



USO DE JOGOS COMO ESTRATÉGIA PARA O ENSINO DE REUTILIZAÇÃO DE SOFTWARE

Diego Cardoso Borda Castro

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientadora: Cláudia Maria Lima Werner

Rio de Janeiro
Março de 2020

USO DE JOGOS COMO ESTRATÉGIA PARA O ENSINO DE REUTILIZAÇÃO DE
SOFTWARE

Diego Cardoso Borda Castro

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO
LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM
CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Orientadora: Cláudia Maria Lima Werner

Aprovada por: Prof. Geraldo Bonorino Xexéo

Prof. Fernando Brito e Abreu

Prof. Marcelo Schots de Oliveira

RIO DE JANEIRO, RJ - BRASIL

MARÇO DE 2020

Castro, Diego Cardoso Borda.

Uso de Jogos como Estratégia para o Ensino de Reutilização de Software / Diego Cardoso Borda Castro. – Rio de Janeiro: UFRJ/COPPE, 2020.

xvi, 138 p.: il.; 29,7 cm.

Orientadora: Cláudia Maria Lima Werner

Dissertação (mestrado) – UFRJ/ COPPE/ Programa de Engenharia de Sistemas e Computação, 2020.

Referências Bibliográficas: p. 121-133.

1. Reutilização de Software. 2. Componentes de Software. 3. Jogos. 4. Game-based learning. I. Werner, Cláudia Maria Lima. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

AGRADECIMENTOS

Aos meus familiares, pela confiança e compreensão, em especial a minha mãe Rosane Cardoso. Obrigado pelos ensinamentos e carinho.

À minha orientadora, Cláudia Werner, pela oportunidade e direcionamento que recebi no mestrado. Obrigado pelos conselhos, confiança depositada ao longo dos trabalhos e ensinamentos que guiaram essa pesquisa. Espero ter contribuído face a confiança em mim depositada.

Aos professores do PESC/COPPE que, com muita dedicação e competência, apoiaram no meu desenvolvimento.

Aos amigos do grupo de Reutilização de Software e do Lab3D da COPPE/UFRJ, pela convivência e amizade.

À CAPES, pelo apoio financeiro durante o mestrado.

A todos aqueles que passaram em minha vida e contribuíram de alguma forma

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

USO DE JOGOS COMO ESTRATÉGIA PARA O ENSINO DE REUTILIZAÇÃO DE SOFTWARE

Diego Cardoso Borda Castro

Março/2020

Orientadora: Cláudia Maria Lima Werner

Programa: Engenharia de Sistemas e Computação

A Reutilização de Software (RS) é uma das disciplinas da Engenharia de Software (ES) e corresponde ao uso de artefatos e conhecimentos de software já existentes para construir algo novo. Uma das principais áreas dessa disciplina é a componentização de software. Apesar das diversas vantagens oferecidas pela RS, a prática sistemática não se tornou padrão nas empresas, sendo a falta de ensino considerada uma das possíveis causas desse problema, com relatos de dificuldades relacionadas a falta de engajamento, motivação e prática do aluno. Para minimizar esses problemas listados, muitos estudiosos estão utilizando jogos como estratégia de ensino, devido às várias vantagens que essa estratégia pode trazer, sendo possível encontrar jogos em diversas áreas da ES. No entanto, o mapeamento que foi realizado nesse trabalho não encontrou nenhum jogo criado com o propósito específico de ensino de RS, sendo este o objetivo deste trabalho. Nesta dissertação, alguns jogos foram criados e avaliados indicando uma satisfação pela maioria dos usuários.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

USE OF GAMES AS A SOFTWARE REUSE TEACHING STRATEGY

Diego Cardoso Borda Castro

March/2020

Advisors: Cláudia Maria Lima Werner

Department: Computer Science and Systems Engineering

Software Reuse (SR) is one of the disciplines of Software Engineering (SE) and corresponds to the use of existing software artifacts and knowledge to build something new. One of the main areas of this discipline is software componentization. Despite the advantages offered by SR, the systematic practice has not become a standard in companies, and the lack of education is considered one of the possible causes of this problem, with reports of difficulties related to the lack of engagement, motivation, and student practice. To minimize these listed problems, many scholars are using games as a teaching strategy, due to the various advantages that this strategy can bring, and it is possible to find games in different areas of SE. However, the mapping that was carried out in this work did not find any game created with the specific purpose of teaching SR, that is the goal of this work. In this dissertation, some games were created and evaluated, indicating satisfaction by the majority of users.

Sumário

1	Introdução	1
1.1	Motivação e Contexto.....	1
1.2	Objetivo	4
1.3	Metodologia.....	5
1.4	Organização do texto	7
2	Fundamentação Teórica.....	8
2.1	Tipos de ensino	8
2.2	Jogos	9
2.2.1	Game-based learning	10
2.2.2	Modelagem de jogos.....	12
2.2.3	Mecânicas para aprendizagem.....	13
2.3	Programação Visual.....	14
2.4	Considerações Finais	15
3	Estudos exploratórios	16
3.1	Organização dos estudos	17
3.2	Protocolo de pesquisa	20
3.2.1	Procedimento	21
3.2.2	Critério de inclusão.....	21
3.2.3	Critério de exclusão	22
3.2.4	Procedimento de extração.....	23
3.3	<i>Strings</i> de busca	23
3.3.1	Uso de jogos para o ensino de reutilização de software	25
3.3.2	Jogos para o ensino de programação com ênfase em reutilização	25

3.3.3	Uso de jogos para o ensino de engenharia de software (mapeamento terciário)	26
3.3.4	Uso de jogos para o ensino de engenharia de software	27
3.3.5	Ensino de reutilização de software	28
3.4	Análise de extração dos artigos	28
3.4.1	Uso de jogos para o ensino de reutilização de software	29
3.4.2	Jogos para o ensino de programação com ênfase em reutilização	30
3.4.3	Uso de jogos para o ensino de engenharia de software (mapeamento terciário)	36
3.4.4	Uso de jogos para o ensino de engenharia de software	38
3.4.5	Uso de jogos para o ensino de reutilização de software	45
3.5	Resumo das descobertas	50
3.5.1	Qual(s) a(s) principal(s) dificuldade(s) do ensino de reutilização de software?	50
3.5.2	Qual(s) a(s) definição(s) de jogos foi utilizada?	53
3.5.3	Qual(s) a(s) principal(s) vantagem(s) / motivação(s) do uso de jogos para o ensino de engenharia de software?	57
3.5.4	Qual(s) a(s) desvantagem(s) do uso de jogos para o ensino de engenharia de software?	64
3.5.5	Qual(s) a(s) principal(s) característica(s) do jogo(s) utilizado(s)?	66
3.5.6	Qual(is) foi(ram) o(s) método(s) de avaliação utilizado(s)?	73
3.5.7	Perguntas extras	75
3.6	Considerações Finais	79
4	<i>Game-Based Learning</i> no ensino de Reutilização de Software	81
4.1	Abordagem inicial	81

4.1.1 Estudo de caso	81
4.1.2 Estudo de viabilidade	86
4.2 Reuse Blocks	89
4.2.1 Contexto	89
4.2.2 Métricas para componentes	90
4.2.3 Descrição do jogo	92
4.2.4 Comandos básicos	95
4.2.5 Exemplo prático.....	96
4.3 Considerações Finais	100
5 Avaliação	102
5.1 Planejamento	102
5.2 Participantes.....	103
5.3 Procedimento	103
5.4 Resultados da avaliação.....	104
5.4.1 Caracterização dos participantes	104
5.4.2 Análise dos resultados	107
5.4.3 Testes estatísticos	110
5.4.4 Considerações Finais	115
6 Conclusão	117
6.1 Contribuições.....	118
6.2 Limitações	119
6.3 Trabalhos Futuros	119
Referências	121
Apêndice A (Questionário MEEGA)	134

Apêndice B (Questionário Educacional)	138
---	-----

Lista de Figuras

Figura 1.1: Metodologia de pesquisa, adaptado de LACERDA <i>et al.</i> (2013).....	5
Figura 2.1: Modelo do aprendizado baseado em jogos, adaptado de PIVÉC <i>et al.</i> (2003) e PERROTTA <i>et al.</i> (2013).....	12
Figura 3.1: Protocolo de pesquisa. Fonte: do autor.	17
Figura 3.2: Sumarização do protocolo de pesquisa baseado no modelo criado por CALDERÓN <i>et al.</i> (2018).....	22
Figura 3.3: Demonstrativo do número de artigos por ano e local de publicação sobre jogos para o ensino de programação com ênfase em reutilização. Fonte: do autor.	34
Figura 3.4: Demonstrativo do número de artigos por ano e local de publicação sobre uso de jogos para o ensino de engenharia de software (mapeamento terciário). Fonte: do autor.	38
Figura 3.5: Demonstrativo do número de artigos por ano e local de publicação sobre uso de jogos para o ensino de engenharia de software. Fonte: do autor.	42
Figura 3.6: Demonstrativo do número de artigos por ano e local de publicação sobre uso de jogos para o ensino de reutilização de software. Fonte: do autor.	48
Figura 3.7: Diagrama de Venn sobre jogos na educação, adaptado de USKOV & SEKAR (2014).	56
Figura 3.8: Sumarização dos achados. Fonte: do autor.	73
Figura 3.9: Principais áreas de ensino da reutilização de software. Fonte: do autor.....	79
Figura 4.1: Jogo Quiz (CASTRO & WERNER, 2019). Fonte: do autor.	83
Figura 4.2: Jogo CodeBoy (CASTRO & WERNER, 2019). Fonte: do autor.	84
Figura 4.3: Avaliação de usabilidade e experiência dos jogos criados, adaptado de PETRI <i>et al.</i> (2019)	88
Figura 4.4: <i>Blueprint</i> do jogo Reuse Block. Fonte: do autor.....	94

Figura 4.5: Reuse Blocks, algoritmo de Fibonacci implementado com FOR. Fonte: do autor.....	97
Figura 4.6: Reuse Blocks, algoritmo de Fibonacci implementado com WHILE. Fonte: do autor.....	97
Figura 4.7: Reuse Blocks, algoritmo de ordenação. Fonte: do autor.....	98
Figura 4.8: Reuse Blocks, algoritmo de retornar maior número. Fonte: do autor.....	99
Figura 4.9: Reuse Blocks, algoritmo de soma de array. Fonte: do autor.....	99
Figura 4.10: Definição de variáveis e valores no Reuse Blocks. Fonte: do autor.....	100
Figura 5.1: Caracterização dos participantes remotos. Fonte: do autor.....	105
Figura 5.2: Caracterização dos participantes observados. Fonte: do autor.....	105
Figura 5.3: Caracterização dos participantes especialistas. Fonte: do autor.....	106
Figura 5.4: Avaliação de experiência, adaptado de PETRI <i>et al.</i> (2019).....	109
Figura 5.5: Avaliação de usabilidade, adaptado de PETRI <i>et al.</i> (2019).....	110
Figura 5.7: Gráficos de normalidade, adaptado de FIELD (2013).....	112
Figura 5.8: Teste de Kruskal-Wallis (BRESLOW, 1970), adaptado de FIELD (2013).	113
Figura 5.9: Teste de Bonferroni (BLAND & ALTMAN, 1995), adaptado de FIELD (2013).....	115

Lista de Tabelas

Tabela 3.1: Modelo do formulário de extração.	23
Tabela 3.2: Tabela de sinônimos e <i>string</i> de busca.	25
Tabela 3.3: <i>String</i> concatenada do uso de jogos para o ensino de reutilização de software.	25
Tabela 3.4: <i>String</i> concatenada do uso de jogos para o ensino de programação com ênfase em reutilização.	26
Tabela 3.5: <i>String</i> concatenada do uso de jogos para o ensino de engenharia de software (mapeamento terciário).	27
Tabela 3.6: <i>String</i> concatenada do uso de jogos para o ensino de engenharia de software.	28
Tabela 3.7: <i>String</i> concatenada do ensino de reutilização de software.	28
Tabela 3.8: Execução do uso de jogos para o ensino de reutilização de software, base Scopus.	29
Tabela 3.9: Execução do uso de jogos para o ensino de reutilização de software, base ScienceDirect.	30
Tabela 3.10: Execução do uso de jogos para o ensino de reutilização de software, base IEEEExplore.	30
Tabela 3.11: Execução do estudo de jogos sobre o ensino de programação com ênfase em reutilização.	31
Tabela 3.12: Execução do estudo de jogos sobre o ensino de programação com ênfase em reutilização, <i>Snowballing backward</i>	31
Tabela 3.13: Execução do estudo de jogos sobre o ensino de programação com ênfase em reutilização, <i>Snowballing forward</i>	32
Tabela 3.14: Conjunto de artigos analisados sobre jogos para o ensino de programação com ênfase em reutilização.	32

Tabela 3.15: Matriz de rastreabilidade sobre jogos para o ensino de programação com ênfase em reutilização.	34
Tabela 3.16: Execução do estudo sobre uso de jogos para o ensino de engenharia de software (mapeamento terciário).	36
Tabela 3.17: Conjunto de artigos analisados sobre uso de jogos para o ensino de engenharia de software (mapeamento terciário).	37
Tabela 3.18: Matriz de rastreabilidade dos artigos sobre uso de jogos para o ensino de engenharia de software (mapeamento terciário).	37
Tabela 3.19: Execução do estudo sobre uso de jogos para o ensino de engenharia de software.	38
Tabela 3.20: Execução do estudo sobre uso de jogos para o ensino de engenharia de software, <i>Snowballing backward</i>	39
Tabela 3.21: Execução do estudo sobre uso de jogos para o ensino de engenharia de software, <i>Snowballing forward</i>	39
Tabela 3.22: Conjunto de artigos analisados sobre uso de jogos para o ensino de engenharia de software.	40
Tabela 3.23: matriz de rastreabilidade sobre uso de jogos para o ensino de engenharia de software.	43
Tabela 3.24: Execução do estudo sobre uso de jogos para o ensino de reutilização de software.	45
Tabela 3.25: Execução do estudo sobre uso de jogos para o ensino de reutilização de software, <i>Snowballing backward</i>	46
Tabela 3.26: Execução do estudo sobre uso de jogos para o ensino de reutilização de software, <i>Snowballing forward</i>	46
Tabela 3.27: Conjunto de artigos analisados sobre uso de jogos para o ensino de reutilização de software.	47

Tabela 3.28: Matriz de rastreabilidade dos artigos sobre o ensino de reutilização de software.	49
Tabela 3.29: Vantagens da utilização dos jogos como técnica de ensino.	59
Tabela 3.30: Características utilizadas nos jogos como técnicas de ensino.	68
Tabela 3.31: Abordagens das avaliações propostas.....	78
Tabela 4.1: Características dos jogos.	82
Tabela 4.2: Tabela de conceituação dos jogos (CASTRO & WERNER, 2019).....	84
Tabela 4.3: Métricas para componentes reutilizáveis.....	91
Tabela 4.4: Tabela de conceituação do jogo Reuse Blocks.....	94
Tabela 4.6: Fases do jogo Reuse Blocks.	96

Lista de Abreviações

GBL	<i>GAME-BASED LEARNING</i>
DGBL	<i>DIGITAL GAME-BASED LEARNING</i>
GDBL	<i>GAME DEVELOPMENT BASED LEARNING</i>
RPG	<i>ROLE PLAYING GAME</i>
ES	ENGENHARIA DE SOFTWARE
RS	REUTILIZAÇÃO DE SOFTWARE
IDE	<i>INTEGRATED DEVELOPMENT ENVIRONMENT</i>
MDA	<i>MECHANICS, DYNAMICS, AESTHETICS</i>
OCR	<i>OBJECTIVES, CHALLENGES, REWARDS</i>

1 Introdução

Este capítulo tem como objetivo apresentar o contexto, a motivação e os problemas abordados nesta dissertação. Além disso, as questões de pesquisa, objetivos e a metodologia adotada para atingi-los também são apresentados, bem como a organização do texto.

1.1 Motivação e Contexto

Muitos alunos com pouco tempo de aula já perderam totalmente a atenção na disciplina que está sendo lecionada. Isso pode ser um reflexo da atual estratégia de ensino sendo fortemente centrado no professor, palestras e slides, não tendo abordagens que capturem a atenção do aluno, levando-o a uma falta de motivação e engajamento durante as aulas (NAVARRO *et al.*, 2004). Além disso, muitos alunos acabam se sentindo frustrados a estudar uma disciplina fora da sala de aula devido a grande quantidade de material teórico (LEGAULT *et al.*, 2006) (LATTANZI & HENRY, 1996).

