie no ensino gerenciamento de riscos na área da ES. Sendo assim, o jogo educacional desenvolvido auxiliará no ensino da ES, pois irá solucionar problemas específicos encontrados dentro do cenário de ensino deste campo de estudo. Os objetivos secundários do trabalho partiram da identificação e compreensão dos aspectos que devem ser ensinados e as dificuldades dos docentes e discentes dentro do processo de ensino e aprendizado da ES. A realização da

verificação das soluções existentes utilizadas dentro dos problemas no ensino de ES, para entender até que ponto essas soluções são efetivas dentro do seu propósito de solução. Sendo possível desenvolver uma solução incorporando à conceituação dos objetos de aprendizado abrangendo os problemas específicos que serão identificados durante o processo de compreensão. Finalmente validando os impactos do jogo desenvolvido no seu cenário ideal, uma disciplina introdutória à ES.

O presente trabalho está organizado como segue. No capítulo 2 é apresentada a fundamentação teórica do trabalho, em que são apresentadas as concepções de Jogos Educacionais, *Design Thinking*, Gerenciamento de Riscos e Jogos Educacionais para o ensino de Gerenciamento de Riscos, temas e metodologia que apoiaram o desenvolvimento do trabalho. O capítulo 3 apresenta o Mapeamento Sistemático da Literatura realizado para a caracterização do estado da arte dos jogos educacionais para o ensino da engenharia de *software*. No capítulo 4 é apresentado o Jogo Arriscando, seu processo de desenvolvimento, elementos, regras e dinâmica. O capítulo 5 detalha as metodologias de avaliação utilizadas nesse trabalho, assim como sua aplicação e principalmente os resultados obtidos. Finalmente no capítulo 6 é realizada a conclusão do trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentadas as concepções de Jogos Educacionais, Design Thinking, Gerenciamento de Riscos e Jogos Educacionais para o ensino de Gerenciamento de Riscos temas e metodologia que formaram a fundamentação teórica do trabalho e apoiaram no seu desenvolvimento.

2.1 Jogos educacionais

Na disciplina de Engenharia de *Software*, o professor preocupa-se não apenas em transmissão de conceitos e teorias, mas principalmente com a associação do campo teórico e prático para o desenvolvimento pleno e eficiente de projetos de *software* que satisfaçam os requisitos dos usuários (PRIKLADNICKI *et al.*, 2009). Sendo assim, ao mesmo tempo que é extremamente relevante o conteúdo a ser ensinado dentro da sala de aula em disciplinas, também é notada a necessidade de manutenção dos aspectos didáticos e pedagógicos no ensino (SAVI; WANGENHEIM; BORGATTO, 2011). Geralmente, a ES é ensinada no nível formal com aulas expositivas, aulas de laboratório, discussão de casos práticos, dinâmicas de grupo e execução de projetos fictícios (PRIKLADNICKI *et al.*, 2009).

Segundo Savi, Wangenheim e Borgatto (2011), um dos grandes desafios enfrentados no ensino de computação é a necessidade de uso de métodos que permitam tornar o processo de ensino mais efetivo. Nesse contexto, a forma tradicional de ensino excessivamente centrado no professor, faz com que falte ao aluno a oportunidade de aplicar os conceitos de forma chamativa e entretida. Com isso, jogos educacionais, também conhecido como jogos sérios, surgiram como uma alternativa e seu diferencial reside no fato do jogo ser projetado com o objetivo da aprendizagem sobre determinados conceitos, informações ou habilidades (SILVA *et al.*, 2016). O JE se mostra como uma grande ferramenta para estimular e despertar o interesse do aluno para aprender o conteúdo exigido (MOURA; SANTOS, 2018). Assim, o uso de JEs durante o processo de ensino-aprendizagem se torna um grande aliado de todos os integrantes do sistema de ensino, pois os mesmos permitem: incitar o interesse dos alunos a aprender, ajudar a fixar o conteúdo abordado, divertir e, conseqüentemente, aumentar o índice de aprovações e diminuindo a evasão (SAVI; WANGENHEIM; BORGATTO, 2011).

2.2 Design thinking

O *Design Thinking* é uma metodologia centrada nas pessoas que integra conhecimentos de *design*, ciências sociais, engenharia e negócios. A metodologia combina o foco no usuário final com colaboração multidisciplinar e aprimoramento iterativo para produzir produtos, sistemas e serviços inovadores. Isso é realizado em um ambiente interativo que promove o aprendizado por meio de prototipagem conceitual e rápida (PLATTNER; MEINE; LEIFER, 2010).

O modelo de *Design Thinking* da d.School da *Stanford University* (HASSO PLATTNER INSTITUTE OF DESIGN AT STANFORD, 2019) é caracterizado por cinco fases iterativas e conectadas, que são Empatia, Definir, Idealizar, Prototipar e Testar. Durante a primeira fase, Empatia, são observados os usuários e seus comportamentos para compreender seu perfil e os problemas que enfrentam. Além disso, são coletadas as experiências dos usuários com os produtos e serviços já existentes no mesmo domínio. Na fase de definição, a finalidade é desenvolver um entendimento mais definido dos usuários do que na fase anterior e, com base nisso, definir uma posição sobre o objetivo. Assim, as ideias obtidas na fase anterior são resumidas em necessidades concretas, a fim de formular um objetivo adequado para o domínio em estudo (SCHALLMO; WILLIAMS; LANG, 2018).

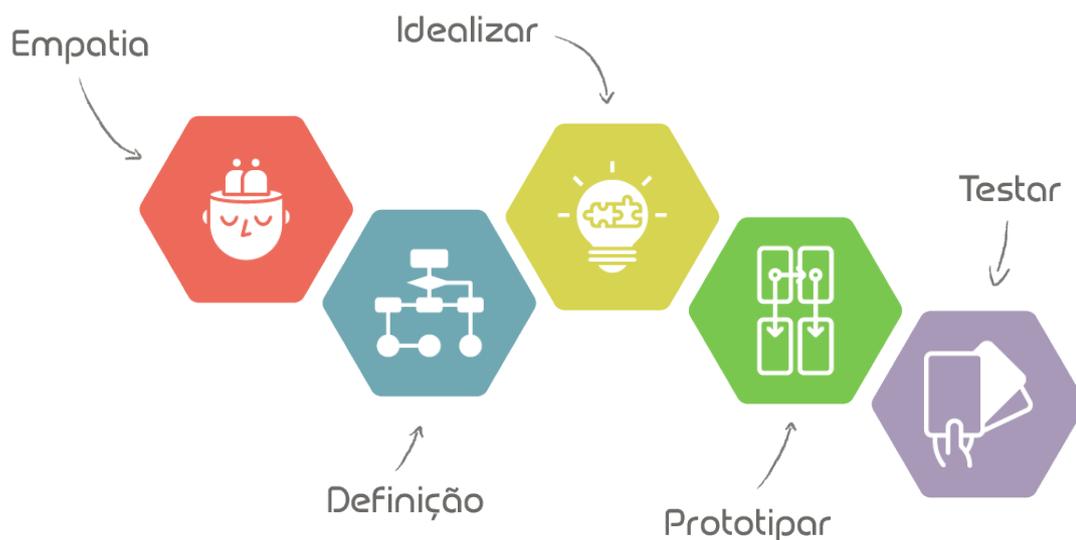


Figura 1: Processo do Design Thinking d.school.

Fonte: Elaborada pelo autor.

Durante a fase de idealização, são obtidas ideias para o problema formulado. O objetivo é derivar muitas ideias diferentes para a próxima fase. A criação de protótipos torna possível transformar ideias em soluções tangíveis. Na fase de prototipação são criados protótipos das ideias levantadas. Os protótipos são usados para a equipe e os usuários testá-los com o objetivo de validar a ideia e fazer melhorias. Os testes ocorrem no contexto dos usuários. Os testes também oferecem uma oportunidade de obter mais informações sobre o usuário e alterar a formulação do desafio, se necessário (SCHALLMO; WILLIAMS; LANG, 2018).

2.3 Gerenciamento de riscos

O objetivo de um projeto de *software* é fornecer o *software* para o cliente no prazo estabelecido, mantendo os custos dentro do orçamento, atendendo as expectativas do cliente e, também, zelar por uma equipe de desenvolvimento que trabalhe em condições amenas (SOMMERVILLE, 2011). O gerente de projetos é o responsável por controlar as pessoas e os processos em um projeto de *software*, sujeito à determinadas restrições de tempo e orçamento. Embora o sucesso de um projeto não seja garantido por um bom gerenciamento, o mau gerenciamento costuma resultar em falha do projeto (SHRIVASTAVA; RATHOD, 2015).

Um risco em projetos de *software* indica um aspecto ou propriedade particular de uma tarefa de desenvolvimento, processo ou ambiente, que, se for ignorado, aumentará a probabilidade de uma falha do projeto como, por exemplo: ameaças à operação de *software*, fontes de retrabalho de *software*, dificuldade de implementação ou demora (LYYTINEN; MATHIASSEN; ROPPONEN, 1998). Segundo Boehm (1991), o gerenciamento de riscos é definido como uma tentativa de formalizar riscos correlacionados a orientações de sucesso em um conjunto prontamente aplicável de princípios e práticas.

Para lidar com os riscos que surgem no processo de desenvolvimento de projetos de *software*, foram formalizadas abordagens de gerenciamento de riscos. Essas abordagens fornecem guias, em um formato relativamente padronizado, de como inquirir, observar, organizar e interpretar observações; e de como, subsequentemente, lançar ações de gerenciamento de riscos (LYYTINEN; MATHIASSEN; ROPPONEN, 1998). Estas abordagens são representadas usando conceitos de categorias de riscos, itens de risco e técnicas de resolução de riscos (LYYTINEN; MATHIASSEN; ROPPONEN, 1998).

Dentre os autores que identificaram os distintos riscos em projetos de desenvolvimento de *software* e suas respectivas mitigações, Boehm (1991) identificou 10 fontes de riscos que podem ocorrer, além das técnicas de gerenciamento que obtiveram maior sucesso em evitar ou

resolver os riscos. Por sua vez, no trabalho de Lyytinen, Mathiassen e Ropponen (1998), os riscos foram sintetizados a partir da literatura sobre gerenciamento de riscos de *software*. Foram classificados os problemas relacionados ao desenvolvimento de projetos e examinadas as estratégias de resolução usadas. Já no trabalho de Shrivastava e Rathod (2015), foram identificados diversos fatores de riscos em projetos de desenvolvimento de *software* distribuídos e seus respectivos métodos de gerenciamento de riscos. Finalmente, Leavitt (1964) propôs um modelo organizacional que pode ser utilizado para mapear as categorias de riscos identificadas, como: tarefa, estrutura, ator e tecnologia; que podem ser combinadas formando outras categorias, como: tarefa-ator, tarefa-tecnologia, estrutura-tarefa, ator-tecnologia, ator-estrutura e estrutura-tecnologia. Na tabela 1 são apresentadas as categorias identificadas nos trabalhos acima, com exemplos de riscos e mitigações.

Tabela 1. Categorias de riscos, com exemplos de riscos e suas respectivas mitigações.