Existe uma diferença entre as habilidades de Engenharia de Software (ES) ensinadas em uma universidade e as habilidades desejadas por uma organização. A indústria está, geralmente, insatisfeita com a falta de preparo dos recém-formados que ingressam no mercado de trabalho (BAKER *et al.*, 2005). Esse problema parece derivar da forma como a ES é tipicamente introduzida aos alunos, onde a teoria é apresentada em aulas e palestras e colocada em prática em um projeto fictício. Embora essas abordagens sejam essenciais para o aluno, elas carecem de um tratamento prático. Em particular, as aulas permitem apenas um aprendizado passivo, devido ao tamanho e escopo dos projetos de classe serem limitados ao ambiente acadêmico (NAVARRO *et al.*, 2004). O processo tradicional de ensino de ES consiste na apresentação de conceitos teóricos para o aprendizado do aluno. No entanto, embora essencial, pesquisas mostram que apenas esse conhecimento não é suficiente para que o aluno esteja preparado para ingressar no mercado de trabalho (DICHEVA *et al.*, 2015). É preciso que existam meios para que alunos consigam colocar em prática o que estão aprendendo, a “prática sem

teoria é cega”, a “teoria sem prática é estéril” (MCCAUGHERTY, 1991), uma completa a outra. Os educadores de ES não deveriam ensinar aplicações práticas apenas com teoria, é necessário ter uma aplicação real. O aluno precisa vivenciar de alguma maneira o que foi ensinado em teoria, para que possa usar seus conhecimentos de uma forma mais produtiva (HADJERROUIT, 2005) (GANNOD *et al.*, 2008).

Devido aos problemas listados anteriormente, os educadores estão buscando cada vez mais estratégias inovadoras de aprendizagem que combinem prazer com educação (RITTERFELD *et al.*, 2009), para que seja possível resolver, principalmente, os problemas de engajamento e motivação dos alunos em relação ao ensino. Tentar tornar o aprendizado prazeroso pode ser de grande valia para os alunos que precisam estudar um assunto. Com base nessa afirmação, uma técnica que pode ser uma possível solução para os problemas listados é a utilização de jogos como meio de ensino. Esses jogos são em sua grande maioria visuais, interativos e práticos, possuindo características que prendem a atenção do usuário, sendo ainda uma das principais maneiras de distração e prazer, tornando-os uma excelente oportunidade para fazer os alunos lembrarem o que foi ensinado de uma maneira diferente e inovadora (HUIZINGA, 1949). No entanto, apesar das vantagens oferecidas pelos jogos educativos, eles não são autossuficientes e devem ser complementados por outros dispositivos de aprendizagem (BOLDBAATAR & ŞENDURUR, 2018), ou seja, esses jogos podem ser utilizados como uma maneira inovadora de complementar a estratégia atual de ensino. Esses tipos de jogos já vêm sendo utilizado por algumas subáreas da ES, como para o ensino de tópicos como: gerenciamento de projetos, teste de software, modelagem de software, entre outras áreas. No entanto, a área de Reutilização de Software (RS) carece de jogos desse tipo.

A RS é uma das disciplinas da ES e é o processo de criar sistemas a partir de um ou mais já existentes, ao invés de criar do zero, ou seja, é o uso de artefatos e conhecimentos de software já existentes para construir algo novo (KRUEGER, 1992). NIU *et al.* (2011) definem reutilização como uma simples visão de “não reinventar a roda” e afirmam que ela já tem sido aplicada com sucesso em indústrias, como manufatura, automóveis e eletrônicos. Apesar de sua promessa, pode-se afirmar que a

RS de forma sistemática não se tornou ainda uma prática padrão, sendo este um problema comumente enfatizado pelas empresas (SINDRE *et al.*, 1992). Dentre as muitas causas desse fracasso, a falta de ensino é considerada uma das mais importantes, sendo esta disciplina ensinada de maneira isolada em poucos currículos acadêmicos (NIU, 2011).

Como afirmado anteriormente, o software ainda é amplamente construído do zero, sendo este um problema enfatizado pelas empresas (SINDRE *et al.*, 1992). Uma possível solução para esse problema seria a prática da reutilização baseada em componentes, que visa a construção de software com base em componentes pré-produzidos. Um componente de software pode ser compreendido como uma unidade de software com uma interface bem definida e dependências explicitamente especificadas. Ele pode possuir diferentes tamanhos e pode ser caracterizado de diferentes formas, desde um pequeno trecho de código, um pacote de software, um serviço da web, um módulo que encapsula um conjunto de funções ou até mesmo pode ser tão grande quanto um programa inteiro (YAO *et al.*, 2014) (SAMETINGER, 1997). Um componente é considerado reutilizável se sua funcionalidade for comum em vários programas (XIE & BROWNE, 2006) (FRAKES & POLE, 1994), ou seja, se ele for utilizado em mais de um local dentro do mesmo projeto ou não.

Além dos jogos existem outras estratégias para aumentar o engajamento, motivação e prática do aluno. Uma estratégia que vem sendo utilizado recentemente é a Programação Visual, que pode ser compreendida como a utilização de expressões visuais em um ambiente de programação como um atalho de edição para gerar código, que pode ou não ter uma sintaxe diferente daquela usada na codificação textual (BURNETT, 1999). Uma das principais plataformas de programação visual da atualidade é o Scratch que possibilita a criação de códigos por meio da combinações de blocos (CHANG, 2014). Fazendo uma comparação entre a área de componentes e a programação visual, é possível perceber uma semelhança, onde os dois conceitos visam criar programas a partir de um “bloco” ou componente pré-produzido.

Dessa forma, com base nos problemas listados anteriormente (i.e., reutilização sistemática não ser uma prática padrão, falta de ensino de RS, carência de jogos para o

ensino de RS), surgem as questões de pesquisa desta dissertação de mestrado: (1) O uso de jogos pode apoiar o ensino de RS?"; (2) Como aplicar jogos para apoiar o ensino de reutilização de software?

1.2 Objetivo

Na tentativa de solucionar os obstáculos de ensino de RS, o presente trabalho de pesquisa propõe utilizar jogos como técnica de ensino trazendo mais prática, interação e diversão para tentar melhorar o cenário atual de aprendizagem. Levando em consideração que, se os futuros engenheiros de software forem ensinados desde cedo a reutilizar, quando chegarem ao mercado de trabalho irão reproduzir o que aprenderam em seus cursos de graduação.

Com o objetivo geral de auxiliar o ensino de RS por meio dos jogos, desenvolveu-se um jogo, denominado Reuse Blocks, que avalia o nível de reutilização de um componente¹ por meio de métricas pré-definidas, visando a construção de componentes melhores. Esse jogo foi desenvolvido em Unity devido a facilidade de implementação do autor desta dissertação e fez uso da estratégia de programação visual, sendo inspirado no Scratch que é uma das principais linguagens visuais (MALONEY *et al.*, 2010), utilizando mecânicas bem parecidas com o jogo. Este objetivo geral pode ser decomposto em objetivos específicos:

- Caracterizar o estado da arte sobre o uso de jogos para o ensino de reutilização de software.
- Avaliar a utilização de jogos como metodologia de ensino para reutilização de software por meio da criação de jogos simples.
- Implementar um jogo que faz uso de uma IDE visual para o ensino de componentes de software.
- Realizar um estudo de avaliação para validação do jogo.

¹ Nível de reutilização de um componente: característica medida através do número de vezes que um componente pode ser reutilizado (BARNES *et al.*, 1988) (GAFFNEY & DUREK, 1989).

1.3 Metodologia

Esta pesquisa seguiu uma metodologia inspirada no modelo de *Design Science Research* (DSR) (LACERDA *et al.*, 2013) que foi adaptado para o contexto do trabalho, sendo necessário fazer algumas modificação no modelo DSR. O estudo exploratório e o desenvolvimento de viabilidade sustentaram a concepção da proposta do jogo para o ensino de RS. A Figura 1.1, apresenta a metodologia utilizada, sendo ela dividida em três grandes partes: **Viabilidade**, atividades iniciais que sustentaram o trabalho; **Agregação**, informações e artefatos que foram gerados a partir das atividades de viabilidade; **Proposta**, desenvolvimento da proposta da dissertação. Alguns artefatos e links foram omitidos da imagem para facilitar a compreensão da mesma.

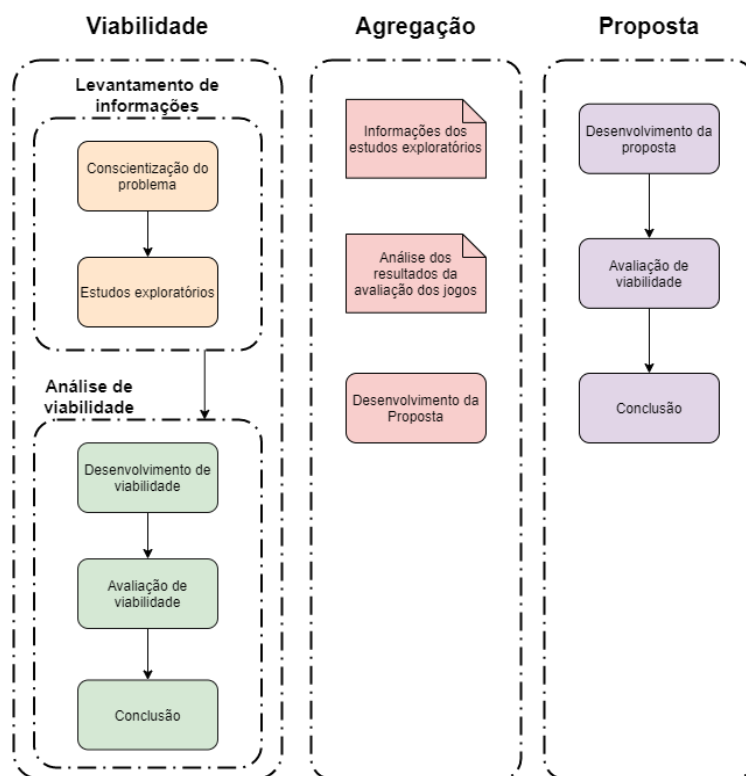


Figura 1.1: Metodologia de pesquisa, adaptado de LACERDA *et al.* (2013).

Cada uma das atividades apresentadas na Figura 1.1 é descrita com mais detalhes a seguir:

- **Viabilidade:** Atividades com objetivo de gerar embasamento teórico e prático para definição da proposta do trabalho.
 - **Levantamento de informações:** Atividades com objetivo de gerar embasamento teórico.
 - **Conscientização do problema:** Elaborar um estudo inicial para ter um primeiro contato com a área pesquisada.
 - **Elaborar estudo exploratório:** Elaborar um estudo prévio para descobrir o que já foi produzido sobre a utilização de jogos como estratégia de ensino de reutilização de software.
 - **Análise de viabilidade:** Atividades com objetivo de gerar embasamento prático.
 - **Desenvolvimento de viabilidade:** Desenvolver um (ou mais) jogo (s) simples baseado no estudo exploratório com o intuito de avaliar sua usabilidade e experiência proporcionada.
 - **Avaliação de viabilidade:** Executar um estudo de viabilidade para avaliar o jogo criado na fase anterior.
 - **Conclusão:** Analisar os resultados da avaliação de viabilidade com objetivo de entender sobre a usabilidade e experiência proporcionada pelo jogo.
- **Agregação**
 - **Informações dos estudos exploratórios:** Informações geradas pelos estudos exploratórios.
 - **Análise dos resultados da avaliação dos jogos:** Informações geradas pela análise da avaliação de viabilidade.
 - **Desenvolvimento da Proposta:** Definir a proposta da dissertação embasada nas informações coletadas nas fases anteriores.

- **Proposta**

- **Desenvolvimento da proposta:** Especificar um jogo para o ensino de RS, a fim de criar condições para que os alunos aprendam o conteúdo a ser transmitido de maneira prática, interativa e divertida.
- **Avaliação:** Executar um estudo para avaliar o jogo criado na fase anterior.
- **Conclusão:** Analisar as informações que foram coletadas na avaliação com objetivo de entender sobre a usabilidade e experiência proporcionada pelo jogo e ainda para avaliar futuras melhorias.

1.4 Organização do texto

Esta dissertação está organizada em seis capítulos. Neste capítulo, foi apresentado o contexto, bem como a motivação e o problema desta pesquisa. A questão de pesquisa foi definida junto com o objetivo e a metodologia para alcançá-lo.

O Capítulo 2 apresenta a fundamentação teórica sobre alguns assuntos que ajudam no entendimento dos demais capítulos.

O Capítulo 3 apresenta uma discussão sobre o que já foi produzido na literatura sobre o uso de jogos como técnica de ensino para RS, em geral.

O Capítulo 4 apresenta os jogos que foram construídos para se alcançar os objetivos desejados. É demonstrada a abordagem inicial para avaliação do uso de jogos para o ensino de reutilização e a proposta final que foi a criação do jogo Reuse Blocks para o ensino de reutilização, por meio da construção de componentes de software de maneira prática, interativa e divertida.

O Capítulo 5 apresenta o método de avaliação e resultados obtidos com o uso do jogo Reuse Blocks.

O Capítulo 6 apresenta uma discussão final sobre o conteúdo que foi apresentado nesse trabalho de pesquisa, suas contribuições, limitações e trabalhos futuros.

2 Fundamentação Teórica

Este capítulo tem como objetivo apresentar a fundamentação teórica necessária para um bom entendimento do trabalho de pesquisa realizado, que pode não ser de conhecimento do leitor. Neste sentido alguns conceitos básicos sobre tipos de ensino, jogos e programação visual são descritos a seguir.

2.1 Tipos de ensino

Existem basicamente três tipos de ensino distintos, cada um com suas particularidades. O primeiro tipo é conhecido como **passivo** e é a metodologia tradicional aplicada nos currículos escolares e acadêmicos, sendo esta fortemente centrada em palestras ministradas por professores através da demonstração de slides e quadro negro. Nesse tipo de ensino, o estudante tem uma postura passiva, tendo a função de receber e absorver o conteúdo a ser lecionado na disciplina (DIESEL *et al.*, 2017).

O segundo tipo de ensino é conhecido como **ativo** e visa, como princípio, dar mais autonomia e reflexão para o aluno, por meio de simulação de um problema que esteja inserido em um contexto real, fazendo uso de trabalhos em equipe, inovações e tendo o professor apenas como um mediador e facilitador do ensino (DIESEL *et al.*, 2017). Existem várias maneiras de utilizar o ensino ativo como, por exemplo, fazendo uso de jogos, simulação, aprendizagem baseada em projetos, instrução dos colegas, ensino sob medida, experimentação e aprendizagem baseada em problemas. Os principais pilares do ensino ativo são: ambientes flexíveis, onde todo o processo gira em torno do aluno, através de um conteúdo ministrado pelo professor; e *feedback* imediato com trabalhos práticos realizados dentro e fora de sala de aula (JAIME *et al.*, 2015) (S. SCHMITZ, 2016).

Por fim, o último tipo de ensino é o **híbrido**, que é uma combinação do ensino ativo e passivo, tirando proveito das principais características desses dois tipos de ensino (CHRISTENSEN *et al.*, 2013).

2.2 Jogos

Na literatura, existem diversas definições que são dadas para jogos. Entretanto, é importante notar que não há uma definição precisa e consensual sobre o que é um jogo (XEXÉO *et al.*, 2017). Um dos pressupostos é que jogos são atividades divertidas, que possuem um resultado imprevisível, dentro de um mundo fictício que não gera algo produtivo (CAILLOIS, 2015). Outra definição é que os jogos são sistemas nos quais os usuários participam de um conflito artificial, definido por regras, que termina em um resultado quantificável (SALEN *et al.*, 2004). Portanto, em geral, jogos podem ser definidos como atividades que usam um mundo abstrato em que decisões, ações e regras são realizadas, visando uma atividade recreativa, na forma de distração ou divertimento.

Os jogos possuem uma série de características, dentre elas se destacam (ADAMS & DORMANS, 2012):

- São atividades, sociais ou culturais: as pessoas interagem enquanto jogam;
- São voluntários: é uma atividade livre, ninguém joga por obrigação, mas sim por vontade própria;
- São significativos e absorventes: os jogos apresentam um significado específico e prendem a atenção do usuário, que por sua vez não percebem o passar do tempo;
- Utilizam um mundo abstrato, um ambiente fora do mundo real;
- São não produtivos, garantindo que o jogo não tenha outras motivações mais importantes do que jogar;
- Possuem um final incerto: grande parte da diversão de um jogo vem do fato de que ele é um desafio auto imposto, o que é garantido pela liberdade de escolha do jogador;
- Podem funcionar para um ou mais jogadores (*single player* ou *multiplayer*);

- Os jogadores podem tomar decisões que modificam, interativamente e de forma quantificável, o estado do sistema;
- Os jogadores muitas vezes buscam por objetivos conflitantes, para criar um ambiente de competição;
- São regulados, orientados e limitados por regras que são livremente aceitas pelos jogadores e;
- Possuem recompensas, para motivar ainda mais o jogador.

Os jogos possuem funções além do entretenimento. Eles podem servir para a transmissão de conteúdo educacional, treinamento para o usuário, saúde, etc. Essa ramificação de jogos possui uma denominação diferente, são chamados de jogos sérios, ou jogos com propósitos. O termo jogo sério foi concebido devido ao fato dos jogos poderem ser usados para outros fins que não apenas o entretenimento (ADAMS & DORMANS, 2012). Estes possuem um propósito, que visa alcançar algum efeito no mundo real, oferecendo uma oportunidade para tentar novas abordagens para solução de problemas de uma forma segura, barata e sem consequências.

Os jogos sérios possuem sub-ramificações, de acordo com o seu propósito, jogabilidade, objetivo e audiência, entre elas, se destacam: *advergames*, que são jogos que visam transmitir uma mensagem de *marketing* ou comunicação; *newsgame*, jogos para transmitir uma mensagem informativa; *exergame*, jogos para treinamentos físicos; *edumarketgame*, jogos que transmitem uma mensagem de *marketing* ou educativa; *health games*, jogos para ajudar em algum tratamento de saúde; *business games*, jogos que visam transmitir alguma mensagem relacionada a alguma propaganda de um produto; *political games*, transmitem alguma mensagem política e; *games for education* ou *game-based learning*, jogos para transmitir uma mensagem educacional, que foi o foco desse trabalho (DJAOUTI *et al.*, 2011).

2.2.1 Game-based learning

Como dito anteriormente, *Game-Based Learning* (GBL) é uma das sub-ramificações dos jogos sérios e pode ser compreendida como a utilização de jogos com propósito educativo, com o objetivo de otimizar a experiência de aprendizagem através

da utilização de características dos jogos, tais como: simulações, aprendizagem baseada em erros e resoluções de problemas (PIVEC *et al.*, 2003). Existe ainda o termo *Digital Game-Based Learning* (DGBL), que representa o uso de jogos com propósitos educativos em suas versões digitais (BAU, 2015c). Vale destacar que esses jogos se diferem um pouco das características que foram demonstradas na seção anterior, sendo eles produtivos e não voluntários e isso implica em um grande desafio que é criar jogos educativos mas que tenham um relacionamento com o entretenimento.

Devido as suas características, muitos estudiosos estão utilizando jogos para criar aulas mais interativas, divertidas e absorventes, onde o aluno consegue se sentir engajado durante o aprendizado.

Para se obter um melhor entendimento de como o aprendizado baseado em jogos funciona, um breve modelo foi derivado a partir da leitura de dois artigos: o primeiro que demonstra o modelo de aprendizagem do GBL (PIVEC *et al.*, 2003) e o segundo que demonstra os mecanismos e os princípios por parte do uso do GBL (PERROTTA *et al.*, 2013). A seguir são apresentadas as três etapas desse processo, que podem ser observadas na Figura 2.1.

1. **Entrada de dados:** é composta basicamente pelo corpo de conhecimento da matéria a ser lecionada e os mecanismos/caraterísticas que serão utilizados no jogo, como regras, níveis, *feedback*, etc;
2. **Ciclo do jogo:** coloca em prática os princípios do GBL (*i.e.*, aumentar a motivação, diversão, autonomia e autenticidade, além de experimentação para o aluno); e
3. **Saída de dados:** a aprendizagem e a recordação do conteúdo que foi ministrado em aula.

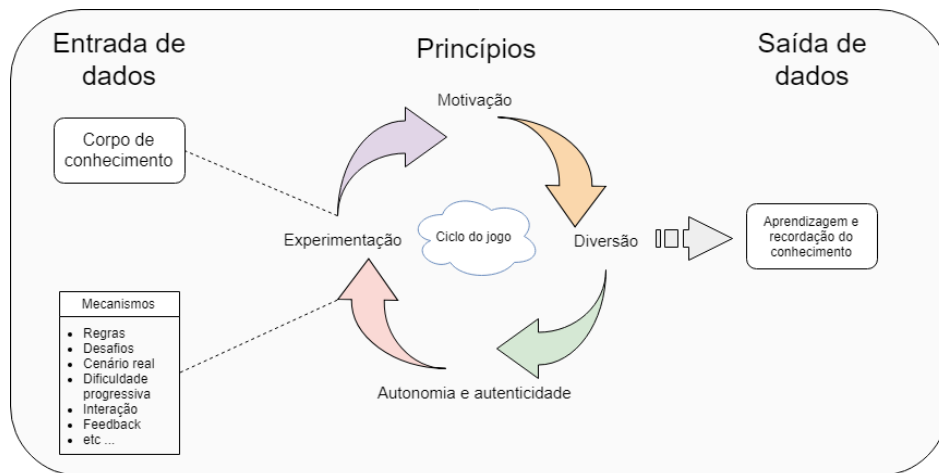


Figura 2.1: Modelo do aprendizado baseado em jogos, adaptado de PIVEC *et al.* (2003) e PERROTTA *et al.* (2013).

2.2.2 Modelagem de jogos

Existem diversas maneiras de organizar as ideias para a construção de um jogo, como modelos de fluxos, *game play*, características, entre outros. Dois modelos bastante conhecidos na área de jogos e que foram utilizados nesse trabalho são o MDA (HUNICKE *et al.*, 2004) e OCR (GUARDIOLA, 2016).

O MDA (*Mechanics, Dynamics Aesthetics*) é uma abordagem para a compreensão de jogos que tenta descrever suas características por meio de três componentes (HUNICKE *et al.*, 2004):

1. **Mechanics:** são os componentes essenciais de um jogo, como as ações e as regras;
2. **Dynamics:** Descreve o comportamento da mecânica, agindo sobre as entradas e as saídas de dados para que, assim, consiga fazer com que o jogador expresse suas sensações a partir do jogo - através da execução de uma mecânica, acontece uma dinâmica que proporciona uma sensação; e
3. **Aesthetics:** Descreve a resposta emocional desejada, invocada no jogador, quando ele interage com o jogo. Existem vários sentimentos que podem

acontecer ao jogar um jogo, entre eles se destacam: Sensação, Fantasia, Narrativa, Desafio, Sociedade, Descoberta, Expressão e Submissão.

O segundo modelo é conhecido como *game loop* ou modelo OCR (*Objectives, Challenges, Rewards*). Ele pode ser entendido como a estrutura básica de um jogo, ou seja, o que tem que acontecer em uma partida de jogo, e pode ser dividido em três etapas (GUARDIOLA, 2016):

1. **Objectives:** o que se deseja alcançar com o jogo a ser criado;
2. **Challenges:** o que o jogador deve realizar para conseguir chegar aos objetivos; e
3. **Rewards:** o que o jogador irá ganhar quando resolver um determinado desafio.

2.2.3 Mecânicas para aprendizagem

Como dito na seção anterior, a mecânica é responsável por gerenciar os componentes essenciais do jogo, como as ações e as regras. Em jogos educativos, é importante ter em mente que as regras e ações que serão impostas para o usuário devem ser pensadas com bastante cuidado, para que não afetem a experiência de aprendizado.

Dependendo do intuito do jogo a ser criado, existem mecânicas que se adequam melhor ao contexto (BOLLER & KAPP, 2017). Se o objetivo é recordar informações que foram aprendidas em sala de aula, jogos do tipo *quiz* e *matching* são mais adequados. Se a finalidade é forçar o aluno a pensar e analisar uma situação, jogos do tipo *puzzle*, RPG e estratégia são mais apropriados. Já para a compreensão de um conteúdo sem um conhecimento prévio, é aconselhado que se utilize jogos que contenham uma história para o jogador como por exemplo um *storytelling*. Para jogos com uma abordagem um pouco mais prática, é aconselhado que se utilize simuladores ou jogos de construção para incentivar a criatividade do aluno (BOLLER & KAPP, 2017). Vale lembrar que esses tipos de jogos não são excludentes em suas finalidades, eles possuem mais de uma, ou seja, um jogo do tipo *storytelling* também pode ser utilizado para incentivar o aluno recordar algo. No entanto, ele é mais recomendado para apresentar conteúdo sem um conhecimento prévio.

2.3 Programação Visual

Programação visual é a utilização de expressões visuais em um ambiente de programação como um atalho de edição para gerar código que pode ou não ter uma sintaxe diferente daquela usada na codificação textual (BURNETT, 1999). Existem diversos exemplos desses ambientes, um dos mais conhecidos é o Scratch, que permite que os usuários iniciantes, em sua grande maioria entre 8 e 16 anos, aprendam programação enquanto trabalham em projetos animados e jogos (MALONEY *et al.*, 2010). Como este, existem diversas outras linguagens como: Alice, Raptor, Etoys, Bubble, Droplet, Google Blockly, etc.

Existem diversas vantagens em fazer uso da programação visual e as principais são (NIU, 2011) (STEINMACHER *et al.*, 2013):

- Geração de códigos por meio da combinação de blocos;
- Tornar a programação mais acessível a um público específico;
- Melhorar a qualidade com que as pessoas executam tarefas de programação;
- Diminuir o número de erros gerados pelos programadores iniciantes, não tendo que se preocupar com a sintaxe da linguagem;
- Melhorar a velocidade com que pessoas executam tarefas de programação;
- *Feedback* imediato e execução visual; e
- Minimizar o conjunto de comandos.

A programação visual vai além de ser utilizada apenas por plataformas como o Scratch, sendo utilizada por novas abordagens como o *No-code* (GSCHLIESSER, 2019) e o *Low-Code* (GOLOVIN, 2017). Essas abordagens são bem parecidas e buscam agilidade e simplicidade no desenvolvimento de software, elas podem ser compreendidas como um editor gráfico com ferramentas de arrastar e soltar que possibilitam o usuário gerar programas de computador. Esse editor fornece modelos para fluxo de trabalho, bibliotecas de elementos e personalização de interface para criar aplicativos com pouca ou nenhuma codificação (GSCHLIESSER, 2019) (ROUSE, 2018). Essa estratégia é apreciada, principalmente, por grandes empresas que precisam

projetar rapidamente aplicativos em larga escala (GSCHLIESSER, 2019). No entanto, existem algumas diferenças entre essas abordagens. O *Low-code* pode ser utilizado para gerar até grandes aplicações e deve ser utilizado por pessoas com um perfil mais técnico, que entendam um pouco de programação, enquanto o *No-code* foi elaborado para ser utilizado por pessoas sem experiência com programação e, atualmente, as plataformas criadas por essa abordagem são bem simplificadas (GSCHLIESSER, 2019) (ROUSE, 2018) (GOLOVIN, 2017).

Nessa dissertação, o jogo criado visou ensinar RS por meio de componentes de software e ser utilizado por usuários em formação, i.e., com pouca experiência de programação, adotando técnicas de programação visual. O *No-code* se relaciona com esse trabalho através da abstração criada pela programação visual, onde o usuário não precisa escrever código. No entanto, esse código também pode ser visualizado, sendo gerado pela aplicação, trazendo traços característicos do *Low-code*.

2.4 Considerações Finais

Este capítulo apresentou a fundamentação teórica julgada relevante para um bom entendimento do trabalho de pesquisa realizado. Inicialmente apresentou-se os tipos de ensino para que o leitor consiga entender sobre a maneira atual de ensino e identificar o ensino ativo como uma alternativa, principalmente através da utilização de jogos.

Em seguida, outros dois conceitos foram apresentados: o de Jogos, mais especificamente *Game-based learning*, e a programação visual, que são apresentados como estratégias a serem utilizadas para ensinar a partir de uma perspectiva visual.

3 Estudos exploratórios

O processo de pesquisa apresentado nesse capítulo foi conduzido durante o período de outubro de 2018 a maio de 2019 e visou identificar os trabalhos que já haviam sido realizados na área de jogos como estratégia para o ensino de RS.

Nesse estudo, foram utilizadas duas abordagens de pesquisa principais, chamadas de mapeamento sistemático e mapeamento terciário. Mapeamento sistemático é um método de estudo secundário que, sistematicamente (ou seja, baseado em um processo ou protocolo estruturado e repetitivo), explora e categoriza os estudos em um determinado campo de pesquisa e fornece uma estrutura no formato de relatórios de pesquisa e resultados, permitindo fornecer uma visão geral sobre um determinado assunto (KITCHENHAM, 2004). Já um mapeamento terciário é um estudo que faz agregação de informações com base em estudos secundários que já foram previamente realizados (KEELE, 2007). O mapeamento desenvolvido segue o protocolo proposto por Kitchenham (KITCHENHAM, 2004).

Inicialmente, o protocolo de pesquisa realizado buscou responder cinco perguntas principais que estavam diretamente ligadas ao ensino da disciplina de reutilização de software. No entanto, foi observado que a *string* de busca inicial não retornou muitos trabalhos que estavam relacionados ao uso de jogos para o ensino dessa disciplina. Esse primeiro estudo retornou trabalhos relacionados a três tópicos principais: ensino de programação através de jogos, ensino de reutilização sem o uso de jogos e ensino de engenharia de software através de jogos. Levando em consideração que a primeira pesquisa não conseguiu responder as perguntas do estudo proposto, esse estudo foi dividido em outros três estudos menores, um para cada tópico que foi encontrado. A Figura 3.1 mostra o protocolo que foi utilizado nesse estudo com mais detalhes, sendo executado inicialmente pelo estudo “Uso de jogos para o ensino de reutilização de software” e seguindo-se da esquerda para a direita.

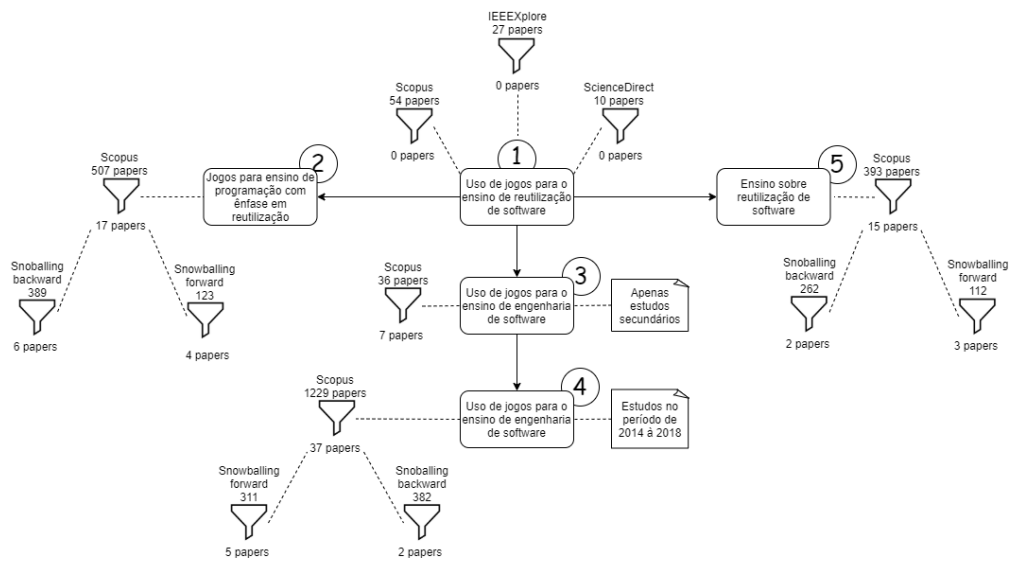


Figura 3.1: Protocolo de pesquisa. Fonte: do autor.

3.1 Organização dos estudos

O estudo “Uso de jogos para o ensino de reutilização de software” é o estudo principal dessa seção de estudos exploratórios, tendo as principais perguntas que esse estudo se propõe a responder. Foi pela falta de estudos relevantes retornados neste estudo que suas ramificações foram criadas. As perguntas que esse trabalho procurou responder são listadas a seguir.

Q1: Qual(s) a(s) principal(s) dificuldade(s) do ensino de reutilização de software?

Q2: Qual(s) a(s) definição(s) de jogos foi(foram) utilizada(s)?

Q3: Qual(s) a(s) principal(s) vantagem(s) / motivação(s) do uso de jogos para o ensino de reutilização de software?

Q4: Qual(s) a(s) desvantagem(s) do uso de jogos para o ensino de reutilização de software?

Q5: Qual(s) a(s) principal(s) característica(s) do jogo(s) utilizado(s)?

Q6: Qual(is) foi(ram) o(s) método(s) de avaliação utilizado(s)?

O segundo estudo realizado foi denominado de “Jogos para o ensino de programação com ênfase em reutilização”. Essa ramificação do estudo principal originou-se a partir da observação de que muitos trabalhos encontrados no estudo anterior tratavam do ensino de programação e que, de maneira geral, a reutilização de software está fortemente enraizada (para as pessoas que não são da área de reutilização) na programação, não sendo adaptável a esse contexto apenas. Com isso, este estudo pretendeu identificar os jogos que estão sendo utilizados para o ensino de programação com ênfase/potencial para reutilização, ou seja, encontrar jogos que foram desenvolvidos para o ensino de programação, mas que poderiam ser utilizados para ensinar alguns dos fundamentos de reutilização de software, como, por exemplo: desenvolvimento de funções, componentes, orientação a objetos, etc. Como este estudo é uma ramificação do estudo principal, ele tem como principal função responder algumas das perguntas que estão contidas no estudo anterior (adaptadas), não se limitando somente a este propósito, sendo elas:

Q1: Qual(s) a(s) definição(s) de jogos foi utilizada?

Q2: Qual(is) a(s) principal(is) vantagem(s) / motivação(s) do uso de jogos para o ensino de programação?