Risco	Mitigação
Categoria: Tarefa	
Indisponibilidade de documentos de requisitos para testes	Ter um representante da equipe no local do cliente
	Envolver a equipe de testes no projeto desde o início
Categoria: Estrutura	
Horários e cronogramas irrealistas	Estimativa detalhada de custo e cronograma de várias fontes
	Projetar de acordo com o custo
	Adotar desenvolvimento Incremental
Categoria: Ator	
Riscos ocasionados pelas deficiências da equipe	Equipe trabalhando por correspondência de trabalho
	Fazer treinamentos cruzados
	Elaborar um pré-cronograma adequado
Categoria: Tecnologia	
Deficiências em componentes fornecidos por agentes externos	Elaborar avaliação comparativa
	Realizar inspeções
	Fazer análise de Compatibilidade
Categoria: Tarefa-Ator	
Atores inadequados desenvolvendo uma determinada tarefa	Treinar a equipe de desenvolvimento
	Definir as tarefas de acordo com as habilidades individuais
	Adotar uma estrutura de governança flexível
Categoria: Tarefa-Tecnologia	
A tecnologia adotada é inadequada para uma determinada tarefa	Adotar modelos de contingência para desenvolvimento de software
	Gerenciar opções de tecnologia
Categoria: Estrutura-Tarefas	
A estrutura existente é inadequada para a realização das tarefas	Adaptar a estrutura de decisão
	Modificar o modelo de processo
Categoria: Ator-Tecnologia	
Atores incompetentes ou competentes demais para a tecnologia dada	Fazer a prototipagem do produto
	Fazer análise técnica do processo

Risco	Mitigação
	Utilizar técnicas de cenários
	Fazer treinamento técnico da equipe
Categoria: Ator-Estrutura	
Falta de compromisso dos stakeholders	Aplicar táticas de liderança apropriadas
	contratar pessoas com boa cooperação e habilidades de gestão
	Adotar programas de formação de equipes
Categoria: Estrutura-Tecnologia	
Ajustes do sistema inadequado	Mudar a autoridade ou fluxo de trabalho
	Adotar / configurar novas tecnologias organizacionais

Fonte: (BOEHM, 1991), (LYYTINEN; MATHIASSEN; ROPPONEN, 1998), (SHRIVASTAVA; RATHOD, 2015)

2.4 Jogos educacionais para gerenciamento de riscos

Diversos jogos que para o ensino de assuntos da área da computação já foram desenvolvidos (BATTISTELLA; VON WANGENHEIM, 2016). Esses jogos geralmente são usados para apresentar conteúdos de forma dinâmica, rápida e despertar um maior interesse dos jogadores pelo assunto. No contexto de gerenciamento de riscos, no jogo *Riskware* (JARAMILLO; ALVAREZ; GONZALEZ-CALDERON, 2013), os jogadores devem percorrer um tabuleiro que aponta os relacionamentos entre riscos, mitigações e os relacionamentos com os outros jogadores. Os riscos ocorrem através da sorte definida pelo jogar de dados, e podem ser mitigados pelos recursos e objetos de controle que são comprados no início do jogo. O jogo se destaca pela capacidade de abordar diretamente a área do Gerenciamento de Riscos, através do relacionamento dos riscos com mitigações e seus impactos, além de proporcionar interação social. Apesar de estar limitado à escolha inicial dos jogadores, que devem usar o orçamento para obter recursos e objetos de controle que talvez não sejam usados.

O uso de jogos para o ensino da ES já é uma prática recorrente. No entanto, os autores deste trabalho não identificaram um jogo educacional em português para o ensino dos riscos, suas categorias e mitigações. Dessa forma, para o prosseguimento do trabalho apresentado, foi realizado um trabalho secundário com o objetivo de identificar outros jogos desenvolvidos para o ensino de ES. A descrição dessa etapa e suas atividades são apresentadas no próximo capítulo.

3 MAPEAMENTO SISTEMÁTICO DA LITERATURA

Para o desenvolvimento desse trabalho foi necessário caracterizar o estado da arte dos jogos educacionais desenvolvidos e utilizados para o ensino de engenharia de *software*, sendo assim possível conhecer as subáreas da ES abordadas por jogos educacionais e sua disponibilidade. Além disso, sendo o trabalho motivado pela existência de poucas informações sintetizadas sobre quais jogos podem ser usados dependendo do conteúdo de ES ministrado.

Desta forma, foi realizado um mapeamento sistemático da literatura para identificar e apresentar o estado da arte dos JE desenvolvidos para o ensino de ES no Brasil. Mapeamentos Sistemáticos da Literatura são um tipo de revisão da literatura que permitem classificar os tópicos estudados dentro de uma área de pesquisa e utilizam procedimentos metodológicos para diminuir as ameaças à validade dos resultados. A busca foi realizada nas bases dos principais veículos de publicações de trabalhos relacionados a ES e a informática na educação no Brasil. A estratégia estabelecida para guiar o processo de busca e extração será descrita nas próximas subseções.

3.1 Questões de pesquisa

Para identificar o estado da arte dos JE desenvolvidos no Brasil para o ensino de ES, foi definida a questão principal e as subquestões (SQ) de pesquisa que puderam ser respondidas a partir da análise dos estudos primários selecionados.

- Questão Principal: Quais jogos educacionais foram desenvolvidos no Brasil para o ensino de Engenharia de *Software*?
 - SQ1: Quais as plataformas onde os jogos funcionam?
 - SQ2: Quais os tópicos da Engenharia de *Software* são ensinados?
 - SQ3: Os jogos estão disponíveis para *download*?
 - SQ4: Qual é a dinâmica de jogo dos jogos educacionais?
 - SQ5: Que aspectos foram considerados no desenvolvimento dos jogos?
 - SQ6: Quais os objetivos de aprendizagem dos jogos educacionais e o perfil de alunos para os que foram desenvolvidos?
 - SQ7: Os jogos educacionais foram avaliados?

3.2 String de busca

Após a definição da questão e das subquestões de pesquisa, foi necessário elaborar uma *string* de busca que englobasse tanto os aspectos de JE quanto os aspectos da ES e seus objetos de estudo. Assim, os termos sobre JE foram derivados do trabalho de Cascini e Campos (2015) e do trabalho de Battistela, Wangenheim e Fernandes (2014). Além disso, os termos relacionados a ES, e seus objetos de estudo, foram selecionados a partir do relacionamento entre os tópicos de interesse de eventos de ES no Brasil (como SBES, SBQS, entre outros) e os tópicos da ES identificados nas ementas das disciplinas de introdução à ES de 10 cursos de graduação em computação segundo o Ranking Universitário da Folha de 2018¹.

Os termos selecionados foram relacionados com o uso das expressões booleanas "AND" e "OR". Adicionalmente, a *string* de busca resultante foi traduzida para a língua inglesa com o intuito de incluir os trabalhos publicados nessa língua. Devido a limitação de caracteres no motor de busca de algumas bases utilizadas no estudo a *string* de busca foi segmentada em duas partes, limitando a *string* em no máximo 255 caracteres, para que fosse possível realizar a busca. A *string* de busca é apresentada na Tabela 2.

Tabela 2: Strings de busca utilizadas no mapeamento sistemático da literatura.

<i>String</i> de busca em português	
("jogo" OR "simulação") AND ("educacional" OR "sério" OR "ensino" OR "aprendizado" OR "treinamento" OR "instrucional") AND ("medição" OR "qualidade" OR "teste" OR "inspeção" OR "manutenção" OR "engenharia de software" OR "ágil" OR "requisitos" OR "modelo" OR "processo" OR "arquitetura" OR "gerência" OR "gerenciamento")	
Parte 1	("jogo" OR "simulação") AND ("educacional" OR "sério" OR "ensino" OR "aprendizado" OR "treinamento" OR "instrucional") AND ("medição" OR "qualidade" OR "teste" OR "inspeção" OR "manutenção")
Parte 2	("jogo" OR "simulação") AND ("educacional" OR "sério" OR "ensino" OR "aprendizado" OR "treinamento" OR "instrucional") AND ("engenharia de software" OR "ágil" OR "requisitos" OR "modelo" OR "processo" OR "arquitetura" OR "gerência" OR "gerenciamento")
<i>String</i> de busca em inglês	
("game" OR "simulation") AND ("education" OR "serious" OR "teaching" OR "learning" OR "training" OR "instructional") AND ("software engineering" OR "agile" OR "requirements" OR "process" OR "architecture" OR "configuration" OR "maintenance" OR "evolution" OR "metrics" OR "measurement" OR "quality" OR "test" OR "verification")	
Parte 1	("game" OR "simulation") AND ("education" OR "serious" OR "teaching" OR "learning" OR "training" OR "instructional") AND ("measurement" OR "quality" OR "test" OR "inspection" OR "maintenance")
Parte 2	("game" OR "simulation") AND ("education" OR "serious" OR "teaching" OR "learning" OR "training" OR "instructional") AND ("software engineering" OR "agile" OR "requirements" OR "model" OR "process" OR "architecture" OR "management")

Fonte: Elaborada pelo autor.

¹ <https://ruf.folha.uol.com.br/2018/ranking-de-cursos/computacao/>

3.3 Veículos de publicação e bases digitais

Para definir as bases onde foram pesquisados os trabalhos relatando estudos primários, foram considerados os veículos de publicação segundo a especificidade do estudo secundário sendo realizado. Foram consideradas as bases de 4 periódicos e 4 eventos científicos brasileiros que tratam do tema Informática na Educação, segundo Detroz, Hinz e Da Silva Hounsell (2015).

As bases com trabalhos voltados aos tópicos de interesse específicos à Engenharia de *Software* no Brasil foram selecionadas a partir da lista de revistas e congressos relevantes para a área de ES segundo a Comissão Especial de Engenharia de *Software*². Devido a relevância para o trabalho desenvolvido, também foi considerado o Simpósio Brasileiro de Games por ser o maior evento acadêmico da América Latina na área de Jogos e Entretenimento Digital. As bases utilizadas no trabalho são apresentadas na Tabela 3.

Tabela 3: Bases utilizadas no mapeamento sistemático da literatura.

Área de Interesse	Bases Selecionadas
Informática na Educação	Revista Brasileira de Informática na Educação – RBIE Revista de Informática Teórica e Aplicada – RITA Revista Informática na Educação: Teoria & Prática – InfEducTeoriaPratica Revista Novas Tecnologias na Educação – RENOTE Simpósio Brasileiro de Informática na Educação – SBIE Workshop de Desafios da Computação Aplicada à Educação – DESAFIE Workshop de Informática na Escola – WIE Workshop sobre Educação em Computação – WEI
Engenharia de Software	Simpósio Brasileiro de Componentes, Arq. e Reutilização de Software – SBCARS Simpósio Brasileiro de Engenharia de Software – SBES Simpósio Brasileiro de Qualidade de Software – SBQS Simpósio Brasileiro de Sistemas de Informação – SBSI Simpósio Brasileiro de Teste de Software Sistemático e Automatizado – SAST Workshop em Engenharia de Requisitos – WER Workshop sobre Aspectos Sociais, Humanos e Econômicos de Software – WASHES Workshop sobre Visualização, Evolução e Manutenção de Software – VEM
Jogos	Simpósio Brasileiro de Games – SBGAMES

Fonte: Elaborada pelo autor.

² <http://comissoes.sbc.org.br/ce-es/eventos.php?lang=pt-br>

Como já exposto, dentre as bases selecionadas algumas possuíam limitação de caractere na busca. Além de outras que não possuíam sistema de busca, nesses casos a execução da busca pelos artigos se deu de forma manual. Nesse contexto, foram analisados manualmente os artigos publicados: nas edições de 2007 a 2019 do SBCARS, nas edições do SBES de 2006, 2008 a 2019, nas edições de 2011, 2014 e 2018 do SBQS, nas edições 2004, 2006 a 2019 do SBGAMES e todas as edições do SAST e WER. As bases são apresentadas quanto suas características de busca na Tabela 4.

Tabela 4: Bases quanto as suas características de busca.

Tipo de Busca	Bases
Busca automática	DESAFIE, InfEducTeoriaPratica, RBIE, RENOTE, RITA, SBIE, SBSI, VEM, WASHES, WEI, WIE
Busca manual	SBCARS, SBES, SBQS, SAST, WER, WASHES, SBGAMES
Busca com limitação de caracteres	DESAFIE, InfEducTeoriaPratica, RBIE, RENOTE, RITA, SBIE, WIE

Fonte: Elaborada pelo autor.

3.4 Critérios de inclusão e exclusão

Para que fosse possível selecionar e analisar com maior precisão os estudos primários que respondam à questão principal de pesquisa do mapeamento sistemático da literatura, foram definidos os seguintes critérios para incluir ou excluir uma publicação.

- Critério de Inclusão: (CI1) O artigo descreve com detalhes um jogo educacional para o ensino de Engenharia de *Software*.
- Critérios de Exclusão: (CE1) O artigo descreve a aplicação de técnicas da Engenharia de *Software* em jogos educacionais, mas o jogo não é de ES; (CE2) O artigo não descreve um jogo educacional para o ensino de Engenharia de *Software*.

3.5 Estratégia para extração de dados

Com o intuito de facilitar a execução das próximas etapas do mapeamento sistemático foi elaborado um formulário para a extração de dados dos trabalhos selecionados na etapa anterior. O formulário ainda funciona como um guia para a extração das respostas às questões secundárias definidas durante o planejamento do estudo secundário, pois apresentam questões

de fácil entendimento e possíveis respostas, ou uma descrição do tipo de resposta esperada (CABREJOS; VIANA; DOS SANTOS, 2018). O processo de extração foi realizado utilizando a ferramenta Parsifal³. A Tabela 5 apresenta parte do formulário de extração preparado na ferramenta Parsifal considerando as subquestões de pesquisa definidas

Tabela 5: Formulário de extração de dados.

Pergunta	Possível Resposta
Qual é o nome do jogo?	Escrever o nome do jogo como descrito na publicação.
O jogo é digital?	<input type="checkbox"/> Sim <input type="checkbox"/> Não
Se o jogo é digital, qual a plataforma em que o jogo funciona?	<input type="checkbox"/> Desktop <input type="checkbox"/> Mobile <input type="checkbox"/> Multiplataforma <input type="checkbox"/> Realidade Aumentada <input type="checkbox"/> Web
Qual é o tópico da Engenharia de Software ensinada?	<input type="checkbox"/> Desenvolvimento Ágil <input type="checkbox"/> Projeto de Software <input type="checkbox"/> Engenharia de Requisitos <input type="checkbox"/> Estimativa de Software <input type="checkbox"/> Gerência de Configuração <input type="checkbox"/> Gerência de Projetos <input type="checkbox"/> Gerenciamento de Riscos <input type="checkbox"/> Processo de Software <input type="checkbox"/> Produto de Software <input type="checkbox"/> Qualidade de Software <input type="checkbox"/> Teste de Software
Como funciona o jogo?	Descrever como o jogo funciona como descrito na publicação.
O jogo está disponível para download?	<input type="checkbox"/> Sim <input type="checkbox"/> Não
Se o jogo estiver funcionando, qual é o link para download?	Citar o link em que o software, material, do jogo está disponível para download.
O jogo seguiu algum Objetivo de Aprendizagem? Se sim, qual(is)?	Descrever quais objetivos de aprendizagem foram apresentados para serem cumpridos por meio da aplicação do jogo.
O jogo seguiu alguma metodologia de desenvolvimento?	<input type="checkbox"/> Sim <input type="checkbox"/> Não

³ <https://parsif.al/>

Se o jogo seguiu alguma metodologia de desenvolvimento, descreva a metodologia usada.	Citar e descrever a metodologia de desenvolvimento utilizada para desenvolver o jogo.
Qual é o público alvo do jogo?	<input type="checkbox"/> Alunos de Ensino Fundamental <input type="checkbox"/> Alunos de Ensino Infantil <input type="checkbox"/> Alunos de Ensino Médio <input type="checkbox"/> Alunos de Graduação - Com Conhecimentos no Tema <input type="checkbox"/> Alunos de Graduação - Novatos <input type="checkbox"/> Profissionais da Área
O jogo foi avaliado?	<input type="checkbox"/> Sim <input type="checkbox"/> Não
Se o jogo foi avaliado, descreva a metodologia usada.	Citar a(s) metodologia(s) utilizadas para avaliar o jogo e descrever os passos realizados.

Fonte: Elaborada pelo autor.

3.6 Resultados e discussão

Ao todo, foram identificados 25 jogos até setembro de 2019. As *strings* de busca retornaram 250 artigos. Após a leitura do título e resumo foram selecionados 21 artigos com base nos critérios de inclusão e exclusão. Após a leitura do artigo completo, 18 artigos foram selecionados. Nas bases em que não havia motor de busca, foram encontrados 7 artigos reportando JE para o ensino de ES. Em ordem decrescente, as bases que continham artigos sobre jogos educacionais para o ensino de ES são: Simpósio Brasileiro de Informática na Educação – SBIE (11), Simpósio Brasileiro de Engenharia de *Software* – SBES (5), Revista Novas Tecnologias na Educação – RENOTE (3), *Workshop* sobre Educação em Computação – WEI (2), Simpósio Brasileiro de Sistemas de Informação – SBSI (2), Simpósio Brasileiro de Qualidade de *Software* – SBQS (1) e Simpósio Brasileiro de *Games* – SBGAMES (1).

Com relação às áreas da ES abordadas, houve jogos que abordaram mais de uma área, principalmente em simulações que abordam todo o processo de desenvolvimento de *software*. É perceptível que a grande parte dos jogos sérios incluídos na pesquisa estão direcionados para o ensino de gerenciamento de projetos, processo de *software* e testes de *software*. Dessa forma, identifica-se a oportunidade ainda pouco explorada de desenvolver jogos que apoiem o ensino de outros tópicos da ES, como Gerenciamento de Riscos, Engenharia de Requisitos ou Modelagem de Sistemas. Além disso, considerando a plataforma, 76% dos jogos identificados são jogos digitais e 24% jogos não digitais. Os jogos digitais foram desenvolvidos em sua maioria (57,9%) para sistemas operacionais de computadores *desktops*. Os jogos para

plataformas *web* representaram 26,3% dos jogos digitais e os jogos *mobile* e multiplataforma são 15,8% em conjunto. Foram 6 jogos não digitais incluídos nesse trabalho que se configuraram em jogos de cartas ou de tabuleiro.

Tabela 6: Sumarização do processo de busca e inclusão dos trabalhos.

Base	Artigos Retornados na Busca	Artigos Seleccionados Após Primeiro Filtro	Artigos Seleccionados Após Segundo Filtro
Bases com Motor de Busca			
Revista Brasileira de Informática na Educação – RBIE	42	0	0
Revista de Informática Teórica e Aplicada – RITA	0	0	0
Revista Informática na Educação: Teoria & Prática – InfEducTeoriaPratica	10	0	0
Revista Novas Tecnologias na Educação – RENOTE	49	3	3
Simpósio Brasileiro de Informática na Educação – SBIE	110	14	11
Workshop de Desafios da Computação Aplicada à Educação – DESAFIE	2	0	0
Workshop de Informática na Escola – WIE	25	0	0
Workshop sobre Educação em Computação – WEI	6	2	2
Simpósio Brasileiro de Sistemas de Informação – SBSI	6	2	2
Workshop sobre Visualização, Evolução e Manutenção de Software – VEM	0	0	0
Bases sem Motor de Busca			
Simpósio Brasileiro de Componentes, Arq. e Reutilização de Software – SBCARS	-	0	0
Simpósio Brasileiro de Engenharia de Software – SBES	15	5	5
Simpósio Brasileiro de Qualidade de Software – SBQS	3	1	1
Simpósio Brasileiro de Teste de Software Sistemático e Automatizado – SAST	-	2	0
Workshop em Engenharia de Requisitos – WER	-	0	0
Workshop sobre Aspectos Sociais, Humanos e Econômicos de Software – WASHES	-	0	0
Simpósio Brasileiro de Games – SBGAMES	-	1	1

Fonte: Elaborada pelo autor.

Com relação à adoção de metodologias de desenvolvimento para a criação dos jogos educacionais, somente 4 jogos apresentaram a descrição das mesmas. Os jogos "Kahoot! PMQuiz" e "Scrum-Scape" utilizaram o ENgAGED (*EducatioNAl GamEs Development*), um processo de desenvolvimento de jogos educacionais iterativo que foi desenvolvido aliando as características do desenvolvimento de jogos e do *design* instrucional (BATTISTELLA, 2016). O livro *Game Design Workshop* (FULLERTON, 2018) foi utilizado para desenvolver o jogo "SCRUMI", no livro é apresentado métodos e ferramentas para desenvolver jogos de forma prática e não técnica. Para o desenvolvimento do jogo "Testing Game" foi utilizado o método AIMED, uma abordagem ágil para o desenvolvimento de jogos e recursos educacionais (ROCHA *et al.*, 2017). Por fim, o jogo "UbiRE", foi utilizado uma sistematização do processo de desenvolvimento de jogos de simulação para treinamento.

Sobre os objetivos de aprendizagem dos JE analisados, não foram especificados de maneira formal nos trabalhos originais. Porém, foi possível identificar o objetivo de aprendizagem conhecimento, compreensão, aplicação e análise do domínio cognitivo, segundo a taxonomia de Bloom. Todos os jogos analisados são voltados para o ensino de Engenharia de *Software* para alunos de graduação com conhecimento prévio sobre o assunto ou não. Alguns jogos, ainda são voltados para profissionais na área como no "SDM", "RSKManager", "Scrum-Scape", "Planager", "Master of the process", "IslandTest" e "2tscrum".

Quanto à avaliação dos jogos, 75% realizaram e relataram atividades de avaliação nos trabalhos analisados. Os jogos "Kahoot! PMQuiz", "EAREq-GAME", "SCRUM-SCAPE", "CleanGame" e "BlackBox" realizaram a sua avaliação utilizando o MEEGA+ (PETRI; VON WANGENHEIM; BORGATTO, 2017), modelo sistemático para avaliação da qualidade de jogos educacionais na perspectiva do aprendiz. Foram utilizadas técnicas de análise de grupos de controle nos jogos "SPARSE", "RSKManager" e "Jogo das 7 falhas", que levantaram a eficácia do jogo através da comparação sistemática do desempenho de alunos que utilizaram o jogo com alunos que não utilizaram o jogo. Por fim, os jogos "SDM", "SCRUMI", "Masters of the Process", "IslandTest", "Agility Scrum", "GameES" e "UbiRe" realizaram testes de usabilidade, no formato de formulário e baseados na teoria das heurísticas de usabilidade. A Tabela 7 apresenta a lista de jogos identificados em ordem alfabética, o tipo de jogo, a plataforma, os tópicos de ES que o mesmo considera e, caso esteja disponível para *download*, o link do mesmo.

Tabela 7: Jogos identificados no mapeamento sistemático.

Referência	Nome do Jogo	Tipo do Jogo	Plataforma que o Jogo Funciona	Tópicos da ES Ensinados	Link
(BRITO; VIEIRA, 2017)	2TScrum	Não Digital	----	Desenvolvimento Ágil	----
(FERREIRA <i>et al.</i> , 2018)	Adventur eSECO	Digital	Desktop	Gerência de Configuração	----
(DE CASTRO <i>et al.</i> , 2017)	Agility Scrum.	Não Digital	----	Desenvolvimento Ágil	----
(FEITOSA; CAMPOS, 2010)	AprendE S	Não Digital	----	Gerência de Projetos, Processo de Software	----
(ARAÚJO <i>et al.</i> , 2017)	BlackBox	Digital	Desktop	Teste de Software	----
(DOS SANTOS <i>et al.</i> , 2019)	CleanGame	Digital	Web	Teste de Software Gerenciamento de Riscos	https://bit.ly/2W6xCIB
(PETRI; CHIAVEGATTI, 2015)	EAReq-GAME.	Digital	Desktop	Engenharia de Requisitos	----
(OLIVEIRA; CINTRA; NETO, 2013)	eRiskGame	Digital	Web	Processo de Software	----
(DE MEDEIROS <i>et al.</i> , 2013)	GameES.	Digital	Web	Gerência de Projetos	----
(BEPPE <i>et al.</i> , 2018)	GreaTest Card Game	Não Digital	----	Teste de Software	----
(GONÇALVES; THIRY; ZOUCCAS, 2011)	Ilha dos Requisitos	Digital	Web	Engenharia de Requisitos	http://www.incremental.com.br/ilhadosrequisitos/
(QUEIROZ; PINTO; SILVA, 2019)	IslandTest.	Digital	Web	Teste de Software	http://www.itaz.com.br/islandtest
(DINIZ; DAZZI, 2012)	Jogo das 7 Falhas	Digital	Desktop	Gerenciamento de Riscos	----
(PETRI <i>et al.</i> , 2016)	Kahoot! PMQuiz.	Digital	Multiplataforma	Gerência de Projetos	Plataforma para elaboração do quiz: https://getkahoot.com/ .
(CAMPOS <i>et al.</i> , 2012)	Kallango	Digital	Mobile	Gerência de Projetos	----
(SARINHO, 2019)	Masters of the Process	Não Digital	----	Gerência de Projetos, Processo de Software	----
(PRIKLADNICK I; ROSA; KIELING, 2007)	Planager	Digital	Desktop	Gerência de Projetos, Processo de Software	----
(MOURA; SANTOS, 2018)	ProcSoft.	Não Digital	----	Gerência de Projetos, Processo de Software	----

Referência	Nome do Jogo	Tipo do Jogo	Plataforma que o Jogo Funciona	Tópicos da ES Ensinados	Link
(PALUDO; RAABE; BENITTI, 2013)	RSK Manager	Digital	Desktop	Gerência de Projetos, Gerenciamento de Riscos	----
(SOUZA <i>et al.</i> , 2017)	SCRUMI	Digital	Desktop	Desenvolvimento Ágil, Gerência de Projetos	----
(BATTISTELLA; DE CAMARGO; VON WANGENHEIM, 2016)	Scrum-Scape.	Digital	Desktop	Desenvolvimento Ágil, Gerência de Projetos	http://www.gqs.ufsc.br/scrum-scape-an-role-playing-game-rpg-to-reinforce-scrum-concepts/
(KOHWALTER; CLUA; MURTA, 2011)	SDM.	Digital	Desktop	Gerência de Projetos, Processo de Software	----
(SOUZA <i>et al.</i> , 2010)	SPARSE	Digital	Desktop	Gerência de Projetos	----
(VALLE; ROCHA; MALDONADO, 2017)	Testing Game.	Digital	Desktop	Teste de Software	----
(CAMPOS <i>et al.</i> , 2012)	UbiRE	Digital	Multiplataforma	Engenharia de Requisitos	----

Fonte: Elaborada pelo autor.