Q3: Qual(is) a(s) desvantagem(s) do uso de jogos para o ensino de programação?

Q4: Qual(is) a(s) principal(s) característica(s) do jogo(s) utilizado(s)?

Q5: Qual(is) foi(ram) o(s) método(s) de avaliação utilizado(s)?

O terceiro estudo realizado foi denominado de “Uso de jogos para o ensino de engenharia de software”. Essa ramificação do estudo principal originou-se a partir da observação de que muitos trabalhos encontrados no estudo principal tratavam do ensino de ES e como a reutilização é uma das disciplinas da ES, optou-se pela criação dessa ramificação, levando em consideração que a RS faz parte da ES, características de ensino utilizadas em ES devem funcionar para RS também.

Esse estudo ao ser executado retornou um resultado de 4.428 artigos. Devido a esse alto número de artigos encontrados e levando em consideração a ideia de reutilização, optou-se pela realização de um estudo terciário, visando diminuir o esforço de trabalho da revisão, ou seja, buscou-se encontrar estudos secundários que já estivessem prontos.

Para mais detalhes sobre esse trabalho como estudo isolado, consultar CASTRO *et al.* (2019).

No entanto, os artigos secundários que foram observados procuravam responder perguntas ligeiramente diferentes das que este estudo procura responder. Buscando encontrar mais informações sobre o uso de jogos para o ensino de ES, o estudo foi executado novamente sem o filtro para trabalhos secundários. No entanto, optou-se pela utilização de um filtro de data, retornando apenas os artigos que foram publicados dentro do período de 5 anos, como realizado por JIANG *et al.* (2015).

As perguntas que os mapeamentos procuraram responder podem ser observadas a seguir. No entanto, a pergunta **Q6** só foi respondida pelo mapeamento terciário para entender quais eram as perguntas que o artigo procurava responder, visando relacionar com as perguntas que serão demonstradas a seguir. Como este estudo é uma ramificação do estudo principal, ele tem como principal objetivo responder algumas das perguntas que estão contidas no estudo anterior (adaptadas), não se limitando somente a este propósito. São estas:

- Q1:** Qual(s) a(s) definição(s) de jogos foi utilizada?
- Q2:** Qual(s) a(s) principal(s) vantagem(s) / motivação(s) do uso de jogos para o ensino de engenharia de software?
- Q3:** Qual(s) a(s) desvantagem(s) do uso de jogos para o ensino de engenharia de software?
- Q4:** Qual(s) a(s) principal(s) característica(s) do jogo(s) utilizado(s)?
- Q5:** Qual(is) foi(ram) o(s) método(s) de avaliação utilizado(s)?
- Q6:** Qual(s) pergunta(s) o artigo procura responder?

Por fim, o quarto e último mapeamento que foi realizado foi denominado de “Ensino de reutilização de software”. Essa ramificação da pesquisa foi executada uma vez que muitos trabalhos do estudo principal trouxeram artigos sobre o ensino de reutilização de software, mas que não utilizavam jogos para isso. A partir desse estudo, pretendeu-se entender quais são as dificuldades, características e áreas que relevantes para o ensino de reutilização de software, para que assim fosse possível criar um jogo

que consiga seja capaz de auxiliar o ensino dessa disciplina. Nesse estudo em particular, quatro novas perguntas foram adicionadas, buscando entender um pouco melhor sobre a área de reutilização. Como este estudo é uma ramificação do estudo principal, ele tem como principal função responder algumas das perguntas que estão contidas no estudo anterior (adaptadas), não se limitando somente a este propósito. As perguntas que esse estudo procura responder são:

Q1: O que é reutilização de software?

Q2: Qual a importância do ensino de reutilização de software?

Q3: Qual(is) a(s) dificuldade(s) do ensino de reutilização de software?

Q4: Qual(is) foi(ram) a(s) característica(s) utilizada(s) no meio de ensino?

Q5: Qual(is) área(s) da reutilização de software foi(ram) ensinada(s)?

3.2 Protocolo de pesquisa

A escolha da base de pesquisa ficou a critério da execução da primeira *string* de busca que foi executada nas bases Scopus², ScienceDirect³ e IEEEExplore⁴, conforme recomendado por LELIS *et al.* (2015) e KITCHENHAM (2004). No entanto, essa execução demonstrou alguns obstáculos relacionados a duas bases de pesquisa, sendo elas a ScienceDirect e a IEEEExplore. Essas bases possuem limitadores de quantidade de operadores (AND e OR) e termos (*keywords*), o que dificulta a criação da *string* de busca. Com isso, a base de pesquisa utilizada em todas as ramificações seguintes foi a Scopus, conforme recomendado por MOTTA *et al.* (2016) e MATALONGA *et al.* (2017).

De acordo com MOTTA *et al.* (2016) e MATALONGA *et al.* (2017), os procedimentos de *snowballing* pode mitigar a falta de outros motores de busca e complementa a estratégia, fazendo a pesquisa através das referências e citações dos artigos. Portanto, para minimizar a perda de alguns artigos (já que a pesquisa não foi

² <https://www.scopus.com>

³ <https://www.sciencedirect.com>

⁴ <https://ieeexplore.ieee.org>

reproduzida em todas as bases e não existe uma *string* de busca perfeita), optou-se pela utilização do procedimento de *snowballing forward*⁵ e *backward*⁶ (um nível), que verifica as referências e as citações dos artigos buscando relevâncias (WOHLIN, 2014).

3.2.1 Procedimento

1. Executar a string de busca;
2. Aplicar os critérios de inclusão baseados no título do artigo;
3. Aplicar os critérios de inclusão baseados no abstract do artigo;
4. Aplicar os critérios de inclusão com base no texto completo do artigo.

Depois do término da seleção (executado por todos os estudos com exceção do mapeamento terciário e estudo principal, já que este não retornou resultados):

5. Execute o procedimento de *snowballing backward* (um nível) e *forward* (WOHLIN, 2014).
 - a. Aplicar os critérios de inclusão baseados no título do artigo;
 - b. Aplicar os critérios de inclusão baseados no abstract do artigo;
 - c. Aplicar os critérios de inclusão com base no texto completo do artigo.

Foi utilizada para o gerenciamento e suporte dos artigos a ferramenta JabRef⁷.

3.2.2 Critério de inclusão

1. O artigo deve estar no contexto do mapeamento a ser executado⁸;
2. O artigo deve relatar um estudo primário ou secundário;
3. O artigo deve relatar um estudo baseado em evidências baseado em métodos empíricos (por exemplo, entrevistas, pesquisas, estudos de caso, experimentos formais, etc.);

⁵ Refere-se à identificação de novos artigos com base nos trabalhos que referenciaram o artigo que foi analisado (WOHLIN, 2014).

⁶ Refere-se à identificação de novos artigos com base nos trabalhos que foram referenciados no artigo que foi analisado (WOHLIN, 2014).

⁷ <http://www.jabref.org>

⁸ Contextos:

1. “Uso de jogos para o ensino de reutilização de software” e “Ensino de reutilização de software”: Reutilização de Software;
2. “Jogos para ensino de programação com ênfase em reutilização”: Programação;
3. “Uso de jogos para o ensino de engenharia de software”: Engenharia de Software

4. O artigo deve fornecer dados para responder a pelo menos uma das questões de pesquisa;
5. O artigo deve ser escrito no idioma inglês.

3.2.3 Critério de exclusão

1. Capítulos de livro, chamada para congressos;
2. Estudos que não puderem ser acessados completamente;
3. Estudos que não estejam dentro da área de Ciência da Computação ou Engenharia.

A Figura 3.2 mostra uma sumarização do modelo do protocolo de pesquisa realizado.

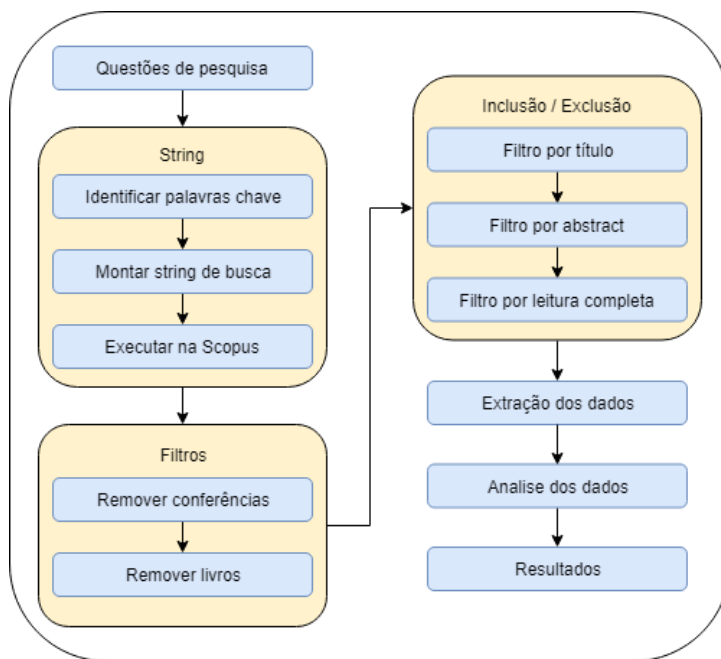


Figura 3.2: Sumarização do protocolo de pesquisa baseado no modelo criado por CALDERÓN *et al.* (2018)

3.2.4 Procedimento de extração

Todos os procedimentos de extrações foram realizados por um pesquisador, utilizando um formulário baseado no que é apresentado a seguir. Cada artigo que passou no processo de seleção foi lido em profundidade e os seguintes dados foram extraídos:

- *Abstract*
- Uma breve descrição do trabalho
- O tipo do estudo (*survey*, revisão sistemática, mapeamento sistemático, etc)
- Questões de pesquisas que foram descritas na Seção 3.1
- Informações extra, caso fosse julgado necessário

Exemplo do formulário de extração que foi utilizado em cada um dos mapeamentos:

Tabela 3.1: Modelo do formulário de extração.

Nome do artigo	
Abstract	
Descrição	
Tipo de estudo	
Q1: Questão de pesquisa 1?	
Q2: Questão de pesquisa 2?	
Q3: Questão de pesquisa 3?	
...	
QN: Questão de pesquisa N?	
Informações extras	

3.3 Strings de busca

A definição das palavras-chave de busca utilizadas na pesquisa foi feita com base na estratégia PICO (PETTICREW & ROBERTS, 2009), utilizando-se três ou dois dos quatro níveis. Essa estratégia visa dividir a *string* de busca em quatro partes: população, intervenção, comparação e resultado. Algumas palavras foram colocadas

com o caractere ‘*’ no final, para que fosse possível trazer qualquer que seja a derivação da palavra original.

Mediante os sinônimos identificados em cada uma das tabelas mostradas a seguir, foi possível derivar as *strings* de busca para cada estudo. As *strings* foram definidas agrupando-se palavras-chave do mesmo domínio com o operador lógico “OR” e agrupando-se os domínios com o operador lógico “AND”.

Todos os mapeamentos utilizaram basicamente as mesmas palavras-chave modificando apenas a população, com exceção do estudo sobre o ensino de reutilização, que fez uso apenas da intervenção. A Tabela 3.2 mostra as palavras-chave divididas de acordo com a estrutura PICO (PETTICREW & ROBERTS, 2009).

A especificação da *string* de busca inicial foi executada primeiramente com diversos sinônimos e subáreas de reutilização de software, entre elas: *domain engineering*, *product line*, *feature modelling*, *domain design*, *domain implementation*, *product family*, *software family* e *component*. No entanto, foi verificado que muitos ruídos foram retornados. Por este motivo, optou-se pela generalização da população de pesquisa. Em uma revisão da literatura, a população da *string* de busca é utilizada para reduzir o número de estudos retornados. No entanto, é importante evitar qualquer restrição à população, utilizando-se o termo mais geral possível se for generalizar (KITCHENHAM, 2004). Por isso, optou-se pela utilização da palavra chave *software reuse*, levando em conta que reutilização de software é um termo geral, sendo um guarda-chuva de informações tendo vários sub-tópicos, ou seja, um trabalho que esteja no contexto de um desses subtipos e seja realmente sobre reutilização, possivelmente utilizará este termo como palavra chave também, reduzindo assim o número de ruídos da *string* de busca.

Cada uma das *strings* de busca foi validada por meio de artigos de controle com base no conhecimento prévio do autor de alguns trabalhos na área por meio de pesquisas *ad-hoc*. Utilizando esse meio de controle, a validação é feita por meio da execução da *string* de busca e a mesma deve retornar o artigo selecionado, para assim saber se a *string* a ser usada está de acordo com o que se busca encontrar.

Tabela 3.2: Tabela de sinônimos e *string* de busca.

PICO	Sinônimos
Population (População)	De acordo com o estudo
Intervention (Intervenção)	<i>Tutoring, teach*, instruction, discipline, schooling, education*, mentoring, course, learn*, train*, syllabus</i>
Comparison (Comparação)	Não se aplica
Outcome (Resultado)	<i>Game*, gami*, play*, "serious games", edutainment, "game based learning", simulation</i>

3.3.1 Uso de jogos para o ensino de reutilização de software

Tabela 3.3: *String* concatenada do uso de jogos para o ensino de reutilização de software.

Artigo de controle	Como não era de conhecimento do autor algum artigo que ensinasse reutilização através de jogos, não utilizou-se esse critério de validação.
População	<i>software reuse</i>
	<i>TITLE-ABS-KEY (("software reuse") AND (tutoring OR teach* OR instruction OR discipline OR schooling OR education* OR mentoring OR course OR learn* OR train* OR syllabus) AND (game* OR gami* OR play* OR "serious games" OR edutainment OR "game based learning" OR simulation))</i>

3.3.2 Jogos para o ensino de programação com ênfase em reutilização

A *string* de busca dessa seção sem nenhum filtro retornava um número de 1.449 artigos. Sendo este um número relativamente alto, optou-se por utilizar um filtro de data, buscando apenas por artigos que foram publicados dentro do período de 5 anos, como realizado por JIANG *et al.* (2015).

Tabela 3.4: *String* concatenada do uso de jogos para o ensino de programação com ênfase em reutilização.

Artigo de controle	<i>Teaching abstraction, function and reuse in the first class of CS1 - A lightbot experience</i>
População	<i>Programming language, algorithm experience, algorithm skills, algorithm alternative, algorithm method, coding experience, coding skills, coding method, coding alternative</i>
Filtro	<i>(LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO (PUBYEAR , 2017) OR LIMIT-TO (PUBYEAR , 2016) OR LIMIT-TO (PUBYEAR , 2015) OR LIMIT-TO (PUBYEAR , 2014))</i>
	<i>TITLE-ABS-KEY (("programming language" OR "algorithm experience" OR "algorithm skills" OR "algorithm alternative" OR "algorithm method" OR "coding experience" OR "coding skills" OR "coding method" OR "coding alternative") AND (tutoring OR teach* OR instruction OR discipline OR schooling OR educat* OR mentoring OR course OR learn* OR train* OR syllabus) AND (game* OR play* OR "serious games" OR gami* OR edutainment) AND (LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO (PUBYEAR , 2017) OR LIMIT-TO (PUBYEAR , 2016) OR LIMIT-TO (PUBYEAR , 2015) OR LIMIT-TO (PUBYEAR , 2014)))</i>

3.3.3 Uso de jogos para o ensino de engenharia de software (mapeamento terciário)

Ao executar a *string* de busca sobre jogos para o ensino de engenharia de software, observou-se que a mesma retornou um resultado de 4.428 artigos, o que seria um número relativamente alto para efetuar o mapeamento. Levando em consideração o alto número de trabalhos relacionados e ideia de reutilização (aproveitar revisões que já estivessem sido realizadas), buscou-se encontrar estudos secundários que já estivessem prontos. Com isso, um novo “AND” foi colocado no final da *string*, concatenando o segmento que pode ser observado no filtro da Tabela 3.5.

Tabela 3.5: *String* concatenada do uso de jogos para o ensino de engenharia de software (mapeamento terciário).

Artigo de controle	<i>Software engineering education and games: A systematic literature review</i>
População	<i>Software engineering</i>
Filtro	<i>("systematic literature review" OR "systematic review*" OR "mapping study" OR "systematic mapping" OR "structured review" OR "secondary study")</i>
	<i>TITLE-ABS-KEY (("Software engineering") AND (tutoring OR teach* OR instruction OR discipline OR schooling OR educat* OR mentoring OR course OR learn* OR train* OR syllabus) AND (game* OR gami* OR play* OR "serious games" OR edutainment OR "game based learning" OR simulation) AND ("systematic literature review" OR "systematic review*" OR "mapping study" OR "systematic mapping" OR "structured review" OR "secondary study"))</i>

3.3.4 Uso de jogos para o ensino de engenharia de software

Com a execução da *string* da Seção 3.3.2, obteve-se um total de 36 artigos, que foram filtrados, sobrando apenas 7 trabalhos que foram julgados como relevantes para essa dissertação. Esses trabalhos foram lidos e analisados, buscando responder as questões de pesquisa da Seção 3.1, entretanto, observou-se que, por se tratar de trabalhos secundários, os trabalhos buscavam responder suas próprias questões de pesquisa, fugindo um pouco do escopo dessa dissertação.