A caracterização do estado da arte dos jogos educacionais para o ensino de Engenharia de *Software* desenvolvidos no Brasil possibilitou a identificação de aspectos decisivos para o desenvolvimento de um novo jogo educacional. Além da escassa disponibilização dos jogos já existentes, de forma que é baixa a probabilidade de que esses sejam aplicados como plataformas de ensino ou a replicação de seus estudos. Ainda foi possível identificar a oportunidade de desenvolver um novo jogo para o ensino de uma subárea da ES menos explorada (Gerenciamento de Riscos). Partiu-se para o desenvolvimento de um novo jogo educacional para o ensino de gerenciamento de riscos em projetos de desenvolvimento de *software*, dessa forma, o jogo Arriscando foi proposto para apoiar o ensino de riscos e mitigações. As etapas de desenvolvimento, aplicação e avaliação serão apresentadas nos próximos capítulos.

4 ARRISCANDO: UM JOGO PARA O ENSINO DE GERENCIAMENTO DE RISCOS EM PROJETOS DE SOFTWARE

Arriscando é um jogo de baralho não digital baseado em uma dinâmica de ataque (o surgimento de riscos em projetos de desenvolvimento de *software*) e defesa (o uso de mitigações adequadas para estes riscos) para abordar os imprevistos que podem ocorrer em um projeto de desenvolvimento de *software*. Foram usados os conteúdos sintetizados de trabalhos que listam categorias de riscos, riscos e mitigações presentes em Boehm (1991), Shrivastava e Rathod (2015) e Lyytinen, Mathiassen e Ropponen (1998) (ver Subseção 2.3).

O objetivo do jogo é apresentar aos participantes as categorias de riscos, riscos existentes dentro destas categorias e as mitigações que podem ser usadas para diminuir o impacto negativo dos riscos nos projetos. Com isso, o jogo permite fazer o relacionamento existente entre categorias de riscos e riscos, assim como riscos e mitigações. Conseqüentemente, espera-se que os participantes adquiram conhecimentos sobre gerenciamento de riscos, permitindo que entendam o impacto que os riscos podem ocasionar nos projetos de *software* e a importância de realizar planos de gerenciamento de riscos. Durante o jogo, os participantes podem usar seus conhecimentos prévios adquiridos durante as aulas de ES juntamente com os elementos do jogo para evitar que riscos aconteçam em seu projeto.

Diante do exposto, os objetivos de aprendizagem do jogo são (ADAMS, 2015): conhecimento e compreensão (listar e interpretar os diferentes riscos, assim como suas mitigações) e análise (relacionar os riscos com suas possíveis mitigações). Estes objetivos podem ser alcançados a partir da repetição e da tentativa de relacionar os riscos com suas respectivas mitigações, avaliando se uma mitigação minimiza um risco.

4.1 Processo de desenvolvimento

O jogo foi desenvolvido utilizando a metodologia *Design Thinking*. Uma abordagem colaborativa e iterativa para desenvolvimento de sistemas e produtos inovadores. As etapas e atividades realizadas no processo de desenvolvimento serão apresentadas a seguir.

4.1.1 Empatia

O desenvolvimento do jogo foi iniciado na primeira etapa da metodologia do *Design Thinking*, a Empatia. Nessa etapa foram identificados os tópicos ensinados nas disciplinas introdutórias à ES, através da análise das ementas dos 10 melhores cursos de computação do

país segundo o ranking da Revista Folha (FOLHA DE SÃO PAULO, 2018). O *ranking* inclui as seguintes universidades em ordem de classificação: UNICAMP, USP, UFRGS, UFMG, UFRJ, UFPE, UFSC, UFSCAR, PUC-RIO, UNB. Também foi levantado os tópicos ensinados na disciplina de Engenharia de *Software I* da Universidade Federal do Maranhão. Os principais tópicos identificados foram: Processo e Produto de *Software*, Estimativa de *Software*, Engenharia de Requisitos, *Design* (Projeto de *Software*), Implementação de *Software*, Qualidade de *Software*, Teste de *Software*, Gerência de Configuração, Gerenciamento de Riscos e Gerência de Projetos.

Para entender melhor o público alvo do jogo foi utilizado o trabalho de Mendes, Costa, et al., (2019) como fonte para a identificação das expectativas e dificuldades de alunos de graduação no ensino de ES. Trabalho que buscou entender as necessidades dos alunos iniciantes ES através da aplicação de um questionário quanto à experiência dos alunos de uma disciplina introdutória de ES. Foi capturada a expectativa de alunos que ainda não cursaram a disciplina, assim como a visão de alunos que já a cursaram, reprovando ou não. Os resultados mostraram que o ensino com abordagem absolutamente teórica não corresponde às expectativas dos alunos para a disciplina. Mostra-se uma insatisfação sobre a didática utilizada pelos docentes e a dificuldade de aplicar os conteúdos na prática (MENDES *et al.*, 2019).

Ainda foi realizado um MSL, apresentado no capítulo 3, que caracterizou o estado da arte dos JE para o ensino da ES. Permitindo identificar os pontos fortes, abordagem dos conteúdos de forma dinâmica e interativa, e fracos, a indisponibilidade dos jogos para replicação, dos jogos disponíveis na literatura. Além das oportunidades existentes quanto a baixa quantidade de jogos desenvolvidos para o ensino de subáreas da ES.

4.1.2 Definição e ideação

Os resultados obtidos na fase de Empatia, foram um guia para a definição dos aspectos que o jogo a ser desenvolvido deveria ter. Tendo identificado o público alvo, os alunos da disciplina Engenharia de *Software I* e a subárea que seria abordada, o Gerenciamento de Riscos. Foi definido que o objetivo do jogo seria apresentar conceitos de gerenciamento de riscos e permitir que os alunos treinassem seus conhecimentos em sala de aula de forma trivial, fácil e lúdica. Visando alcançar os objetivos de aprendizagem conhecimento, compreensão e análise.

No processo de ideação foram realizadas diversas propostas de gêneros de jogos e abordagens a serem utilizadas no jogo para o ensino de gerenciamento de riscos em projetos de *software*. Foi proposto o desenvolvimento de um jogo no estilo de perguntas e respostas (quiz),

porém essa ideia foi descartada devido a sua simplicidade, na abordagem e transmissão do conteúdo, e por já serem altamente exploradas em diversos jogos educacionais. Outra proposta foi a de desenvolver um jogo de tabuleiro que apresentasse todo o processo de desenvolvimento de projetos de *software*, mas com enfoque na apresentação de conceitos de gerenciamento de riscos, porém essa foi descartada por ter uma grande demanda, de tempo e conteúdo, tanto dos executores, quanto dos jogadores.

Ainda foi proposta a realização da gamificação do ensino de gerenciamento de riscos na disciplina, através da inserção dos elementos de derivados dos *games*, como pontuação, recompensas, classificação, entre outros. Considerando o esforço necessário para adequar a nova abordagem a proposta foi descartada. Dessa forma a ideia que prevaleceu foi a de desenvolver um jogo de baralho, não digital, apresentando uma dinâmica de ataques e defesas. Para que fosse possível apresentar o conteúdo aos alunos iniciantes na ES e permitir a aprendizagem, reconhecimento e associação dos conceitos.

4.1.3 Prototipação e testes

Para o processo de prototipação do jogo foi necessário identificar os riscos e suas respectivas mitigações existentes. Dessa forma foram buscados trabalhos científicos na área de GR que apresentavam tais informações de forma objetiva e condensada. Assim, foi identificado abordagens formalizadas de gerenciamento de riscos, que são guias de como inquirir, observar, organizar, interpretar para lançar ações de gerenciamento de riscos.

As abordagens, que incluem riscos, categorias de riscos e mitigações para os riscos foram sintetizados dos trabalhos de Boehm (1991), Shrivastava e Rathod (2015) e Lyytinen, Mathiassen e Ropponen (1998). Os trabalhos apresentam riscos sintetizados e classificados, além das respectivas estratégias de resolução usadas, as fontes de riscos e as técnicas de gerenciamento que obtiveram maior sucesso em evitar ou resolver os riscos segundo os fatores de riscos encontrados em projetos de desenvolvimento de *software* distribuídos e seus respectivos métodos de gerenciamento. Todos os trabalhos utilizam o modelo de classificação de Leavitt (1964), que classifica os elementos de uma organização em quatro componentes (tarefas, pessoas, estrutura, tecnologia). Esses componentes podem ser combinados entre si formando novos componentes compostos. Essa classificação evidencia que mudanças realizadas em um componente podem acarretar impactos nos outros componentes.

Para a validação da dinâmica do jogo e do conteúdo utilizado diversas partidas de testes foram realizadas. Dessa forma foi possível corrigir a apresentação dos conceitos principais do

jogo, os ricos, mitigações e categorias. Além de aperfeiçoar a dinâmica, balanceando a competitividade e adicionando estratégias que estimulem a jogar permitindo que seja facilitado o entendimento e a jogabilidade.

Como explicado acima, o conteúdo e a dinâmica do jogo foram pensados durante um conjunto de reuniões entre alunos e professores do grupo de Engenharia de *Software* da UFMA, visando o aprendizado de conceitos essenciais do GR em projetos de desenvolvimento de *software*. Desta forma, as decisões de *design* do jogo foram pensadas para que seja possível fazer um relacionamento direto dos elementos do jogo com as situações existentes em projetos de *software*, como: as categorias usadas para classificar os elementos, os recursos de uma organização, os riscos que podem ocorrer nos projetos e as mitigações que podem ser feitas para acabar ou diminuir os efeitos negativos de um risco. Finalmente, considerando que o GR é um tópico que é apresentado perto do final de disciplinas introdutórias de ES, o aluno pode utilizar os conhecimentos adquiridos nos outros tópicos para pensar em possíveis mitigações que serão contrastadas com as soluções indicadas pelos trabalhos indicados na Seção 2.3. Os elementos da versão final do jogo Arriscando e sua jogabilidade será apresentada na próxima seção.

4.2 Elementos e dinâmica do jogo

O jogo Arriscando foi projetado para ser jogado em grupos de 3 a 5 pessoas. Em que cada participante irá interpretar o papel de um gerente de projetos que deverá desenvolver planos de gerenciamento de riscos. O objetivo do jogo é ser o participante que detém a maior quantidade de unidades de dinheiro ao fim da partida. Para isso, a dinâmica do jogo constitui-se em ataques com cartas de risco e defesas através do uso de cartas de mitigação, ou o próprio conhecimento do jogador.

O jogo é composto pelos seguintes elementos: 28 cartas de risco, 19 cartas de mitigação, 25 notas de 5 unidades de dinheiro, 5 notas de 10 unidades de dinheiro e 1 dado, que são apresentados na Figura 2. A seguir os itens do jogo e o processo de utilização dos mesmos durante o jogo serão descritos.

Cartas de Risco: são cartas de cor alaranjada, que possuem o cabeçalho “Risco”, evidenciando sua função. Na frente das cartas de risco são apresentados: (a) uma categoria de risco, que pode ser simples ou composta (uma combinação de duas categorias de risco); (b) a descrição de um risco; e (c) uma imagem para a categoria daquele risco. No verso é apresentada

a mesma categoria e descrição do risco que é apresentado na frente da carta, acrescentando as mitigações para este risco.

Cartas de Mitigação: são cartas esverdeadas, que possuem o cabeçalho “Mitigação”, evidenciando sua função. Na frente é apresentado somente a função da carta. No verso das cartas de mitigação são apresentados: (a) uma categoria de risco, que pode ser simples ou composta; e (b) várias mitigações que podem ser usadas para mitigar riscos daquela categoria do risco.

Notas de Unidades de Dinheiro: as notas de unidades de dinheiro serão pagas após analisar o efeito da ação dos jogadores. Há notas laranjas que valem 5 unidades de dinheiro e notas verdes que valem 10 unidades de dinheiro.

Dado: o dado será usado para fazer o sorteio de qual participante irá iniciar a partida.

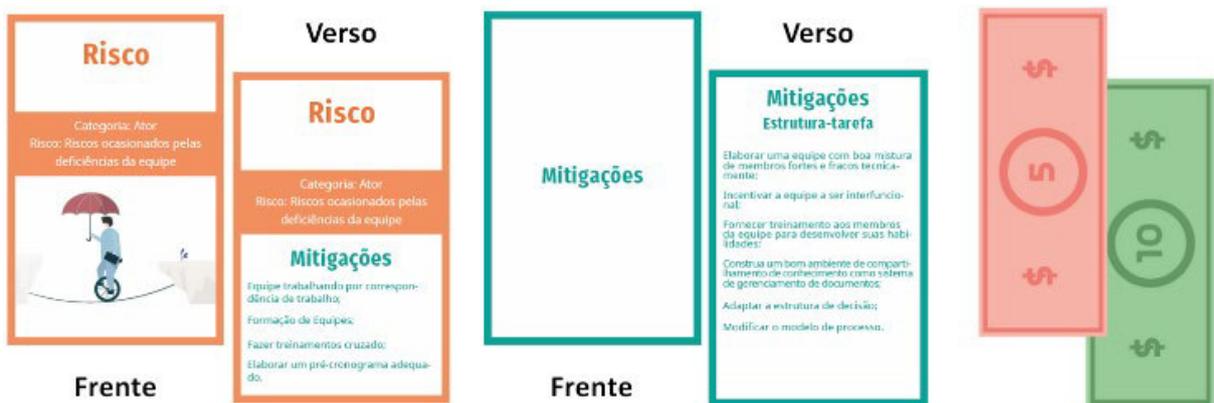


Figura 2: Elementos do jogo Arriscando.

Fonte: Elaborado pelo autor.

O jogo acontece em 4 rodadas. Com o grupo formado, cada pessoa deve, individualmente, receber 3 cartas de risco e 2 cartas de mitigação, além de uma quantia de 30 unidades de dinheiro do jogo (4 notas de 5 unidades de dinheiro e 1 nota de 10 unidades de dinheiro). Cada participante joga o dado uma vez e o participante que tirar o maior número no dado irá iniciar a partida, seguindo a ordem de jogo em sentido horário.

O jogador da vez sempre terá que realizar um ataque, para isso ele deverá escolher uma carta de risco, dentre as que estão na sua mão, e atacar um jogador de sua escolha. Deverá ser verbalizado para todos os jogadores da partida quem é a pessoa que será atacada, qual a categoria e o risco da carta escolhida, frase que deverá ser repetida 3 vezes. Quando um jogador é atacado, é possível se defender de duas formas: (a) usando uma carta de mitigação da mesma categoria do risco com o que foi atacado para se defender automaticamente; ou (b) se não possuir ou não desejar usar uma carta de mitigação, deverá verbalizar pelo menos uma

mitigação presente na carta de risco da qual foi atacado em, no máximo, 1 minuto. Se foi feito um ataque com sucesso, ou seja, o jogador que foi atacado não conseguiu se defender, esse deverá pagar 5 unidades de dinheiro para o jogador que o atacou. Se foi feita uma defesa adequada, em outras palavras um ataque sem sucesso, então o atacante deverá pagar 5 unidades de dinheiro para o jogador que fez a defesa da forma correta.

A Figura 3 apresenta um exemplo de ataque e defesa em uma das rodadas do jogo, entre dois jogadores. Neste exemplo, o jogador atacante utiliza a carta de risco “Indisponibilidade de documentos de requisitos para teste” da categoria “Tarefa”. O jogador na defesa possui uma carta de mitigação da mesma categoria. Portanto, ele pode se defender usando esta carta e faturar 5 unidades de dinheiro do jogador que o atacou. Caso este não tenha uma carta da mesma categoria, ou não queira usar sua carta de mitigação, ele poderá utilizar seus conhecimentos para responder com uma mitigação adequada ao risco apontado, em no máximo 1 minuto. O jogador atacante deverá ouvir as mitigações verbalizadas pelo jogador em defesa e indicar se ele acertou alguma das mitigações descritas na carta de risco sendo usada no ataque. Ao final de um minuto, se o jogador na defesa não tiver acertado nenhuma das possíveis mitigações listadas, o jogador atacante deverá ler todos as possíveis mitigações para o risco indicado. Esta ação é feita para que os jogadores ouçam as possíveis mitigações para o risco em questão, além de permitir que haja uma discussão quanto ao que o jogador na defesa sugeriu como mitigações.

Vale ressaltar que caso o jogador atacado fale uma mitigação usando palavras diferentes, o mesmo poderá pedir que sua resposta seja considerada. Se o jogador na defesa não conseguir mitigar o risco que atacou seu projeto, ele deverá pagar 5 unidades de dinheiro ao jogador atacante. O fato de perder dinheiro se o jogador atacado se defender corretamente foi pensado para que haja uma estratégia e balanceamento nos ataques por parte dos jogadores.

Todas as cartas usadas nas ações são descartadas para o deck da mesa e, no começo de cada rodada, todos os participantes recebem uma carta de risco. Essa dinâmica é repetida durante as 4 rodadas e ganha o jogador que no final das rodadas possuir mais unidades de dinheiro. Além disso, caso um jogador perca todo seu dinheiro, ele poderá ficar fora do jogo. No entanto, ainda há um elemento de desafio que permite que o jogador volte a jogar. O desafio consiste em usar uma carta de risco do deck da mesa contra o jogador, para que seja verbalizada pelo menos uma mitigação, em no máximo 1 minuto. Se o jogador obtiver sucesso ele irá ganhar 10 unidades de dinheiro, se não, será eliminado. Este mesmo desafio é utilizado em caso de empate na última rodada do jogo. Neste caso, o desafio será usar uma carta de risco do deck da mesa para que cada um dos jogadores empatados responda alternadamente uma mitigação adequada, o jogador que acertar mais riscos irá ganhar o jogo.



Figura 3: Exemplo de processo de ataque e defesa do ponto de vista do atacante e a defesa.

Fonte: Elaborado pelo autor.

5 AVALIAÇÃO

Para fazer coleta e análise dos dados referentes às percepções dos estudantes com o jogo Arriscando, foi necessário avaliar o mesmo por meio de um instrumento consolidado. Nesse contexto, foi usado o MEEGA+, um modelo de avaliação sistemático de jogos sérios. Além disso, também foi utilizada a dinâmica de *Focus Group: Lovers VS Haters* para coleta de dados qualitativos. As metodologias utilizadas na avaliação do jogo são descritas a seguir.

5.1 MEEGA+

O MEEGA+ é um modelo sistemático para analisar jogos educacionais (digitais e não digitais), a fim de avaliar sua qualidade percebida a partir da perspectiva dos alunos no contexto da educação (PETRI; VON WANGENHEIM; BORGATTO, 2017). Os fatores de qualidade do MEEGA+ são usabilidade e experiência do jogador. As dimensões relacionadas com a experiência do jogador são: foco e atenção, diversão, desafio, interação social, confiança, relevância, satisfação, aprendizado percebido; já as dimensões relacionadas com usabilidade são: capacidade de aprendizado, operacionalidade, estética, acessibilidade e proteção contra erros do usuário.

Para usar o MEEGA+, um questionário, baseado nas dimensões definidas no modelo de avaliação, é usado de modo a operacionalizar a coleta de dados. As respostas do questionário são fornecidas através de uma escala Likert de 5 pontos, variando de discordo totalmente e concordo totalmente, permitindo que o participante da avaliação expresse sua percepção sobre o objeto de estudo com mais precisão ao ter a possibilidade de responder de forma neutra e aumentar a qualidade da resposta (DEVELLIS, 2016). Os dados coletados são analisados em termos de distribuição de frequência e tendência central para cada fator de qualidade e suas dimensões (PETRI; VON WANGENHEIM; BORGATTO, 2017).

5.2 *Focus group: Lovers VS Haters*

Focus Group (ou Grupo de Foco) é uma técnica em que um grupo de pessoas é reunido e questionados sobre suas percepções em relação a um produto, serviço, conceito ou ideia (FRANÇA *et al.*, 2015). Assistidos pelos facilitadores, os participantes são convidados a falar com os outros participantes, experimentar, opinar, debater e se posicionar em relação ao tema da pesquisa. O modelo da dinâmica de *Focus Group* utilizado na avaliação do jogo Arriscando

seguiu a metodologia adotada por França *et al.* (2015). Esta metodologia é dividida em duas partes: a primeira é uma atividade de interpretação de papéis, em que o grupo é dividido em dois: “*Lovers*”, que deverão argumentar a favor do objeto de estudo e, “*Haters*”, que terão que construir argumentos contra o objeto de estudo. O segundo momento do *Focus Group* não é mais uma atividade de interpretação de papéis, e sim um momento em que todos terão a liberdade para discutir os pontos levantados durante o levantamento de argumentos e sugerir melhorias para o objeto de estudo.

Na primeira atividade do *Focus Group: Lovers VS Haters*, pode ser adotado o uso de quadros com espaços específicos para cada tipo de papel interpretado, em que os participantes são encorajados a usar cartões de cores diferentes para preencher estes espaços. Para o papel de *Lover*, os participantes devem pensar em aspectos que façam o objeto de estudo: ser útil, ser fácil de usar e ser adequado para o objetivo. Para o papel de *Hater*, os participantes devem pensar nos aspectos que fazem o objeto de estudo ser o contrário.

Para a aplicação da dinâmica, primeiro deve ser explicado a todo o grupo como funcionarão as atividades e suas ferramentas. Após, os participantes devem ser divididos em grupos de *Lovers* e *Haters*, que por sua vez irão discutir e elaborar os argumentos de acordo com o papel que receberam. No fim do tempo determinado pelos moderadores, ou quando os grupos já tiverem finalizado a elaboração dos argumentos e posicionado os cartões nas seções corretas do quadro, deve ser iniciada a discussão. A argumentação deve seguir o seguinte fluxo: um grupo expressa seu argumento, o grupo contrário contra-argumenta, e a discussão segue até que os dois grupos concordarem com pontos em comum dos argumentos levantados. O debate deve seguir até o fim dos argumentos das duas equipes e deve sempre ser mediado, para que seja organizada a discussão e os argumentos apresentados.

Os dados obtidos a partir da dinâmica utilizada são os cartões elaborados pelas equipes, assim como áudios do debate realizado e anotações dos moderadores que estiveram presentes durante a atividade. Esses dados podem ser analisados para mostrar diretamente pontos positivos, negativos e melhorias do objeto de estudo na percepção individual e coletiva dos participantes em relação a utilidade, facilidade de uso e percepção sobre a adequação do objeto de estudo para atingir o objetivo estipulado.

5.3 Aplicação do jogo

Para fazer a aplicação do jogo Arriscando, foram impressos 4 kits do jogo. Foi necessário um dado de 6 faces para sortear o participante que iria começar a partida (também

foi usado um aplicativo de sorteios para dispositivos móveis). Os materiais foram preparados para serem usados em turmas de até 20 pessoas, em que cada um dos kits do jogo seria usado por grupos de até 5 pessoas.

Foi preparada uma adaptação do modelo de avaliação da qualidade de jogos educacionais (MEEGA+) para que os participantes pudessem avaliar os aspectos educacionais e lúdicos do jogo. O questionário foi criado através da plataforma de formulários online e gratuita *Google Forms*. Além disso, foi preparada a dinâmica de *Focus Group: Lovers VS Haters*, os quadros e cartões com as frases necessárias para que a dinâmica fosse aplicada, extrair os pontos positivos e negativos do jogo e ouvir melhorias.

O jogo Arriscando foi aplicado nas turmas de Engenharia de *Software* I (disciplina introdutória de Engenharia de *Software*) e Engenharia de Requisitos do curso de Ciência da Computação e na turma de Engenharia de *Software* I do curso de Engenharia da Computação, ambos da Universidade Federal do Maranhão. A aplicação foi feita separadamente em cada turma, utilizando dois horários de aula, totalizando 100 minutos. Os alunos foram previamente informados sobre a aplicação do jogo, assinaram um termo de consentimento para utilização dos dados coletados e foram incentivados a participar através de bonificação de 0,5 pontos em uma nota da disciplina para os jogadores vencedores.

Nos momentos de aplicação estavam presentes o professor e alunos do grupo de pesquisa de Engenharia de *Software* da UFMA, desempenhando o papel de moderadores. Nos 10 primeiros minutos da aplicação, o professor apresentou uma breve introdução sobre gerenciamento de riscos em projetos de *software*, questionando e apresentando conceitos sobre riscos e mitigações. Depois, foram falados os conceitos e as regras do jogo, em 20 minutos.