Buscando encontrar mais informações sobre as perguntas dessa dissertação, executou-se novamente a *string* da Seção 3.3.2 sem o filtro de mapeamentos terciários. No entanto, como já foi dito a *string* sem um filtro estava retornando muitos trabalhos, por isso, optou-se por utilizar um filtro de data, buscando apenas por artigos que foram publicados dentro do período de 5 anos, como realizado por JIANG *et al.* (2015).

Tabela 3.6: *String* concatenada do uso de jogos para o ensino de engenharia de software.

Artigo de controle	(1) <i>Software engineering education and games: A systematic literature review;</i> (2) <i>SimuLES-W: A collaborative game to improve software engineering teaching</i>
População	<i>Software engineering</i>
Filtro	(<i>LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO (PUBYEAR , 2017) OR LIMIT-TO (PUBYEAR , 2016) OR LIMIT-TO (PUBYEAR , 2015) OR LIMIT-TO (PUBYEAR , 2014)</i>)
	<i>TITLE-ABS-KEY (("Software engineering") AND (tutoring OR teach* OR instruction OR discipline OR schooling OR educat* OR mentoring OR course OR learn* OR train* OR syllabus) AND (game* OR gami* OR play* OR "serious games" OR edutainment OR "game based learning" OR simulation) AND (LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO (PUBYEAR , 2017) OR LIMIT-TO (PUBYEAR , 2016) OR LIMIT-TO (PUBYEAR , 2015) OR LIMIT-TO (PUBYEAR , 2014)))</i>

3.3.5 Ensino de reutilização de software

Tabela 3.7: *String* concatenada do ensino de reutilização de software.

Artigo de controle	<i>Reuse a "software reuse" course</i>
População	<i>software reuse</i>
	<i>TITLE-ABS-KEY (("software reuse") AND (tutoring OR teach* OR instruction OR discipline OR schooling OR education* OR mentoring OR course OR learn* OR train* OR syllabus))</i>

3.4 Análise de extração dos artigos

Com exceção do mapeamento terciário e do estudo principal “Uso de jogos para o ensino de reutilização de software”, todas tabelas de conjunto final de artigos analisados possuem uma coluna com o título de “origem”. Esta foi preenchida com três

termos distintos “EP” com significado de estudo principal (referindo-se aos artigos provenientes da execução da string de busca), “SB” com significado de *snowballing backward* e “SF” com significado de *snowballing forward* (referindo-se aos artigos que foram incluídos por meio da execução do procedimento de *snowballing*).

3.4.1 Uso de jogos para o ensino de reutilização de software

3.4.1.1 Execução

Tabela 3.8: Execução do uso de jogos para o ensino de reutilização de software, base Scopus.

Atividade	Data de execução	Resultado	Número de papers
Primeira execução	11/10/2018 - 20h:23min	54 adicionados	54
Remover conferência/workshops	12/10/2018	11 removidos	43
Remover livros	12/10/2018	0 removidos	43
Remover por título	12/10/2018	26 removidos	17
Remover por abstract	12/10/2018	15 removidos	2
Artigos não encontrados	12/10/2018	0 removidos	2
Artigos para ler	13/10/2018	0 removidos	2
Remover por leitura completa	13/10/2018	2 removidos	0
Total de artigos incluídos	13/10/2018	0 documentos	

Tabela 3.9: Execução do uso de jogos para o ensino de reutilização de software, base ScienceDirect.

Atividade	Data de execução	Resultado	Número de papers
Primeira execução	14/10/2018 – 10h:07min	10 adicionados	10
Remover conferência/workshops	14/10/2018	0 removidos	10
Remover livros	14/10/2018	1 removidos	9
Remover por título	14/10/2018	5 removidos	4
Remover por abstract	14/10/2018	4 removidos	0
Artigos não encontrados	14/10/2018	0 removidos	0
Artigos para ler	14/10/2018	0 removidos	0
Remover por leitura completa	14/10/2018	0 removidos	0
Total de artigos incluídos	14/10/2018	0 documentos	

Tabela 3.10: Execução do uso de jogos para o ensino de reutilização de software, base IEEEExplore.

Atividade	Data de execução	Resultado	Número de papers
Primeira execução	14/10/2018 – 17h:22min	27 adicionados	27
Remover conferência/workshops	14/10/2018	0 removidos	27
Remover livros	14/10/2018	0 removidos	27
Remover por título	14/10/2018	17 removidos	10
Remover por abstract	14/10/2018	7 removidos	3
Artigos não encontrados	14/10/2018	0 removidos	3
Artigos para ler	15/10/2018	0 removidos	3
Remover por leitura completa	15/10/2018	3 removidos	0
Total de papers incluídos	15/10/2018	0 documentos	

3.4.2 Jogos para o ensino de programação com ênfase em reutilização

3.4.2.1 Execução

Tabela 3.11: Execução do estudo de jogos sobre o ensino de programação com ênfase em reutilização.

Atividade	Data de execução	Resultado	Número de papers
Primeira execução	17/10/2018 – 13h:44min	507 adicionados	507
Artigos repetidos	17/10/2018	6 removidos	501
Artigos em outra língua	17/10/2018	0 removidos	501
Remover conferência/workshops	17/10/2018	16 removidos	485
Remover livros	17/10/2018	0 removidos	485
Remover por título	17/10/2018	368 removidos	99
Remover por abstract	18/10/2018	65 removidos	34
Artigos não encontrados	18/10/2018	0 removidos	34
Remover por leitura completa	19/10/2018 - 25/10/2018	17 removidos	17
Total de papers incluídos	25/10/2018	17 documentos	
Artigos extraídos	25/10/2018 - 05/11/2018	17 documentos	

Tabela 3.12: Execução do estudo de jogos sobre o ensino de programação com ênfase em reutilização, *Snowballing backward*.

Atividade	Data de execução	Resultado	Número de papers
Primeira execução	05/11/2018	389 adicionados	389
Remover por data	05/11/2018	294 removidos	95
Artigos repetidos	05/11/2018	14 removidos	81
Artigos em outra língua	05/11/2018	0 removidos	81
Remover livros	05/11/2018	0 removidos	81
Remover por título	05/11/2018	46 removidos	35
Remover por abstract	07/11/2018	17 removidos	18
Artigos não encontrados	07/11/2018	0 removidos	18
Remover por leitura completa	07/11/2018 - 13/11/2018	12 removidos	6
Total de artigos incluídos	13/11/2018	6 documentos adicionados	
Artigos extraídos	13/11/2018 - 18/11/2018	6 documentos adicionados	

Tabela 3.13: Execução do estudo de jogos sobre o ensino de programação com ênfase em reutilização, *Snowballing forward*.

Atividade	Data de execução	Resultado	Número de papers
Primeira execução	19/11/2018	123 adicionados	123
Remover por data	19/11/2018	1 removidos	122
Artigos repetidos	19/11/2018	16 removidos	107
Artigos em outra língua	19/11/2018	13 removidos	94
Remover livros	19/11/2018	0 removidos	94
Remover por título	19/11/2018	58 removidos	36
Remover por abstract	20/11/2018	18 removidos	18
Artigos não encontrados	20/11/2018	0 removido	18
Artigos para ler	20/11/2018	18 removidos	18
Remover por leitura completa	20/11/2018 - 26/11/2018	15 removidos	3
Total de artigos incluídos	26/11/2018	3 documentos adicionados	
Artigos extraídos	26/11/2018 - 28/11/2018	3 documentos adicionados	

3.4.2.2 Conjunto final de artigos analisados

Tabela 3.14: Conjunto de artigos analisados sobre jogos para o ensino de programação com ênfase em reutilização.

Autor	Título	Ano	Origem
Marimuthu and Govender	<i>Perceptions of Scratch programming among secondary school students in KwaZulu-Natal, South Africa</i>	2018	SF
Agalbato and Loiacono	<i>Robo3: A Puzzle Game to Learn Coding</i>	2018	EP
Topalli and Cagiltay	<i>Improving programming skills in engineering education through problem-based game projects with Scratch</i>	2018	EP
Law	<i>Introducing novice programmers to functions and recursion using computer games</i>	2018	EP
Chandrashekar et al.	<i>Introducing programming using “scratch” and “greenfoot”</i>	2018	EP
Boldbaatar and	<i>Developing Educational 3D Games With StarLogo: The Role of Backwards Fading in</i>	2018	EP

Şendurur	<i>the Transfer of Programming Experience</i>		
Pellas and Vosinakis, S.	<i>Learning to think and practice computationally via a 3D simulation game</i>	2018	EP
Agalbato	<i>Design and implementation of Robo3 : an applied game for teaching introductory programming</i>	2017	SF
Duarte et al.	<i>A cross-cultural review of lightbot for introducing functions and code reuse</i>	2017	SF
Daungcharone et al.	<i>Using Digital Game as Compiler to Motivate C Programming Language Learning in Higher Education</i>	2017	EP
Vincur et al.	<i>Cubely: Virtual reality block-based programming environment</i>	2017	EP
Saito et al.	<i>Analysis of the learning effects between text-based and visual-based beginner programming environments</i>	2017	EP
Yamashita et al.	<i>Visual programming language for model checkers based on google blockly</i>	2017	EP
Sierra et al.	<i>Educational resource based on games for the reinforcement of engineering learning programming in mobile devices</i>	2016	EP
López et al.	<i>Teaching abstraction, function and reuse in the first class of CSI - A lightbot experience</i>	2016	EP
Tabet et al.	<i>From Alice to Python Introducing text-based programming in middle schools</i>	2016	SB
Sáez-López et al.	<i>Visual programming languages integrated across the curriculum in elementary school: A two year case study using Scratch” in five schools</i>	2016	SB
Serrano-Laguna et al.	<i>Building a Scalable Game Engine to Teach Computer Science Languages</i>	2015	EP
Jordine et al.	<i>A mobile-device based serious gaming approach for teaching and learning Java programming</i>	2015	EP
Vaca-Cárdenas et al.	<i>Coding with Scratch: The design of an educational setting for Elementary pre-service teachers</i>	2015	SB
Bau	<i>Droplet, a Blocks-based Editor for Text Code</i>	2015	SB
Bau	<i>Integrating Droplet into Applab – Improving the usability of a blocks-based text edit</i>	2015	SB
Pellas	<i>The development of a virtual learning platform for teaching concurrent programming languages in secondary education: The use of</i>	2014	EP

	<i>open Sim and Scratch4OS</i>		
Chang	<i>Effects of using Alice and Scratch in an introductory programming course for corrective instruction</i>	2014	EP
Lee et al.	<i>A structured approach to teaching recursion using cargo-bot</i>	2014	EP
Kalelioglu and gulbahar	<i>The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners, Perspective, Informatics in Education</i>	2014	SB

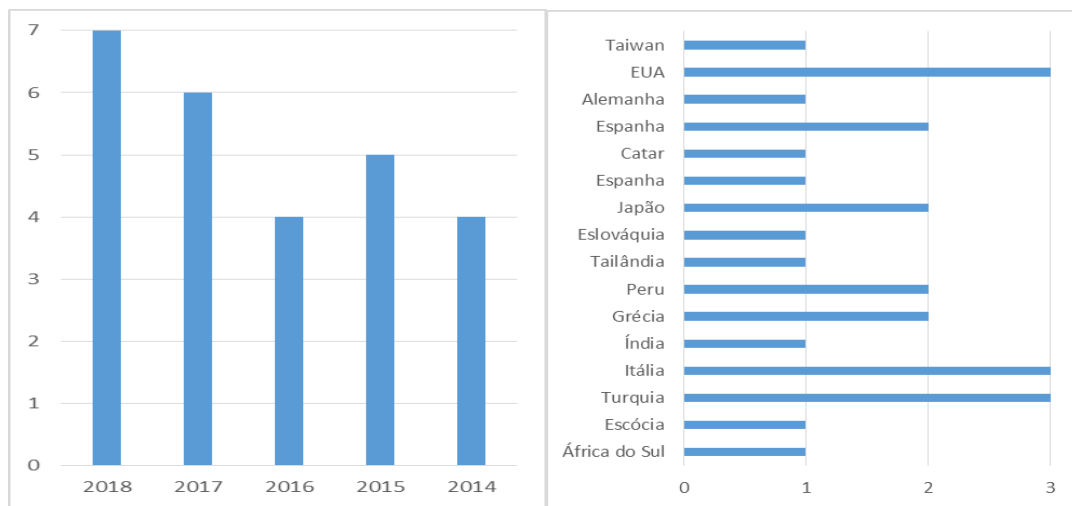


Figura 3.3: Demonstrativo do número de artigos por ano e local de publicação sobre jogos para o ensino de programação com ênfase em reutilização. Fonte: do autor.

3.4.2.3 Matriz de rastreabilidade

Tabela 3.15: Matriz de rastreabilidade sobre jogos para o ensino de programação com ênfase em reutilização.

	Q1	Q2	Q3	Q4	Q5
<i>A cross-cultural review of lightbot for introducing functions and code reuse</i>	X		X		X
<i>A mobile-device based serious gaming approach for</i>	X		X		

<i>teaching and learning Java programming</i>					
<i>A structured approach to teaching recursion using cargo-bot</i>	X		X	X	
<i>Analysis of the learning effects between text-based and visual-based beginner programming environments</i>	X		X	X	
<i>Building a Scalable Game Engine to Teach Computer Science Languages</i>	X		X	X	
<i>Coding with Scratch: The design of an educational setting for Elementary pre-service teachers</i>	X		X	X	X
<i>Cubely: Virtual reality block-based programming environment</i>	X		X	X	
<i>Design and implementation of Robo3 : an applied game for teaching introductory programming</i>	X		X	X	
<i>Developing Educational 3D Games With StarLogo: The Role of Backwards Fading in the Transfer of Programming Experience</i>	X	X	X	X	X
<i>Droplet, a Blocks-based Editor for Text Code</i>	X		X	X	
<i>Educational resource based on games for the reinforcement of engineering learning programming in mobile devices</i>	X		X	X	
<i>Effects of using Alice and Scratch in an introductory programming course for corrective instruction</i>	X		X	X	X
<i>From Alice to Python Introducing text-based programming in middle schools</i>	X		X	X	X
<i>Improving programming skills in engineering education through problem-based game projects with Scratch</i>	X		X	X	
<i>Integrating Droplet into Applab – Improving the usability of a blocks-based text edit</i>	X		X	X	X
<i>Introducing novice programmers to functions and recursion using computer games</i>	X		X	X	
<i>Introducing programming using “scratch” and “greenfoot”</i>	X		X	X	X
<i>Learning to think and practice computationally via a 3D simulation game</i>	X		X	X	X
<i>Perceptions of Scratch programming among secondary school students in KwaZulu-Natal, South Africa</i>	X		X	X	
<i>Robo3: A Puzzle Game to Learn Coding</i>	X	X	X	X	X
<i>Teaching abstraction, function and reuse in the first class of CS1 - A lightbot experience</i>	X		X	X	
<i>The development of a virtual learning platform for teaching concurrent programming languages in secondary education: The use of open Sim and Scratch4OS</i>	X	X	X	X	X
<i>The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners,</i>	X		X	X	

<i>Perspective, Informatics in Education</i>					
<i>Using Digital Game as Compiler to Motivate C Programming Language Learning in Higher Education</i>	X		X	X	
<i>Visual programming language for model checkers based on google blockly</i>	X		X	X	X
<i>Visual programming languages integrated across the curriculum in elementary school: A two year case study using Scratch” in five schools</i>	X		X	X	X

3.4.3 Uso de jogos para o ensino de engenharia de software (mapeamento terciário)

3.4.3.1 Execução

Tabela 3.16: Execução do estudo sobre uso de jogos para o ensino de engenharia de software (mapeamento terciário).

Atividade	Data de execução	Resultado	Número de papers
Primeira execução	30/11/2018 – 18h:24min	36 adicionados	36
Artigos repetidos	30/11/2018	2 removidos	34
Artigos em outra língua	30/11/2018	1 removido	33
Remover conferência / workshops	30/11/2018	6 removidos	27
Remover livros	30/11/2018	0 removidos	27
Remover por título	01/12/2018	10 removidos	17
Remover por abstract	01/12/2018	8 removidos	9
Artigos não encontrados	01/12/2018	0 removidos	9
Artigos para ler	01/12/2018	0 removidos	9
Remover por leitura completa	01/12/2018 - 07/12/2018	2 removidos	7
Total de artigos incluídos	07/12/2018	7 documentos	
Artigos extraídos	07/11/2018 - 12/12/2018	7 documentos	

3.4.3.2 Conjunto final de artigos analisados

Tabela 3.17: Conjunto de artigos analisados sobre uso de jogos para o ensino de engenharia de software (mapeamento terciário).