Durante a aplicação do jogo, foram formados grupos de 4 ou 5 jogadores e distribuído um kit do jogo para cada grupo para que, com ajuda do professor ou de algum moderador, os elementos do jogo fossem distribuídos de acordo com as regras. Para reforçar as regras e introduzir a dinâmica do jogo, foi feita uma partida de experimentação com somente 2 rodadas, no intervalo de 20 minutos. Após a partida de experimentação o jogo foi recomeçado para que uma partida completa fosse jogada, com 4 rodadas durante 40 minutos, e os alunos pudessem competir pela bonificação. A Figura 4 apresenta uma equipe de 4 jogadores da disciplina de Engenharia de *Software* I em uma das rodadas do jogo.



Figura 4: Alunos jogando o Arriscando em sala de aula.

Fonte: Elaborado pelo autor.

O professor e os moderadores estavam sempre presentes e disponíveis para orientar e tirar as dúvidas dos participantes em relação às regras, à dinâmica e o conteúdo do jogo. A maioria dos grupos concluiu as 4 rodadas no tempo de 40 minutos. Porém, houve grupos que concluíram a partida em, no máximo, 10 minutos a mais ou 10 minutos a menos. No fim do horário da aula, ou quando todos os grupos tivessem concluído a partida, o professor usou os 10 minutos finais para falar sobre o impacto dos riscos em projetos de desenvolvimento de *software*, a importância de mitigá-los e fazer o encerramento da aplicação do jogo.

A avaliação do jogo foi feita na aula posterior a aplicação do jogo, novamente utilizando dois horários de aula, totalizando 100 minutos. Os alunos foram previamente informados sobre a dinâmica de avaliação que seria feita e receberam o link para o questionário online via sistema acadêmico, para que o formulário fosse respondido com antecedência. Todos os alunos que participaram da aplicação do jogo e participaram da dinâmica de avaliação receberam 0,5 ponto em uma das notas da disciplina como forma de incentivo. Para evitar que os alunos falassem apenas bem do jogo devido ao bônus recebido pela participação, foi decidido utilizar a dinâmica de *Lovers VS Haters* (ver Seção 5.2).

Primeiramente, foi dado à turma um total de 20 minutos para que o questionário fosse respondido por quem não o tivesse feito previamente. A seguir, foi explicado para a turma como funcionaria a dinâmica de avaliação e apresentados os elementos que seriam usados. Com isso, a turma foi dividida em dois grupos, os que interpretariam os *Lovers* e os que interpretariam os

Haters e foram distribuídos os cartões contendo as afirmativas de cada papel definido. Os *Lovers* deveriam escrever afirmações nos cartões para preencher os quadros com as seguintes perguntas: “O jogo Arriscando é aplicável para o ensino de gerenciamento de riscos porque...”, “O jogo Arriscando é útil porque...” e “O jogo Arriscando é fácil de usar porque...”. Os *Haters* deveriam escrever afirmações nos cartões para as seguintes perguntas: “O jogo Arriscando não é aplicável para o ensino de gerenciamento de riscos porque...”, “O jogo Arriscando não é útil porque...” e “O jogo Arriscando não é fácil de usar porque...”. Foi destinado um total de 30 minutos para a discussão das afirmativas e a sua escrita.

Ao final do tempo, as equipes deveriam confrontar, argumento por argumento, até chegarem em concordância com os pontos de vista apresentados, discutindo todos os argumentos previamente preenchidos nos cartões de cada equipe, em um tempo total máximo de 40 minutos. Os 20 minutos finais da dinâmica de avaliação foram usados para ouvir a turma como um todo, e sem os papéis de *Lovers* ou *Haters*, sobre sugestões e melhorias em relação ao jogo e encerrar a dinâmica e a avaliação do jogo. A seguir são apresentados os resultados da avaliação através do questionário MEEGA+ e a dinâmica de *Lovers VS Haters*.

5.4 Resultados da aplicação do MEEGA+ e questões abertas

O questionário de avaliação foi respondido por 31 pessoas, sendo 25 homens e 6 mulheres, na faixa etária de 18 a 28 anos. Os resultados são apresentados conforme o modelo de avaliação MEEGA+, que categoriza a avaliação nos fatores de qualidade Experiência do Jogador e Usabilidade. As Figuras 5 e 6 apresentam os resultados da avaliação.

No fator de qualidade Experiência do Jogador, sobre a afirmação “A organização do conteúdo me ajudou a estar confiante de que eu iria aprender com este jogo”, da dimensão Confiança, 72% dos alunos concordaram, demonstrando que o jogo passa segurança sobre o conteúdo abordado. Na dimensão de Desafio, a afirmação “Este jogo é adequadamente desafiador para mim”, em que nenhum dos alunos discordaram, contrasta com as duas outras afirmações da mesma dimensão “O jogo oferece novos desafios (oferece novos obstáculos, situações ou variações) com um ritmo adequado” e “O jogo não se torna monótono nas suas tarefas (repetitivo ou com tarefas chatas)” em que houve uma discordância de 10% e 29% respectivamente. Isto pode ter sido ocasionado pelo fato de o jogo seguir uma dinâmica comum de jogos de cartas.

Em relação a dimensão de Satisfação, houve um máximo de 6% de discordância, e as afirmações “Completar as tarefas do jogo me deu um sentimento de realização” e “É devido ao

meu esforço pessoal que eu consigo avançar no jogo” tiveram 23% e 29% de avaliações neutras, respectivamente, mostrando que em geral os alunos se sentiram satisfeitos com o jogo.

As dimensões Interação Social e Diversão foram as mais bem avaliadas, em que todas as afirmações tiveram mais de 90% de concordância. A afirmação “Eu esqueci sobre o ambiente ao meu redor enquanto jogava este jogo” da dimensão Atenção Focada obteve a maior porcentagem de neutralidade (39%), que pode ser justificado pela falta de imersão que o jogo proporciona.

No que diz respeito à dimensão Relevância, as afirmações “É claro para mim como o conteúdo do jogo está relacionado com a disciplina” e “O jogo é um método de ensino adequado para esta disciplina”, mostra que os alunos captaram e aceitaram com facilidade os conceitos apresentados no jogo.

Por fim, na dimensão Percepção de Aprendizagem, os itens foram preparados seguindo as sugestões de Savi et al. (2011), de acordo com os objetivos do jogo no ensino de categorias de riscos, riscos, mitigações e seu relacionamento. Nesse contexto, as afirmações “O jogo contribuiu para conhecer os diferentes riscos que acontecem em projetos de desenvolvimento de *software*” e “O jogo contribuiu para conhecer as diferentes mitigações que podem ser aplicadas em projetos de desenvolvimento de *software*” obtiveram 90% de concordância. Já a afirmação “O jogo foi eficiente para minha aprendizagem, em comparação com outras atividades da disciplina” apresenta 32% de respostas neutras e 6% de respostas discordantes, evidenciando que há alunos que preferem outras atividades, fora o jogo, como forma de ensino.

No fator de qualidade Usabilidade, na dimensão Estética e na dimensão Acessibilidade houve aproximadamente 10% de discordâncias, além de 35% e 29%, respectivamente, de respostas neutras sobre as afirmações “O design do jogo é atraente (tabuleiro, cartas, interface, gráficos, etc)” e “Os textos, cores e fontes combinam e são consistentes”, evidenciando que o jogo pode melhorar em relação ao consistência e acessibilidade do design.

A afirmação “Aprender a jogar este jogo foi fácil para mim” obteve 80% de concordância; a afirmação “Eu acho que a maioria das pessoas aprenderiam a jogar este jogo rapidamente” obteve 64% de concordância; e a afirmação “Eu considero que o jogo é fácil de jogar” da dimensão Operabilidade obteve 67% de concordância. Estas afirmativas mostram que, na sua maioria, os alunos aprovaram o jogo, mas que também pode ser difícil de aprendê-lo.

As questões abertas do formulário foram usadas para entender melhor a percepção dos alunos com relação ao conteúdo e a facilidade de uso do jogo, além de coletar pontos positivos, negativos e oportunidades de melhorias. Na pergunta “Você sente que aprendeu mais sobre

gerenciamento de riscos jogando o "Arriscando"? Por quê?", 90% dos alunos responderam que sim e os 10% restantes responderam que aprenderam pouco, porém se tivesse mais tempo disponível poderiam ter aprendido mais. Além disso, alguns pontos positivos apontados pelos alunos foram: a dinâmica usada no jogo, o fato de ser um *card game*, a competitividade gerada, a interação social e o aprendizado através do divertimento.

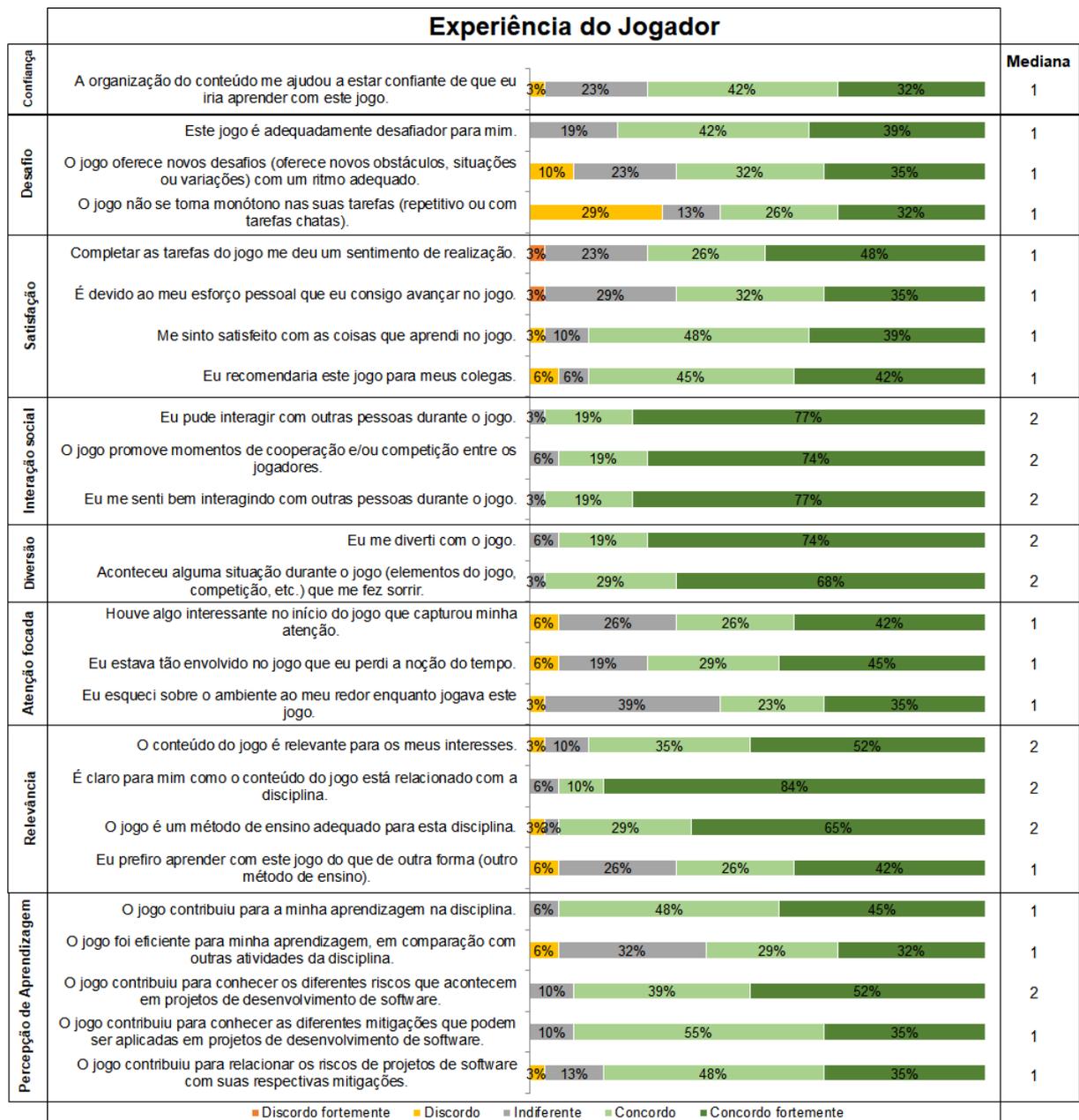


Figura 5: Resultados do MEEGA+ Experiência do Jogador.

Fonte: Elaborado pelo autor.