Autor	Título	Ano
Alhammad and Moreno	<i>Gamification in software engineering education: A systematic mapping</i>	2018
Mauricio et al	<i>A systematic mapping study on game-related methods for software engineering education</i>	2018
Souza et al.	<i>Games for learning: Bridging game-related education methods to software engineering knowledge areas</i>	2017
Kosa et al.	<i>Software engineering education and games: A systematic literature review</i>	2016
Jiang et al.	<i>Process simulation for software engineering education</i>	2015
Marques et al	<i>A systematic mapping study on practical approaches to teaching software engineering</i>	2015
Caulfield et al.	<i>A systematic survey of games used for software engineering education</i>	2011

3.4.3.3 Matriz de rastreabilidade

Tabela 3.18: Matriz de rastreabilidade dos artigos sobre uso de jogos para o ensino de engenharia de software (mapeamento terciário).

	Q1	Q2	Q3	Q4	Q5	Q6
<i>A systematic mapping study on game-related methods for software engineering education</i>	X	X	X	X	X	X
<i>A systematic mapping study on practical approaches to teaching software engineering</i>	X					X
<i>A Systematic Survey of Games Used for Software Engineering Education</i>	X	X	X	X	X	X
<i>Games for learning: Bridging game-related education methods to software engineering knowledge areas</i>	X	X	X			X
<i>Gamification in software engineering education A systematic mapping</i>	X	X	X	X	X	X
<i>Process simulation for software engineering education</i>	X	X	X	X	X	X
<i>Software Engineering Education and Games A Systematic Literature Review</i>			X	X		X

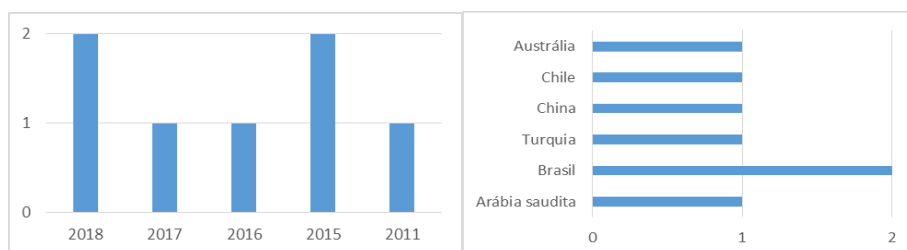


Figura 3.4: Demonstrativo do número de artigos por ano e local de publicação sobre uso de jogos para o ensino de engenharia de software (mapeamento terciário). Fonte: do autor.

3.4.4 Uso de jogos para o ensino de engenharia de software

3.4.4.1 Execução

Tabela 3.19: Execução do estudo sobre uso de jogos para o ensino de engenharia de software.

Atividade	Data de execução	Resultado	Número de papers
Primeira execução	15/12/2018 – 14h:09min	1229 adicionados	1229
Artigos repetidos	15/12/2018	32 removidos	1197
Artigos em outra língua	15/12/2018	12 removido	1185
Remover conferência / workshops	15/12/2018	57 removidos	1128
Remover livros	15/12/2018	43 removidos	1085
Remover por título	15/12/2018 - 16/12/2018	638 removidos	447
Remover por abstract	16/12/2018 - 19/12/2018	379 removidos	68
Artigos não encontrados	14/12/2018	0 removidos	68
Artigos para ler	14/12/2018	68 removidos	68
Remover por leitura completa	14/12/2018 - 03/01/2019	31 removidos	37
Total de papers incluídos	03/01/2019	37 documentos	
Artigos extraídos	03/01/2019 - 25/01/2019	37 documentos	

Tabela 3.20: Execução do estudo sobre uso de jogos para o ensino de engenharia de software, *Snowballing backward*.

Atividade	Data de execução	Resultado	Número de papers
Primeira execução	25/01/2019	382 adicionados	382
Artigos repetidos	25/01/2019	36 removidos	346
Artigos em outra língua	25/01/2019	17 removidos	329
Remover conferência/workshops	25/01/2019	0 removidos	329
Remover livros	26/01/2019	21 removidos	308
Remover por título	28/01/2019	236 removidos	72
Remover por abstract	30/01/2019	64 removidos	8
Artigos não encontrados	30/01/2019	1 removidos	7
Artigos para ler	30/01/2019	0 removidos	7
Remover por leitura completa	30/01/2019 - 14/02/2019	5 removidos	2
Total de artigos incluídos	14/02/2019	2 documentos adicionados	
Artigos extraídos	14/02/2019	2 documentos adicionados	

Tabela 3.21: Execução do estudo sobre uso de jogos para o ensino de engenharia de software, *Snowballing forward*.

Atividade	Data de execução	Resultado	Número de papers
Primeira execução	16/02/2019	311 adicionados	311
Artigos repetidos	16/02/2019	78 removidos	233
Artigos em outra língua	16/02/2019	51 removidos	182
Remover conferência/workshops	16/02/2019	0 removidos	182
Remover livros	16/02/2019	4 removidos	178
Remover por título	16/02/2019	112 removidos	66
Remover por abstract	16/02/2019	51 removidos	15
Artigos não encontrados	16/02/2019	2 removido	13
Artigos para ler	16/02/2019	0 removidos	13
Remover por leitura completa	16/02/2019 - 02/03/2019	8 removidos	5
Total de artigos incluídos	02/03/2019	5 documentos adicionados	
Artigos extraídos	02/03/2019 - 08/03/2019	5 documentos adicionados	

3.4.4.2 Conjunto final de artigos analisados

Tabela 3.22: Conjunto de artigos analisados sobre uso de jogos para o ensino de engenharia de software.

Autor	Título	Ano	Origem
Calderón et al.	<i>A serious game to support the ISO 21500 standard education in the context of software project management</i>	2018	EP
Morales-Trujillo et al.	<i>What can go wrong in a software project? Have fun solving it</i>	2018	EP
Dalpiaz and Cooper	<i>Games for Requirements Engineers: Analysis and Directions</i>	2018	EP
Morschheuse et al.	<i>How to design gamification? A method for engineering gamified software</i>	2018	EP
Larenas et al.	<i>Classutopia: A serious game for conceptual modeling design</i>	2018	EP
Monsalve et al.	<i>SimulES-W: A collaborative game to improve software engineering teaching</i>	2018	EP
Garcia et al.	<i>Experiences of using a game for improving learning in software requirements elicitation</i>	2018	EP
Vasconcelos et al.	<i>Gamification Applied in the Teaching of Agile Scrum Methodology</i>	2018	EP
Jesus et al.	<i>Gamification in Software Testing: A Characterization Study</i>	2018	SF
Souza et al.	<i>Gamification in Software Engineering Education: An Empirical Study</i>	2017	EP
Aydan et al.	<i>Teaching ISO/IEC 12207 software lifecycle processes: A serious game approach</i>	2017	EP
Calderón et al.	<i>Integrating serious games as learning resources in a software project management course: The case of ProDec</i>	2017	EP
Matsubara and Da Silva	<i>Game elements in a software engineering study group: A case study</i>	2017	EP
Clegg et al.	<i>Teaching software testing concepts using a mutation testing game</i>	2017	EP
Pieper et al.	<i>A case study of software engineering methods education supported by digital game-based learning: Applying the SEMAT Essence kernel in games and course projects</i>	2017	EP

Milosz and Milosz	<i>Developing and implementation of decision-making games for business education of engineering students</i>	2017	EP
Lopes and Petrillo	<i>Simkan: Training kanban practices through stochastic simulation</i>	2017	EP
Herranz et al.	<i>Deploying a gamification framework for software process improvement: Preliminary results</i>	2017	EP
Calderón et al.	<i>Coverage of ISO/IEC 29110 project management process of basic profile by a serious game</i>	2017	SF
Calderón et al.	<i>Coverage of the ISO 21500 Standard in the Context of Software Project Management by a Simulation-Based Serious Game</i>	2017	SF
Fraser	<i>Gamification of software testing</i>	2017	SF
Maxim et al.	<i>An agile software engineering process improvement game</i>	2016	EP
Valencia et al.	<i>GSDgame: A serious game for the acquisition of the competencies needed in GSD</i>	2016	EP
Su	<i>The effects of students' motivation, cognitive load and learning anxiety in gamification software engineering education: a structural equation modeling study</i>	2016	EP
Ramirez-Rosales et al.	<i>A serious game to promote object oriented programming and software engineering basic concepts learning</i>	2016	EP
Jimenez-Hernandez et al.	<i>Methodology to construct educational video games in software engineering</i>	2016	EP
Fuchs and Wolff	<i>Improving programming education through gameful, formative feedback</i>	2016	EP
Calderón and Ruiz	<i>Coverage of ISO/IEC 12207 software lifecycle process by a simulation-based serious game</i>	2016	EP
Unkelos-Shpigel	<i>Ace that game: Educating students to gamified design thinking</i>	2016	EP
Wee Leong	<i>SCRUM-X: An interactive and experiential learning platform for teaching scrum</i>	2016	SB
Albilali and Qureshi	<i>Proposal to Teach Software Development Using Gaming Technique</i>	2016	SF
Lino et al.	<i>Project management game 2D (PMG-2D): A serious game to assist software project managers training</i>	2015	EP
Peixoto et al.	<i>The issues of adopting simulation games in software engineering classes</i>	2015	EP
Chaves et	<i>Experimental evaluation of a serious game for</i>	2015	EP

al.	<i>teaching software process modeling</i>		
Uskov and Sekar	<i>Gamification of software engineering curriculum</i>	2015	EP
Uyaguari et al.	<i>Gamification proposal for a software engineering risk management course</i>	2015	EP
Dlamini	<i>Examining the effectiveness of 2D and 3D online environment in enhancing students' learning of software engineering</i>	2015	EP
Atal and Sureka	<i>Anukarna: A software engineering simulation game for teaching practical decision making in peer code review</i>	2015	EP
Laskowski	<i>Implementing gamification techniques into university study path - A case study</i>	2015	EP
Pötter et al.	<i>InspectorX: A game for software inspection training and learning</i>	2014	EP
Qu et al.	<i>Research on teaching gamification of software engineering</i>	2014	EP
Szabo	<i>Evaluating GameDevTycoon for teaching Software Engineering</i>	2014	EP
Nassal	<i>A general framework for software project management simulation games</i>	2014	EP
Telukunta et al.	<i>StrateJect: An Interactive Game for Project Management Experiential Learning</i>	2014	SB

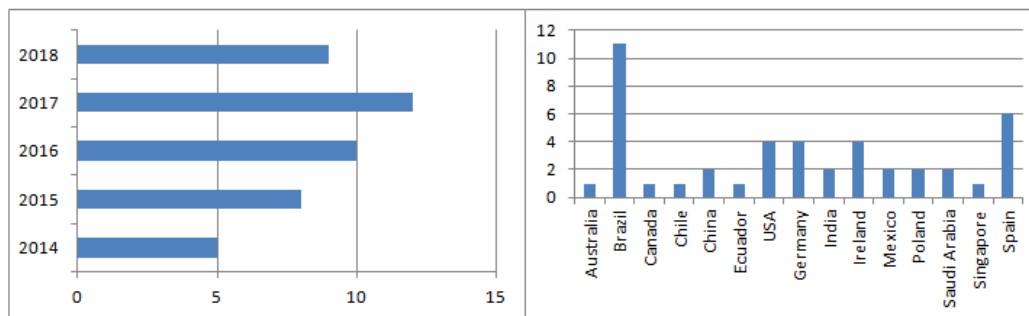


Figura 3.5: Demonstrativo do número de artigos por ano e local de publicação sobre uso de jogos para o ensino de engenharia de software. Fonte: do autor.

3.4.4.3 Matriz de rastreabilidade

Tabela 3.23: matriz de rastreabilidade sobre uso de jogos para o ensino de engenharia de software.

	Q1	Q2	Q3	Q4	Q5
<i>A case study of software engineering methods education supported by digital game-based learning: Applying the SEMAT Essence kernel in games and course projects</i>	X	X		X	X
<i>A general framework for software project management simulation games</i>	X	X		X	
<i>A serious game to promote object oriented programming and software engineering basic concepts learning</i>	X	X		X	X
<i>A serious game to support the ISO 21500 standard education in the context of software project management</i>	X	X		X	X
<i>Ace that game: Educating students to gamified design thinking</i>	X	X		X	X
<i>An agile software engineering process improvement game</i>	X	X		X	X
<i>Anukarna: A software engineering simulation game for teaching practical decision making in peer code review</i>		X		X	X
<i>Classutopia: A serious game for conceptual modeling design</i>	X	X		X	X
<i>Coverage of ISO/IEC 12207 software lifecycle process by a simulation-based serious game</i>	X	X		X	
<i>Coverage of ISO/IEC 29110 project management process of basic profile by a serious game</i>	X	X		X	X
<i>Coverage of the ISO 21500 Standard in the Context of Software Project Management by a Simulation-Based Serious Game</i>	X	X		X	
<i>Deploying a gamification framework for software process improvement: Preliminary results</i>	X	X	X	X	
<i>Developing and implementation of decision-making games for business education of engineering students</i>	X	X		X	X
<i>Evaluating GameDevTycoon for teaching Software Engineering</i>	X	X		X	X
<i>Examining the effectiveness of 2D and 3D online environment in enhancing students' learning of software engineering</i>	X	X		X	X
<i>Experiences of using a game for improving learning in software requirements elicitation</i>	X	X		X	
<i>Experimental evaluation of a serious game for teaching software process modeling</i>	X	X		X	X
<i>Game elements in a software engineering study group: A</i>	X	X		X	X

<i>case study</i>					
<i>Games for Requirements Engineers: Analysis and Directions</i>	X	X		X	X
<i>Gamification Applied in the Teaching of Agile Scrum Methodology</i>	X	X		X	X
<i>Gamification in Software Engineering Education: An Empirical Study</i>	X	X		X	X
<i>Gamification in Software Testing: A Characterization Study</i>	X	X		X	
<i>Gamification of software engineering curriculum</i>	X	X	X	X	X
<i>Gamification of software testing</i>	X	X		X	
<i>Gamification proposal for a software engineering risk management course</i>	X	X		X	
<i>GSDgame: A serious game for the acquisition of the competencies needed in GSD</i>	X	X	X	X	
<i>How to design gamification? A method for engineering gamified software</i>	X	X	X	X	X
<i>Implementing gamification techniques into university study path - A case study</i>	X	X		X	X
<i>Improving programming education through gameful, formative feedback</i>	X	X		X	
<i>InspectorX: A game for software inspection training and learning</i>	X	X		X	X
<i>Integrating serious games as learning resources in a software project management course: The case of ProDec</i>	X	X		X	X
<i>Methodology to construct educational video games in software engineering</i>	X	X		X	X
<i>Project management game 2D (PMG-2D): A serious game to assist software project managers training</i>	X	X	X	X	X
<i>Proposal to Teach Software Development Using Gaming Technique</i>	X	X		X	X
<i>Research on teaching gamification of software engineering</i>	X	X		X	
<i>SCRUM-X: An interactive and experiential learning platform for teaching scrum</i>	X	X		X	X
<i>Simkan: Training kanban practices through stochastic simulation</i>	X	X		X	X
<i>SimulES-W: A collaborative game to improve software engineering teaching</i>		X		X	X
<i>StrateJect: An Interactive Game for Project Management Experiential Learning</i>		X		X	X
<i>Teaching ISO/IEC 12207 software lifecycle processes: A serious game approach</i>	X	X		X	X
<i>Teaching software testing concepts using a mutation testing game</i>	X	X		X	

<i>The effects of students' motivation, cognitive load and learning anxiety in gamification software engineering education: a structural equation modeling study</i>	X	X	X	X	X
<i>The issues of adopting simulation games in software engineering classes</i>	X	X		X	X
<i>What can go wrong in a software project? Have fun solving it</i>	X	X	X	X	X

3.4.5 Uso de jogos para o ensino de reutilização de software

3.4.5.1 Execução

Tabela 3.24: Execução do estudo sobre uso de jogos para o ensino de reutilização de software.

Atividade	Data de execução	Resultado	Número de papers
Primeira execução	11/03/2019 – 22h:09min	393 adicionados	393
Artigos repetidos	12/03/2019	4 removidos	389
Artigos em outra língua	12/03/2019	0 removido	389
Remover conferência / workshops	12/03/2019	36 removidos	353
Remover livros	12/03/2019	13 removidos	340
Remover por título	12/03/2019 - 13/03/2019	177 removidos	163
Remover por abstract	14/03/2019 - 21/03/2019	137 removidos	26
Artigos não encontrados	21/03/2019	3 removidos	23
Artigos para ler	21/03/2019	0 removidos	23
Remover por leitura completa	21/03/2019 - 05/04/2019	8 removidos	15
Total de papers incluídos	05/04/2019	15 documentos	
Artigos extraídos	05/04/2019 - 12/04/2019	15 documentos	

Tabela 3.25: Execução do estudo sobre uso de jogos para o ensino de reutilização de software, *Snowballing backward*.