Os pontos negativos apontados foram: a pouca quantidade de cartas, consequentemente de conteúdo (categorias, riscos e mitigações) gerando uma repetição constante, as cartas terem

pouco apelo visual e do das relações riscos-mitigações desbalanceadas dentre o jogo. Vale ressaltar que embora o total de itens tenha sido considerada pequena, os mesmos vieram da literatura e manteve-se a ideia de que uma mitigação poderia ser usada em diferentes riscos. Finalmente, com relação aos pontos de melhoria indicados no questionário, foram apontados aumentar o número de cartas, diminuir o número de mitigações por carta e melhorar o *design*.

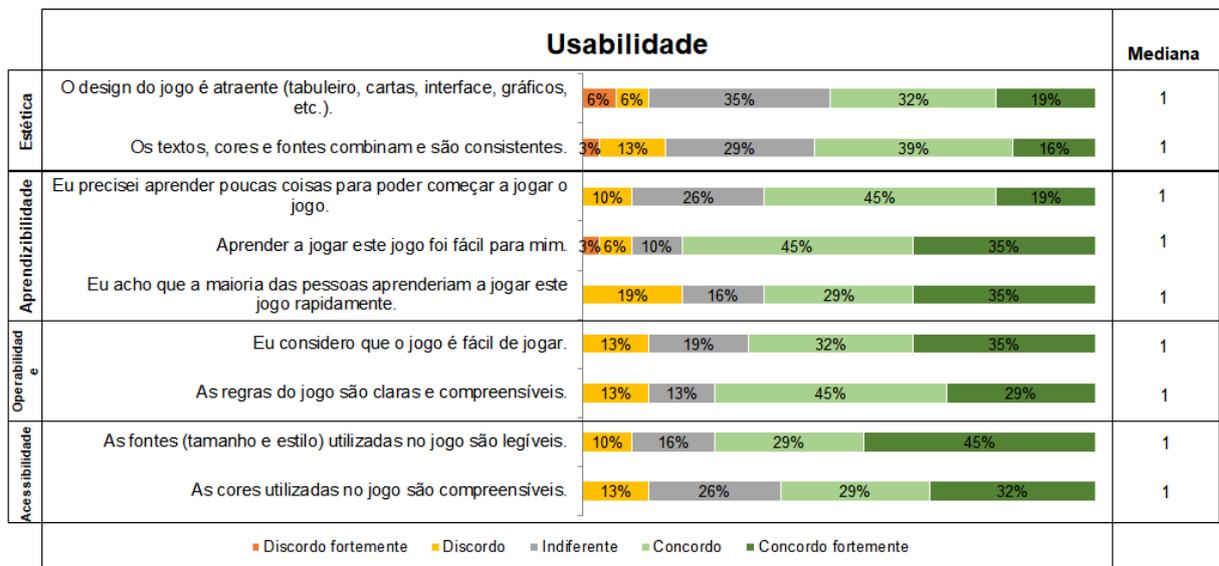


Figura 6: Resultados do MEEGA+ usabilidade.

Fonte: Elaborado pelo autor.

5.5 Resultados da aplicação do *Focus Group – Lovers VS Haters*

Com relação à dinâmica de *Focus Group – Lovers VS Haters*, a mesma resultou em 25 cartões com argumentos positivos, dos *Lovers*, e 18 cartões com argumentos negativos, dos *Haters*. No quadro que afirmava que o jogo é aplicável para o ensino de gerenciamento de riscos foram elaborados cartões com argumentos que diziam que a aula se tornou mais divertida, foi obtido um maior dinamismo no aprendizado e é utilizado métodos em que as pessoas possuem familiaridade. Em contrapartida, os *Haters* afirmaram que o jogo não incentiva o aprendizado, mas somente a leitura do conteúdo e que a competição pode desviar o objetivo do jogo, para somente buscar a vitória. Após a discussão, os alunos entraram em consenso quando às seguintes afirmativas: o jogo ajuda no aprendizado, torna a aula dinâmica e divertida e a competição incentiva a captação do conhecimento. Porém, a maioria dos alunos também concordou que havia vários conteúdos no jogo que tornaram as partidas previsíveis e em certos pontos desestimulantes. Estes aspectos negativos sugerem que embora haja um consenso

quanto ao impacto positivo no aprendizado, ainda existem oportunidades de melhora na mecânica do jogo para que os alunos foquem em ganhar conhecimento no lugar de vencer.

A argumentação sobre a utilidade do jogo foi a que obteve mais argumentos, como: incentivar a interação entre os alunos enquanto se joga e aprende, criando um ambiente agradável em que é incentivada a competição saudável, que não obteve contra-argumentos. Outro argumento bastante expressivo foi o de que é possível conhecer e vivenciar situações e cenários em que o gerenciamento é aplicável, fazendo os alunos pensarem de maneira crítica em soluções para os riscos. Os argumentos para que o jogo não possa ser considerado útil foram que: é possível alcançar a vitória decorando as respostas facilmente devido a repetição constante do conteúdo e que a aplicação é complexa e demanda recursos. Estes argumentos foram debatidos, chegando à conclusão que há uma complexidade na explicação da dinâmica do jogo, mas ao jogar as primeiras rodadas o jogo é facilmente entendido. Também foi discutido que repetição era boa no longo prazo, visto que, no mínimo, o aluno já lembraria de alguns riscos e mitigações.

Por fim, o pouco apelo visual dos elementos do jogo, o formato da apresentação das regras e a linguagem técnica usada nas cartas fizeram com que o jogo não fosse fácil de usar, argumentos que foram discutidos, principalmente quando os *Lovers* argumentaram que o jogo e sua linguagem é simples e clara, possui uma curva de aprendizagem baixa, não requer muito tempo para jogar e não é necessário ter conhecimento prévio sobre gerenciamento de riscos para jogar. Após as discussões, as principais conclusões foram que o jogo é simples e fácil de jogar, que a linguagem é bastante técnica, mas que é possível entender sem possuir conhecimento prévio de gerenciamento de riscos.

Após o momento de argumentação e discussão, foram ouvidas as sugestões dos alunos para a melhoria do jogo. As melhorias citadas foram: aumentar o número de cartas, conseqüentemente, aumentando a variedade de riscos e mitigações, fazer com que as cartas de risco tenham quantidades iguais ou parecida de mitigações, ou, no caso de haver uma grande diferença de quantidade de mitigações entre as cartas de risco, as cartas que tiverem menos mitigações valerem mais dinheiro e as que tiverem menos valer menos. Além disso, foi apontada a necessidade de aumentar o apelo visual e a qualidade dos materiais, como o uso de papel cartão, plastificação das cartas, aumentar o uso de cores e figuras. Também foi falado sobre a criação de cartas de regras, para que fossem usadas como lembrete durante o jogo.

5.6 Ameaças à validade

Foram consideradas ameaças à validade identificadas na literatura de engenharia de *software* experimental e na aplicação de jogos educacionais para o ensino de qualidade de *software* (MOREIRA; DOS SANTOS MARQUES, 2018), (MOURA; SANTOS, 2018) e (WOHLIN *et al.*, 2012). Nesse contexto, uma ameaça pode ser o público com o que o jogo foi testado. No entanto, os participantes do estudo são estudantes de cursos de graduação com pouca, ou nenhuma, experiência em metodologias de gerenciamento de riscos e sem contato prévio com o jogo, tornando-os representativos para a avaliação da aprendizagem com o jogo. Além disso, o número limitado de alunos impacta na generalização dos resultados. Mesmo assim, os resultados da aplicabilidade do jogo em três turmas de disciplinas introdutórias (ou não) de engenharia de *software* são indícios da viabilidade do jogo para ensino de gerenciamento de riscos. Desta forma, para aumentar a confiabilidade dos resultados, o jogo poderia ser testado em outras instituições de ensino superior, sendo necessária a condução de novos estudos.

Com relação às categorias de riscos, riscos e mitigações, o material do jogo foi traduzido e revisado para evitar inconsistências com o material original. Porém, informações foram adicionadas nos itens para que fossem melhor compreendidos dentro do contexto do jogo. Estas informações foram verificadas pelos professores das disciplinas de engenharia de *software*, antes da aplicação do jogo para corrigir qualquer inconsistência ou ambiguidade. Além disso, todo o conteúdo do jogo foi baseado em publicações relacionadas ao gerenciamento de riscos em projetos de *software*.

Finalmente, os instrumentos de coleta foram utilizados conforme o MEEGA+ propõe, sendo realizadas apenas adaptações nos itens que avaliam os objetivos de aprendizagem. O MEEGA+ possui métricas para avaliar tanto a aprendizagem do conteúdo abordado no jogo como a experiência dos jogadores. Além disso, as respostas associadas à dimensão Percepção de Aprendizagem, conforme previsto na metodologia utilizada, são informadas pelos próprios participantes. A atividade de *Lovers VS Haters* aplicada permitiu a discussão e elaboração de argumentos concisos, tanto positivos quanto negativos. Por ser um dinâmica de *Focus Group*, alguns alunos não participaram efetivamente, omitindo sua opinião individual em face ao grupo. Nesse contexto, questões abertas adicionais foram acrescentadas aos questionários para permitir que os mesmos pudessem expressar sua opinião, caso o quisessem.

6 CONCLUSÃO

Este trabalho apresentou o jogo Arriscando, um jogo para o ensino de gerenciamento de riscos em projetos de desenvolvimento de *software*. Desenvolvido utilizando a metodologia colaborativa e iterativa *Design Thinking*, o jogo contou com o envolvimento de um profissional e docente na área da computação e estudantes de computação e *design* na sua concepção. O objetivo do jogo é ensinar os conceitos iniciais de gerenciamento de riscos em projetos de desenvolvimento de *software* de forma interativa e lúdica. Além de alcançar os objetivos de aprendizagem de compreensão, análise e associação dos diferentes tipos de categorias de riscos, riscos e mitigações.

Para validação da proposta, o jogo foi aplicado em 3 turmas de disciplinas de Engenharia de *Software* onde o tópico de gerenciamento de riscos é apresentado e discutido. Destaca-se a utilização de dois métodos de avaliação (MEEGA+ e *Focus Group – Lovers VS Haters*), cujos resultados individuais foram relacionados. A partir dos resultados apresentados, é possível concluir que o jogo Arriscando foi bem aceito pelos alunos, em que todos os itens do modelo de avaliação tiveram mediana 1 ou 2, indicando respostas positivas. As dimensões de Diversão e Interação Social, do MEEGA+, percebidas pelos participantes não obtiveram nenhuma resposta discordante com as afirmações. Como pontos negativos foi destacado a pouca variedade de cartas no jogo, gerando uma repetitividade do conteúdo, a falta de material de apoio e um *design* pouco apelativo.

Por estar presente na maioria das grades curriculares das disciplinas introdutórias de Engenharia de *Software* dos cursos de computação do país, o gerenciamento de riscos se mostra como uma das áreas da qualidade de *software* de grande importância para o crescimento acadêmico e para a garantia do sucesso de projetos de *software*. Assim, evidenciamos a necessidade de criação de novas estratégias de ensino de gerenciamento de riscos em que é possível disseminar conhecimento especializado e aplicável ao mundo real de forma dinâmica e divertida. Espera-se que o jogo Arriscando se torne uma ferramenta para o ensino desse tópico essencial para o gerenciamento de projetos; e que, a partir do seu uso, seja possível disseminar e fixar o conhecimento de forma dinâmica e diferenciada.

As principais contribuições desse trabalho foram a caracterização do estado da arte dos jogos educacionais desenvolvidos para o ensino da Engenharia de *Software* no Brasil, através de um Mapeamento Sistemático da Literatura, que foi utilizado nesse trabalho. O desenvolvimento de um jogo para o ensino de gerenciamento de riscos em projetos de *software*, área de interesse da Engenharia de *Software* e de extrema importância para a garantia da

qualidade de projetos. A disseminação do trabalho desenvolvido através da publicação e apresentação do artigo “*Risking: A Game for Teaching Risk Management in Software Projects*” no *XVIII Brazilian Symposium on Software Quality (SBQS'19)*.

Como trabalhos futuros, pretende-se aplicar as sugestões de melhorias no jogo Arriscando, para desenvolver uma nova versão mais completa e robusta. Primeiramente adicionando mais conteúdo ao jogo, para diminuir a repetição dos itens e elaborar o balanceamento entre as cartas e seus relacionamentos para que corresponda a impactos do mundo real. Além disso, novas versões do jogo podem ser feitas considerando riscos e mitigações específicos de projetos de *software* de contextos distintos.

REFERÊNCIAS

- ADAMS, N. E. **Bloom's taxonomy of cognitive learning objectives**. Journal of the Medical Library Association: JMLA, v. 103, n. 3, p. 152, 2015.
- ARAÚJO, N. et al. **Avaliando a Viabilidade do BlackBox em Sala de Aula: Um Jogo Sério para Ensino de Teste Funcional de Software**. Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). [S.l.]: [s.n.]. 2017. p. 817.
- BATTISTELLA, P. E.; WANGENHEIM, C. G. V.; FERNANDES, J. M. **Como jogos educacionais são desenvolvidos? Uma revisão sistemática da literatura**. XXII Workshop sobre Educação em Computação (WEI 2014). [S.l.]: Sociedade Brasileira de Computação. 2014.
- BATTISTELLA, P. E. A. V. W. C. G. **Engaged: Um processo de desenvolvimento de jogos para ensinar computação**. Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE), v. 27, n. 1, 2016.
- BATTISTELLA, P. E.; DE CAMARGO, A. S.; VON WANGENHEIM, C. G. **Scrum-scape: Jogo educacional de role-playing game (rpg) para ensinar scrum**. Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). [S.l.]: [s.n.]. 2016. p. 330.
- BEPPE, T. A. et al. **GreaTest: A Card Game to Motivate the Software Testing Learning**. Proceedings of the XXXII Brazilian Symposium on Software Engineering. [S.l.]: ACM. 2018. p. 298-307.
- BOEHM, B. W. **Software risk management: principles and practices**. IEEE software, v. 8, n. 1, p. 32-41, 1991.
- BRITO, A.; VIEIRA, J. **'2TScrum': A Board Game to Teach Scrum**. Proceedings of the 31st Brazilian Symposium on Software Engineering. [S.l.]: ACM. 2017. p. 279-288.
- CABREJOS, L. J. E. R.; VIANA, D.; DOS SANTOS, R. P. **Planejamento e Execução de Estudos Secundários em Informática na Educação: Um Guia Prático Baseado em Experiências**. Jornada de Atualização em Informática na Educação, v. 7, n. 1, p. 21-52, 2018.
- CAMPOS, A. M. et al. **Um jogo voltado à prática de gerenciamento de projetos**. Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). [S.l.]: [s.n.]. 2012.
- CAMPOS, B. et al. **Experiência de Projeto e Desenvolvimento de Jogo para Ensino de Engenharia de Requisitos para Sistemas Ubíquos**. Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). [S.l.]: [s.n.]. 2012.
- CASCINI, D.; CAMPOS, S. **Avaliação de jogos educacionais multiusuários: Uma revisão sistemática da literatura**, v. 26, n. 1, p. 519, 2015.

- DAMIAN, D.; HADWIN, A.; AL-ANI, B. **Instructional design and assessment strategies for teaching global software development: a framework**. Proceedings of the 28th international conference on Software engineering, p. 685-690, 2006.
- DE CASTRO, R. M. et al. **AGILITY SCRUM-Um Jogo para Ensino da Metodologia SCRUM**. Workshop sobre Educação em Computação (WEI 2017). [S.l.]: SBC. 2017.
- DE MEDEIROS, R. A. et al. **GameES: Um Jogo para a Aprendizagem de Engenharia de Software**. Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE), v. 24, n. 1, p. 945, 2013.
- DETROZ, J. P.; HINZ, M.; DA SILVA HOUNSELL, M. **Uso de Pesquisa Bibliográfica em Informática na Educação: um Mapeamento Sistemático**. Revista Brasileira de Informática na Educação. [S.l.]: [s.n.]. 2015.
- DEVELLIS, R. F. **Scale development: Theory and applications**. [S.l.]: Sage publications, v. 26, 2016.
- DINIZ, L. L.; DAZZI, R. L. **Jogo para o apoio ao ensino do teste de caixa-preta**. Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). [S.l.]: [s.n.]. 2012.
- DOS SANTOS, H. M. et al. **CleanGame: Gamifying the Identification of Code Smells**. XXXIII Brazilian Symposium on Software Engineering. [S.l.]: ACM. 2019. p. 437-446.
- EDUARDO BATTISTELLA, P.; GRESSE VON WANGENHEIM, C. **Games for Teaching Computing in Higher Education – A Systematic Review**. IEEE Technology and Engineering Education, v. 1, p. 8-30, mar. 2016.
- FEITOSA, A. C.; CAMPOS, G. M. **AprendES: um jogo educacional para auxiliar o processo de ensino-aprendizagem da Engenharia de Software**. Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). [S.l.]: [s.n.]. 2010.
- FERREIRA, T. D. S. D. et al. **AdventureSECO: Jogo Educacional para o Ensino de Conceitos sobre Ecossistemas de Software**. Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE). [S.l.]: [s.n.]. 2018. p. 178-179.
- FOLHA DE SÃO PAULO. Computação - Ranking de Cursos. **Ranking Universitário Folha**, 2018. Disponível em: <<https://ruf.folha.uol.com.br/2018/ranking-de-cursos/computacao/>>. Acesso em: 5 dez. 2019.
- FRANÇA, B. B. et al. **Using Focus Group in Software Engineering: lessons learned on characterizing software technologies in academia and industry**. CIBSE 2015 - XVIII Ibero-American Conference on Software Engineering, Março 2015.
- FULLERTON, T. **Game design workshop: a playcentric approach to creating innovative games**. [S.l.]: AK Peters/CRC Press, 2018.

GONÇALVES, R. Q.; THIRY, M.; ZOUCAS, A. **Avaliação da aprendizagem em experimentos com jogo educativo de engenharia de requisitos**. X Simpósio Brasileiro de Qualidade de Software (SBQS). Curitiba: [s.n.]. 2011.

HASSO PLATTNER INSTITUTE OF DESIGN AT STANFORD. **Design Thinking Bootleg. d.School**, 2019. Disponível em: <<https://dschool.stanford.edu/resources/design-thinking-bootleg>>. Acesso em: 28 nov. 2019.

HUANG, S.; DISTANTE, D. **On Practice-Oriented Software Engineering Education**. 19th Conference on Software Engineering Education and Training Workshops (CSEETW'06), p. 15-15, 2006.

JARAMILLO, C. M. Z.; ALVAREZ, M. C. G.; GONZALEZ-CALDERON, G. **Riskware: a game for teaching software project risk management**. Developments in Business Simulation and Experiential Learning: Proceedings of the Annual ABSEL conference. [S.l.]: [s.n.]. 2013.

KOHWALTER, T. C.; CLUA, E. W.; MURTA, L. G. **SDM-An educational game for software engineering**. 2011 Brazilian Symposium on Games and Digital Entertainment. [S.l.]: IEEE. 2011. p. 222-231.

LEAVITT, H. J. **Applied organization change in industry: structural, technical and human approaches**, 1964.

LYYTINEN, K.; MATHIASSEN, L.; ROPPONEN, J. **Attention shaping and software risk—a categorical analysis of four classical risk management approaches**. Information Systems Research, v. 9, n. 3, p. 233-255, 1998.

MENDES, J. et al. **Identificação das Expectativas e Dificuldades de Alunos de Graduação no Ensino de Engenharia de Software**. Anais do XXVII Workshop sobre Educação em Computação. Porto Alegre: Sociedade Brasileira de Computação, Belém, July 2019. 334-347.

MOREIRA, G. G.; DOS SANTOS MARQUES, A. B. **Evaluating the students' experience with the Scrum Card Game: an experience report in a Software Engineering course**. Proceedings of the 17th Brazilian Symposium on Software Quality. [S.l.]: ACM. 2018. p. 344-353.

MOURA, V.; SANTOS, G. **ProcSoft: A Board Game to Teach Software Processes Based on ISO/IEC 29110 Standard**. Proceedings of the 17th Brazilian Symposium on Software Quality, p. 363-372, 2018.

OLIVEIRA, C. D.; CINTRA, M. E.; NETO, F. M. M. **Jogo sério para o ensino da Gestão de Riscos em Projetos de Softwares usando Inteligência Artificial**. RENOTE, v. 11, n. 1, 2013.

PALUDO, L.; RAABE, A. L. A.; BENITTI, F. B. V. **RSKManager - um jogo para apoiar o ensino de gerência de riscos em projetos de software**. RENOTE, v. 11, n. 3, 2013.

- PETRI, G. et al. **Um Quiz Game para a Revisão de Conhecimentos em Gerenciamento de Projetos**. Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE), v. 27, n. 1, p. 320, 2016.
- PETRI, G.; CHIAVEGATTI, N. C. **Um role playing game para o ensino de elicitação e análise de requisitos**. RENOTE, v. 13, n. 1, 2015.
- PETRI, G.; VON WANGENHEIM, C. G.; BORGATTO, A. F. **Evolução de um Modelo de Avaliação de Jogos para o Ensino de Computação**. 25º Workshop sobre Educação em Computação (WEI 2017). [S.l.]: SBC. 2017.
- PLATTNER, H.; MEINEL, C.; LEIFER, L. **Design Thinking: Understand – Improve – Apply**. [S.l.]: Springer Science & Business Media, 2010.
- PRIKLADNICKI, R. et al. **Ensino de engenharia de software: desafios, estratégias de ensino e lições aprendidas**. FEES-Fórum de Educação em Engenharia de Software. [S.l.]: [s.n.]. 2009. p. 1-8.
- PRIKLADNICKI, R.; ROSA, R.; KIELING, E. **Ensino de Gerência de Projetos de Software com o Planager**. Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). [S.l.]: [s.n.]. 2007. p. 11-20.
- QUEIROZ, R.; PINTO, F.; SILVA, P. **IslandTest: jogo educativo para apoiar o processo ensino-aprendizagem de testes de software**. Anais do XXVII Workshop sobre Educação em Computação. [S.l.]: SBC. 2019. p. 533-542.
- ROCHA, R. V. et al. **AIMED: agile, integrative and open method for open educational resources development**. 2017 IEEE 17th International Conference on Advanced Learning Technologies (ICALT). [S.l.]: IEEE. 2017. p. 163-167.
- ROSE, K. H. **A Guide to the Project Management Body of Knowledge (PMBOK Guide)—Fifth Edition**. Project management journal, v. 44, n. 3, p. e1-e1, 2013.
- SANTOS, S. et al. **Risking: A Game for Teaching Risk Management in Software Projects**. Proceedings of the XVIII Brazilian Symposium on Software Quality, Fortaleza, Brazil, 2019. 188-197. Disponível em: <<http://doi.acm.org/10.1145/3364641.3364662>>.
- SARINHO, V. T. **Masters of the Process: A Board Game Proposal for Teaching Software Management and Software Development Process**. Proceedings of the XXXIII Brazilian Symposium on Software Engineering. [S.l.]: ACM. 2019. p. 532-536.
- SAVI, R.; WANGENHEIM, C.; BORGATTO, A. **Um modelo de avaliação de jogos educacionais na engenharia de software**. Anais do XXV Simpósio Brasileiro de Engenharia de Software (SBES 2011), São Paulo, 2011.
- SCHALLMO, D.; WILLIAMS, C.; LANG, K. **An Integrated Design Thinking Approach—Literature Review, Basic Principles and Roadmap for Design Thinking**. ISPIM Innovation Symposium. [S.l.]: [s.n.]. 2018. p. 1-18.

SHRIVASTAVA, S. V.; RATHOD, U. **Categorization of risk factors for distributed agile projects**. Elsevier, v. 58, p. 373-387, 2015.

SILVA, C. et al. **gTest Learning: Um Jogo para Ensino B**

SOMMERVILLE, I. **Engenharia de Software - 9ª Edição**. São Paulo: Pearson Prentice Hall, 2011.

SOUZA, A. et al. **SCRUMI: Um Jogo Virtual Sérió de Tabuleiro para o Ensino do Framework SCRUM**. Anais do XIII Simpósio Brasileiro de Sistemas de Informação. [S.l.]: SBC. 2017. p. 520-526.

SOUZA, M. M. et al. **Sparse: Um ambiente de ensino e aprendizado de engenharia de software baseado em jogos e simulação**. Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). [S.l.]: [s.n.]. 2010.

VALLE, P. H. D.; ROCHA, R. V.; MALDONADO, J. C. **Testing Game: An Educational Game to Support Software Testing Education**. Proceedings of the 31st Brazilian Symposium on Software Engineering. [S.l.]: ACM. 2017. p. 289-298.

VON WANGENHEIM, C. G.; SHULL, F. **To game or not to game?** IEEE software, v. 26, n. 2, p. 92--94, 2009.

WOHLIN, C. et al. **Experimentation in software engineering**. [S.l.]: Springer Science & Business Media, 2012.