Atividade	Data de execução	Resultado	Número de papers
Primeira execução	14/04/2019	262 adicionados	262
Artigos repetidos	14/04/2019	44 removidos	218
Artigos em outra língua	14/04/2019	0 removidos	218
Remover conferência/workshops	14/04/2019	0 removidos	218
Remover livros	14/04/2019	3 removidos	215
Remover por título	14/04/2019	109 removidos	106
Remover por abstract	14/04/2019 - 18/04/2019	95 removidos	11
Artigos não encontrados	18/04/2019	3 removidos	8
Artigos para ler	18/04/2019	0 removidos	8
Remover por leitura completa	18/04/2019 - 23/04/2019	6 removidos	2
Total de artigos incluídos	23/04/2019	2 documentos adicionados	
Artigos extraídos	23/04/2019 - 27/04/2019	2 documentos adicionados	

Tabela 3.26: Execução do estudo sobre uso de jogos para o ensino de reutilização de software, *Snowballing forward*.

Atividade	Data de execução	Resultado	Número de papers
Primeira execução	28/04/2019	112 adicionados	112
Artigos repetidos	28/04/2019	4 removidos	108
Artigos em outra língua	28/04/2019	12 removidos	96
Remover conferência/workshops	28/04/2019	0 removidos	96
Remover livros	29/04/2019	0 removidos	96
Remover por título	29/04/2019	66 removidos	30
Remover por abstract	29/04/2019	24 removidos	6
Artigos não encontrados	29/04/2019	1 removido	5
Artigos para ler	29/04/2019	0 removidos	5
Remover por leitura completa	29/04/2019 - 03/05/2019	2 removidos	3
Total de artigos incluídos	03/05/2019	3 documentos adicionados	
Artigos extraídos	04/05/2019 - 07/05/2019	3 documentos adicionados	

3.4.5.2 Conjunto final de artigos analisados

Tabela 3.27: Conjunto de artigos analisados sobre uso de jogos para o ensino de reutilização de software.

Autor	Título	Ano	Origem
Liu et al.	Incorporating service-oriented programming techniques into undergraduate CS and SE curricula	2013	EP
Niu et al.	Reuse a "software reuse" course	2011	EP
Lisboa et al.	A case study in software product lines: An educational experience	2008	EP
Caspersen and Christensen	Frameworks in teaching	2008	EP
Clúa et al.	Work in progress - Evolution of code reusing practices during a sequence of three CS courses	2007	EP
Ceddia et al.	W.I.E.R - Implementing artifact reuse in an educational environment with real projects-work in progress	2001	EP
Wang et al.	Teaching software reuse with JavaBeans	2000	EP
Nada et al.	Software reuse technology practices and assessment tool-kit	2000	EP
Mili et al.	Toward an engineering discipline of software reuse	1999	EP
Lattanzi and Henry	Teaching the Object-Oriented Paradigm and Software Reuse: Notes from an Empirical Study	1996	SF
Tewari	Software Reuse and Object-Oriented Software Engineering in the Undergraduate Curriculum	1995	EP
Smith	CARDS training: Transferring reuse knowledge	1995	EP
Tewari and Gitlin	On object-oriented libraries in the undergraduate curriculum: importance and effectiveness	1994	SF
Ramakrishnan et al.	Planned software reuse in object-oriented software engineering education	1994	SF
Cardow and Watson	A practical approach to teaching software reuse	1994	EP
Sitaraman and Gray	Software reuse: A context for introducing software engineering principles in a traditional computer science second	1993	EP

	Course		
Tewari and Friedman	A Framework for Incorporating Object-Oriented Software Engineering in the Undergraduate Curriculum	1993	EP
Frakes	A graduate course on software reuse, domain analysis, and re-engineering	1993	SB
Gray	Teaching the second computer science course in a reuse-based setting: A sequence of laboratory assignments in ada	1993	SB
Sindre et al.	Software reuse in an educational perspective	1992	EP

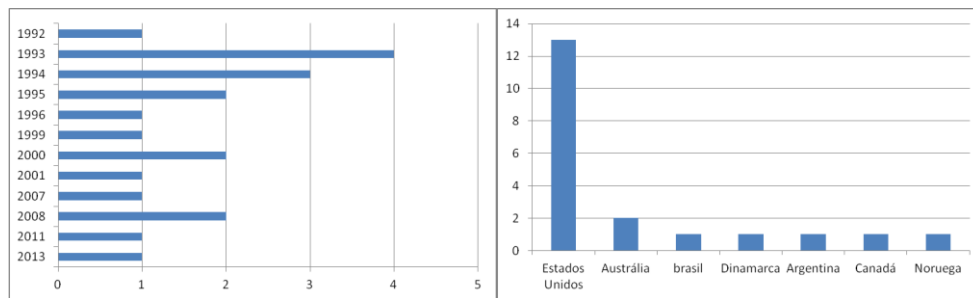


Figura 3.6: Demonstrativo do número de artigos por ano e local de publicação sobre uso de jogos para o ensino de reutilização de software. Fonte: do autor.

3.4.5.3 Matriz de rastreabilidade

Tabela 3.28: Matriz de rastreabilidade dos artigos sobre o ensino de reutilização de software.

	Q1	Q2	Q3	Q4	Q5
<i>A case study in software product lines: An educational experience</i>		X	X	X	X
<i>A Framework for Incorporating Object-Oriented Software Engineering in the Undergraduate Curriculum</i>	X	X		X	
<i>A graduate course on software reuse, domain analysis, and re-engineering</i>				X	X
<i>A practical approach to teaching software reuse</i>		X	X	X	X
<i>CARDS training: Transferring reuse knowledge</i>		X		X	X
<i>Frameworks in teaching</i>		X		X	
<i>Incorporating service-oriented programming techniques into undergraduate CS and SE curricula</i>				X	X
<i>On object-oriented libraries in the undergraduate curriculum: importance and effectiveness</i>	X	X	X	X	X
<i>Planned software reuse in object-oriented software engineering education</i>		X	X	X	X
<i>Reuse a "software reuse" course</i>	X	X	X	X	X
<i>Software Reuse and Object-Oriented Software Engineering in the Undergraduate Curriculum</i>		X		X	X
<i>Software reuse in an educational perspective</i>		X	X	X	X
<i>Software reuse technology practices and assessment tool-kit</i>		X	X	X	
<i>Software reuse: A context for introducing software engineering principles in a traditional computer science second Course</i>	X	X		X	
<i>Teaching software reuse with JavaBeans</i>	X	X	X	X	X
<i>Teaching the Object-Oriented Paradigm and Software Reuse: Notes from an Empirical Study</i>		X	X	X	
<i>Teaching the second computer science course in a reuse-based setting: A sequence of laboratory assignments in ada</i>	X	X		X	X
<i>Toward an engineering discipline of software reuse</i>	X	X	X		X
<i>W.I.E.R - Implementing artifact reuse in an educational environment with real projects-work in progress</i>	X		X		
<i>Work in progress - Evolution of code reusing practices during a sequence of three CS courses</i>	X			X	X

3.5 Resumo das descobertas

A partir dos mapeamentos realizados, várias referências foram encontradas. Para que fosse possível fazer uma rastreabilidade entre os mapeamentos e as referências que são citadas, cada um dos mapeamentos recebeu um identificador diferente que é utilizado em sua citação. A referência acontece de forma que todas as referências que vierem antes do identificador apresentado fazem parte do estudo que é demonstrado a seguir.

Jogos para o ensino de programação com ênfase em reutilização: *

Uso de jogos para o ensino de engenharia de software (map. terciário): **

Uso de jogos para o ensino de engenharia de software: ***

Ensino de reutilização de software: ****

3.5.1 Qual(s) a(s) principal(s) dificuldade(s) do ensino de reutilização de software?

Em uma pesquisa realizada em 1993, entre 113 entrevistados de 29 organizações, principalmente nos EUA, apenas 13% disseram que aprenderam sobre a reutilização em seus cursos de graduação. Apesar de suas vantagens e promessas, a reutilização de software não se tornou uma prática padrão para a construção de software, pois dentre as muitas causas deste fracasso, **a falta de ensino é considerada uma das mais importantes** (NIU, 2011) ****. Observando o estado atual da reutilização, acredita-se que esse número de apenas 13% tenha aumentado bastante, devido às novas ferramentas que hoje estão sendo utilizadas. Entretanto, com base nos anos dos artigos que foram encontrados e analisados no presente estudo, é possível observar que mais da metade deles foram publicados até o ano 2000 e o artigo mais atual que foi analisado foi publicado em 2013. A partir dessa análise e considerando o estado atual da reutilização, é possível inferir que o uso da reutilização pode ter aumentado, mas pouco se tem falado da mesma. Devido a esses números, acredita-se que hoje em dia **a maioria das pessoas faz reutilização, mas não de uma maneira sistemática e pensada**, as pessoas simplesmente reutilizam de suas maneiras.

De acordo com o estudo que foi realizado por LISBOA *et al.* (2008) ****, que visava explicar reutilização de software dentro de uma contexto de linha de produto, foi possível observar que, em geral, os alunos tinham a mesma expectativa sobre o curso: entender os conceitos teóricos de reutilização de software e depois aplicar esses conceitos em uma **experiência prática**. No entanto, alguns alunos não cumpriram todas as suas expectativas, devido ao **tempo restrito, a teoria não poderia ser muito detalhada** e a **fase prática deveria levar mais tempo** para obter um resultado melhor (LISBOA *et al.*, 2008) ****. Por outro lado, os alunos foram suficientemente desafiados, mas a maioria concluiu o projeto com sucesso. No entanto, foi observado que muitos alunos reclamaram do **material teórico extenso** (LATTANZI & HENRY, 1996) (LISBOA *et al.*, 2008) ****.

A reutilização é **frequentemente proposta e muitas vezes falha**. A principal dificuldade apontada por eles é que, para aprender sobre reutilização, deve-se impor a noção **de ter componentes disponíveis** para suportar o desenvolvimento do sistema. Entretanto, esses **componentes não estão sempre disponíveis** na mão de um professor para que o mesmo possa ensinar. Outra dificuldade apontada por eles é que uma reutilização sem análise de domínio oferecerá pouco ganho em produtividade (CARDOW & D., 1994) ****. Portanto, para que a reutilização de software funcione, **a ênfase deve estar no desenvolvimento do sistema, não no desenvolvimento de reutilização**, normalmente os desenvolvedores fazem a reutilização somente por reutilizar e não com uma análise prévia. O ensino de reutilização de software em geral enfrenta problemas relacionados a: **falta de suporte de ferramentas; modelos inadequados de comportamento de componentes; problemas de seleção de idioma; dificuldade de encontrar software reutilizável e dificuldade de desenvolver projetos e tarefas do curso** (WANG & WANG, 2000) ****.

Uma dificuldade bastante recorrente é a **falta de boas ferramentas** de desenvolvimento na academia. Um ambiente de desenvolvimento voltado para a reutilização pode fazer diferença. Os benefícios da reutilização não serão totalmente percebidos até que ferramentas adequadas sejam usadas com foco na reutilização de

software. A sobrecarga de reutilização ainda é alta para convencer os programadores de que vale a pena (LATTANZI & HENRY, 1996) ****.

Outro problema associado ao ensino de engenharia de software em geral é a **dificuldade de simular realisticamente desenvolvimentos de projetos de força industrial** em grande escala em um ambiente educacional (RAMAKRISHNAN *et al.*, 1994) ****. Muitos cursos são dados com projetos fictícios para que o aluno consiga desenvolver algum protótipo ao longo do curso. Entretanto, ao chegar ao mercado de trabalho, os problemas ganham outras proporções e esses alunos muitas vezes não estão preparados por causa da **falta de treinamento adequado** nos cursos de graduação atuais.

Um exercício prático de reutilização que não seja precedido por uma introdução teórica completa provavelmente falhará. Portanto, **a prática da reutilização deve vir acompanhada previamente de uma introdução teórica**. De fato, a prática seria boa para realmente ensinar reutilização, ao invés de apenas discutir sobre isso (SINDRE *et al.*, 1992) ****. No entanto, existem alguns problemas em colocar a reutilização em prática na educação como, por exemplo, **o ciclo completo de reutilização leva muito tempo, a necessidade de uma biblioteca de componentes, a necessidade da simulação de um problema real**, etc.

Há outra série de fatores que dificultam a reutilização direta de código no curso de graduação como, por exemplo, **existe uma rápida taxa de mudança nas ferramentas e nas linguagens de programação** que são utilizadas (CEDDIA *et al.*, 2001) ****. Por isso, um curso de reutilização de software deve ser criado para ser o mais flexível possível, para que seja possível alterar suas propriedades internas de uma maneira fácil e sem a necessidade de um alto custo com novas ferramentas.

Por fim, **problemas de direitos autorais** também podem restringir o uso de código. Apesar de existirem diversos códigos *open source* em diversas plataformas, ainda é prática padrão existir licenças de uso na maioria dos componentes (CEDDIA *et al.*, 2001) ****. Por este motivo, existe uma barreira do que pode ou não ser reutilizado, restringindo mais ainda a quantidades de componentes que podem ser utilizados em curso.

3.5.2 Qual(s) a(s) definição(s) de jogos foi utilizada?

Os videogames estão em nossas vidas há mais de 50 anos, tornando-se rapidamente uma das maneiras mais importantes, lucrativas e influentes de entretenimento (RAMÍREZ-ROSALES *et al.*, 2016) ***. Um jogo é uma atividade entre dois ou mais tomadores de decisão independentes que buscam atingir seus objetivos em algum contexto limitado. Uma definição mais convencional diria que um jogo é um contexto com regras entre adversários que tentam conquistar objetivos (CALDERÓN *et al.*, 2018) ***. Outra definição utilizada é que um jogo pode ser entendido como uma atividade não séria e voluntária, intencionalmente separada do mundo "cotidiano" (LASKOWSKI, 2015) ***. Uma atividade que absorve o jogador de maneira plena e intensa e não está relacionada a nenhum tipo de benefício material. Ele exige que o jogador jogue em um tempo e lugar específicos e faça isso de acordo com a ordem e regras estabelecidas (LASKOWSKI, 2015) ***.

Na pesquisa que foi realizada, foi possível observar alguns termos que estão diretamente relacionados com a área de jogos e de ensino, dentre eles cabe citar: **jogos sérios, aprendizagem baseada em jogos, gamificação, aprendizagem baseada no desenvolvimento de jogos e simulação.**

A gamificação tem sido considerada uma das principais tecnologias de ensino emergente e hoje em dia é amplamente adotada na educação. Uma das definições encontradas para esse termo é o uso da filosofia, elementos e mecanismos de jogos em ambientes que não sejam jogos, com o intuito de induzir certo comportamento nas pessoas, bem como para melhorar sua motivação e envolvimento em uma tarefa específica (MAURICIO *et al.*, 2017) **. Embora existam muitas definições de gamificação, todas giram em torno de um princípio, o uso de elementos de jogos em outros contextos que não sejam apenas lúdicos (ALHAMMAD & MORENO, 2018) **.

Um dos primeiros jogos sérios a ser desenvolvido foi o Army Battlezone, um projeto desenvolvido pela Atari no início dos anos 80 que foi criado para treinar soldados em um campo de batalha (LINO *et al.*, 2015) ***. Entretanto, existem divergências sobre a criação do primeiro jogo sério onde outras fontes afirmam que a

primeira vez que o termo foi utilizado foi em 2002, em um lançamento de um simulador de treinamento militar chamado “America's Army”, criado pelo exército americano e distribuído gratuitamente pela internet (VALENCIA *et al.*, 2016) ***.

Os jogos sérios são projetados de tal forma que o objetivo principal não é entreter o usuário, mas sim treiná-lo em uma determinada área. Isso não significa que o jogo não possa ser divertido, mas o entretenimento derivado do jogo é projetado para se obter algum resultado, de modo que a experiência de aprendizado do jogador se torne divertida (DUARTE & PEARCE, 2017) * (MAURICIO *et al.*, 2017) (ALHAMMAD & MORENO, 2018) ** (VALENCIA *et al.*, 2016) ***. Jogos sérios são projetados para um propósito diferente do que apenas entreter o usuário (CAULFIELD *et al.*, 2011) ** (LARENAS *et al.*, 2018) (MILOSZ & MILOSZ, 2017) (CALDERÓN *et al.*, 2017a) (ALBILALI & QURESHI, 2016) ***. O adjetivo “sério” indica que seu objetivo é mais do que apenas diversão e que eles são projetados para educar, treinar ou informar os usuários (CALDERÓN & RUIZ, 2016) ***. Os jogos sérios fazem uso do meio artístico dos jogos para transmitir uma mensagem, ensinar uma lição ou fornecer uma experiência (JORDINE *et al.*, 2014) * (MAXIM *et al.*, 2016) ***. Os jogos sérios são projetados com um objetivo diferente do que apenas entretenimento, são ferramentas poderosas que permitem aos participantes experimentar, aprender com seus próprios erros e adquirir experiência, além de ajudar os treinadores a ensinar conhecimento prático em um ambiente livre de riscos (LEE *et al.*, 2014) (SERRANO-LAGUNA *et al.*, 2015) * (CALDERÓN *et al.*, 2017a) ***.

Um jogo sério com um propósito didático é aquele em que se obtém algum tipo de conhecimento por meio do brincar, facilitando o processo de ensino e aprendizagem (AGALBATO, 2017) (AGALBATO & LOIACONO, 2018) (LAW, 2018) * (RAMÍREZ-ROSALES *et al.*, 2016) ***. Essa termo de utilizar jogos com propósitos educativos é conhecido como aprendizagem baseada em jogos, ou mais comumente conhecido pelo seu termo em inglês *Game-Based-Learning* (GBL) (BOLDBAATAR & ŞENDURUR, 2018) * (GARCIA *et al.*, 2019) ***. O GBL pode ser definida como a aplicação dos princípios dos jogos tradicionais em situações da vida real, com o objetivo de obter a atenção dos alunos e incentivar a criação de conhecimento. Assim, o conceito

de GBL se refere amplamente ao uso de jogos para apoiar o ensino e a aprendizagem (MARIMUTHU & GOVENDER, 2018) (BAU, 2015a) (BAU, 2015b) * (GARCIA *et al.*, 2019) (SU, 2016) ***. GBL pode ser compreendido como qualquer utilização de um jogo com propósitos educacionais, entretanto, existe ainda o termo DGBL (*Digital-Game-Based-Learning*) que compreende os jogos educacionais em suas versões digitais (BAU, 2015b) (TABET *et al.*, 2016) * (USKOV & SEKAR, 2014) ***.

Outro técnica de aprendizado através da utilização jogos é a Aprendizagem Baseada no Desenvolvimento de Jogos, que é a prática de desenvolver jogos como experiência de aprendizagem prática (SOUZA *et al.*, 2017a) **. Jogos são um domínio bem conhecido dos alunos, facilitando, assim, a compreensão dos requisitos. Em segundo lugar, os jogos são atraentes, motivadores e envolvem os alunos em seu processo de criação. Finalmente, a complexidade variada dos jogos pode fornecer aos educadores um grande conjunto de opções para tarefas e projetos em sala de aula (SOUZA *et al.*, 2017a) **.

Existe mais um termo que está diretamente ligado com os jogos, esse termo é conhecido como simulação. A simulação pode ser entendida como uma imitação da operação de um processo ou sistema do mundo real ao longo do tempo (JIANG *et al.*, 2015) **, ou seja, é uma imitação de aspectos da realidade, o que permite trabalhar em condições semelhantes às reais, mas com variáveis controladas em um ambiente que se assemelha ao real. Um jogo é diferente de um modelo ou simulação. Um modelo é uma representação menor de uma realidade complexa que reflete certas características selecionadas do sistema que ele representa (CAULFIELD *et al.*, 2011) **. Ele é útil na medida em que retrata com precisão aquelas características que são interessantes num dado momento. Enquanto isso, uma simulação é um tipo especial de modelo que exhibe processos de alguma forma, e mostra como esses processos mudam do estado A para o estado B. Já um jogo é uma simulação que é propositalmente executada sendo total ou parcialmente determinada pelas decisões dos jogadores, dentro de algumas circunstâncias predeterminadas.

Por fim, também foi encontrado um modelo de ensino denominado *e-learning* que se caracteriza por um ensino não presencial apoiado por tecnologias para

comunicação e distribuição de conteúdo. Apesar dele não ser considerado um método de ensino, ele faz uso de alguns métodos que foram citados até aqui para trazer um maior entretenimento para as aulas (USKOV & SEKAR, 2014) ***.

Foram apresentadas muitas definições sobre muitos conceitos sobre a aplicação de jogos em um determinado contexto. Muitos desses conceitos se complementam e acabam trazendo benefícios maiores para o usuário quando usados juntamente. A seguir é demonstrado um diagrama de Venn que foi adaptado do modelo sugerido por Uskov e Sekar (USKOV & SEKAR, 2014) ***, adicionando três elementos no diagrama (gamificação, simulação e GDBL), para assim mostrar as interseções entre todos os conceitos que foram apresentados. Apesar do e-learning não ser considerado um método de ensino apoiado em jogos, o mesmo faz uso de muitas das características dos métodos apresentados nesse trabalho e por isso também será demonstrado no diagrama.

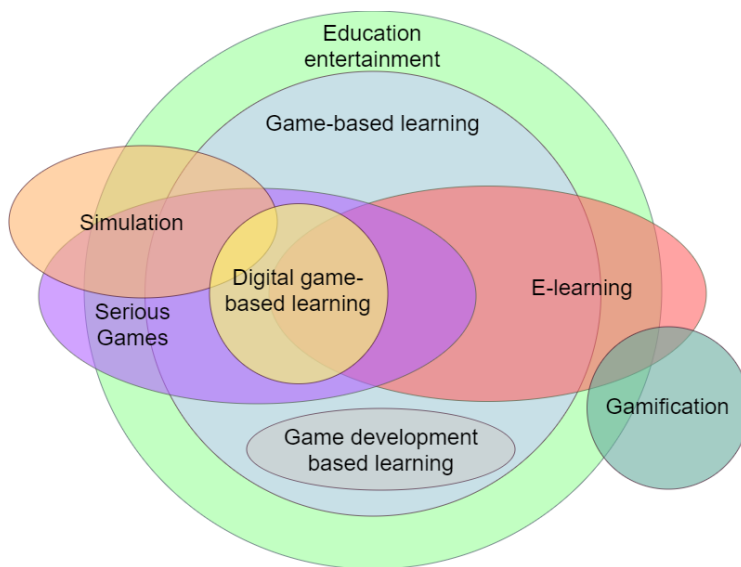


Figura 3.7: Diagrama de Venn sobre jogos na educação, adaptado de USKOV & SEKAR (2014).

3.5.3 Qual(s) a(s) principal(s) vantagem(s) / motivação(s) do uso de jogos para o ensino de engenharia de software?

Usar jogos como uma ferramenta de reforço para ensinar habilidades pode ser uma estratégia muito benéfica para os alunos. Eles têm se mostrado ser uma ferramenta útil para complementar os métodos convencionais de aprendizagem. De fato, os jogos permitem visualizar conceitos que poderiam ser percebidos como abstratos demais. Eles também ajudam a se familiarizar com o conhecimento e os métodos que talvez sejam tediosos para estudar, oferecendo um ciclo de desafios e recompensas que impulsiona a experiência de aprendizado (AGALBATO, 2017) (BOLDBAATAR & ŞENDURUR, 2018) * (VALENCIA *et al.*, 2016) (QU *et al.*, 2014) (JIMÉNEZ-HERNÁNDEZ *et al.*, 2016) ***.

Existem muitos benefícios que estão atrelados à utilização dos jogos na educação, como o aumento da colaboração e competição, a criação de um *feedback* imediato, a possibilidade de uma reflexão dos resultados alcançados e a transferência de conteúdo de maneira que a aprendizagem é parte integrante da jogabilidade, onde o aluno deve usar os conhecimentos adquiridos em aula para resolver algum problema dentro do jogo (JORDINE *et al.*, 2014) (LEE *et al.*, 2014) (CHAVES *et al.*, 2015) * (SOUZA *et al.*, 2017a) ** (PIEPER *et al.*, 2017) ***. Outra vantagem atribuída aos jogos sérios é a capacidade do usuário de assumir papéis reais (SIERRA *et al.*, 2016) ***. Como ferramentas de treinamento, os jogos sérios aumentam o conhecimento conceitual dos alunos, aumentam a conclusão das tarefas, a confiança dos alunos e melhoram a retenção de conhecimento (BOLDBAATAR & ŞENDURUR, 2018) (CHANG, 2014) (DUARTE & PEARCE, 2017) (AGALBATO, 2017) (LAW, 2018) (MARIMUTHU & GOVENDER, 2018) (SIERRA *et al.*, 2016) (VINCUR *et al.*, 2017) (TOPALLI & CAGILTAY, 2018) (CHANDRASHEKAR *et al.*, 2018) (PELLAS & VOSINAKIS, 2017) (MARIMUTHU & GOVENDER, 2018) * (GARCIA *et al.*, 2019) (CALDERÓN & RUIZ, 2016) (POTTER *et al.*, 2014) ***.

A educação em ES carece de experiência prática, o que é essencial para entender os problemas e especialmente suas soluções (CHANG, 2014) (VINCUR *et al.*, 2017) (PELLAS & VOSINAKIS, 2017) * (CAULFIELD *et al.*, 2011) ** (BAU, 2015a) ***.

É de suma importância que essas experiências sejam baseadas em cenários reais, para que o aluno consiga ter um bom aproveitamento do conteúdo a ser lecionado. Muitas áreas da ES são difíceis de praticar devido a sua longa duração e ao potencial de alto dano. Entretanto, os jogos são um instrumento alternativo para ganhar experiência de forma controlada e sem risco (RAMÍREZ-ROSALES *et al.*, 2016) (LINO *et al.*, 2015) (NASSAL, 2014) ***. Jogar não precisa de muito tempo e cuidado como em um projeto real e, em caso de falha, o jogo pode ser reiniciado. Os jogos possuem uma vantagem muito relevante para o ensino que é a sua repetibilidade, eles podem ser reiniciados a qualquer momento, fazendo ainda com que os alunos consigam aprender através de suas falhas (RAMÍREZ-ROSALES *et al.*, 2016) (CALDERÓN *et al.*, 2018) (SZABO, 2014) (ATAL & SUREKA, 2015) ***.

Os jogos são muitas vezes competitivos, o que significa que existem um vencedor e um perdedor. Muitos estudiosos acreditam no poder do engajamento criado pela competição, fazendo os alunos estudarem ainda mais para poderem superar seus adversários (MORALES-TRUJILLO *et al.*, 2018) (SU, 2016) (MILOSZ & MILOSZ, 2017) ***. Essa competição ainda pode trazer alguns estímulos, como: auto realização, desafio, vitória, recompensa, prazer e diversão (TELUKUNTA *et al.*, 2014) ***. Apesar de todas as vantagens já descritas, um dos pontos fundamentais da utilização de jogos na educação é o divertimento, fazendo com que os estudos virem uma diversão e esta vire um estudo (CHAVES *et al.*, 2015) ***.

Um jogo é uma representação visível e física de um espaço problemático; um modelo mental capturado que pode ser repetido **. Como tal, eles são: lugares para testar novas ideias e experimentar teorias estabelecidas; repeti-las quantas vezes for necessário; lugares onde o tempo e o espaço podem ser contraídos ou expandidos; lugares onde é aceitável apenas tentar coisas diferentes e onde é possível aprender mais pelo fracasso do que pelo sucesso (CAULFIELD *et al.*, 2011) ** (RAMÍREZ-ROSALES *et al.*, 2016) (NASSAL, 2014) ***.

A simulação aumenta a oportunidade de aprender pelo fracasso sem perda alguma na realidade, mas de uma maneira muito oportuna e, além disso, ainda consegue trazer motivação e engajamento por parte dos alunos (JIANG *et al.*, 2015) **. Uma das

principais vantagens que a simulação pode trazer é o *feedback*, que dessa maneira o aluno sabe se está fazendo certo ou errado uma determinada tarefa, influenciando seriamente nos efeitos de aprendizagem. Além disso, os jogos de simulação promovem inúmeros benefícios cognitivos aos alunos, como: motivação, efeitos visuais, experimentação, auto eficácia, auto monitoramento, resolução de problemas e habilidades de pensamento crítico (CALDERÓN *et al.*, 2017c) (LEE, 2016) ***.

Por fim, outros dois conceitos que também apresentam vantagens no ensino são o *e-learning* e a gamificação. O *e-learning* possui vantagens como: a rápida atualização das informações, conteúdos permanentemente disponíveis, acesso a conteúdo personalizado, redução de custos e a definição do ritmo de aprendizado do aluno (USKOV & SEKAR, 2014) ***. Já a gamificação pode ter um impacto positivo no aumento da motivação e engajamento do usuário em relação a um determinado comportamento (ALHAMMAD & MORENO, 2018) **.

Várias vantagens foram atribuídas a utilização dos jogos como estratégia de ensino, a Tabela 3.29 a seguir demonstra cada uma das vantagens que foram encontradas nos estudos citados.

Tabela 3.29: Vantagens da utilização dos jogos como técnica de ensino.

<p>Entretenimento</p>	<p>(DUARTE & PEARCE, 2017) (JORDINE <i>et al.</i>, 2014) (SERRANO-LAGUNA <i>et al.</i>, 2015) (TOPALLI & CAGILTAY, 2018) (LAW, 2018) (BOLDBAATAR & ŞENDURUR, 2018) (CHANG, 2014) (VINCUR <i>et al.</i>, 2017) (LEE <i>et al.</i>, 2014) (MARIMUTHU & GOVENDER, 2018) *</p> <p>(ALHAMMAD & MORENO, 2018) (SOUZA <i>et al.</i>, 2017a) (ATAL & SUREKA, 2015) (JIANG <i>et al.</i>, 2015) (CAULFIELD <i>et al.</i>, 2011) **</p>
-----------------------	---

	<p>(QU <i>et al.</i>, 2014) (CALDERÓN <i>et al.</i>, 2018) (MORALES-TRUJILLO <i>et al.</i>, 2018) (LARENAS <i>et al.</i>, 2018) (SUESCÚN, 2018) (VASCONCELOS <i>et al.</i>, 2018) (SOUZA <i>et al.</i>, 2017b) (CALDERÓN <i>et al.</i>, 2017a) (PIEPER <i>et al.</i>, 2017) (HERRANZ <i>et al.</i>, 2017) (MAXIM <i>et al.</i>, 2016) (DUARTE & PEARCE, 2017) (BAU, 2015a) (RAMÍREZ-ROSALES <i>et al.</i>, 2016) (JIMÉNEZ-HERNÁNDEZ <i>et al.</i>, 2016) (UNKELOS-SHPIGEL, 2016) (LEE, 2016) (ALBILALI & QURESHI, 2016) (CHAVES <i>et al.</i>, 2015) (USKOV & SEKAR, 2014) (DLAMINI, 2015) (SZABO, 2014) (ATAL & SUREKA, 2015) (MILOSZ & MILOSZ, 2017) (CHAVES <i>et al.</i>, 2015) ***</p>
<p>Conhecimento prático</p>	<p>(JORDINE <i>et al.</i>, 2014) (SERRANO-LAGUNA <i>et al.</i>, 2015) (LEE <i>et al.</i>, 2014) (PELLAS & VOSINAKIS, 2017) (POTTER <i>et al.</i>, 2014) (MARIMUTHU & GOVENDER, 2018) (TABET <i>et al.</i>, 2016) (PELLAS, 2014) (SALEN <i>et al.</i>, 2004) (KALELIOGLU & GÜLBAHAR, 2014) (SÁEZ-LÓPEZ <i>et al.</i>, 2016) (VACA-CÁRDENAS, 2015) (YAMASHITA <i>et al.</i>, 2017) *</p> <p>(JIANG <i>et al.</i>, 2015) (CAULFIELD <i>et al.</i>, 2011) **</p> <p>(QU <i>et al.</i>, 2014) (SOUZA <i>et al.</i>, 2017b) (PIEPER <i>et al.</i>, 2017) (JIMÉNEZ-HERNÁNDEZ <i>et al.</i>, 2016) (MAXIM <i>et al.</i>, 2016) (RAMÍREZ-ROSALES <i>et al.</i>, 2016) (USKOV & SEKAR, 2014) (LEE, 2016) (DLAMINI, 2015) (MORALES-TRUJILLO <i>et al.</i>, 2018) (ATAL & SUREKA, 2015) (VASCONCELOS</p>

		<i>et al.</i> , 2018) ***
Engajamento e motivação		(DUARTE & PEARCE, 2017) (LEE <i>et al.</i> , 2014) (VINCUR <i>et al.</i> , 2017) (LAW, 2018) (BOLDBAATAR & ŞENDURUR, 2018) (SIERRA <i>et al.</i> , 2016) (LOPEZ <i>et al.</i> , 2016) * (ALHAMMAD & MORENO, 2018) (SOUZA <i>et al.</i> , 2017a) (KOSA <i>et al.</i> , 2016) (JIANG <i>et al.</i> , 2015) (CAULFIELD <i>et al.</i> , 2011) ** (QU <i>et al.</i> , 2014) (SUESCÚN <i>et al.</i> , 2018) (VASCONCELOS <i>et al.</i> , 2018) (SOUZA <i>et al.</i> , 2017b) (CALDERÓN <i>et al.</i> , 2017c) (HERRANZ <i>et al.</i> , 2017) (MAXIM <i>et al.</i> , 2016) (VALENCIA <i>et al.</i> , 2016) (RAMÍREZ-ROSALES <i>et al.</i> , 2016) (JIMÉNEZ-HERNÁNDEZ <i>et al.</i> , 2016) (UNKELOS-SHPIGEL, 2016) (LEE, 2016) (ALBILALI & QURESHI, 2016) (USKOV & SEKAR, 2014) (DLAMINI, 2015) (ATAL & SUREKA, 2015) (SHABALINA <i>et al.</i> , 2013) ***
Retenção de conhecimento		(JORDINE <i>et al.</i> , 2014) (LEE <i>et al.</i> , 2014) (VINCUR <i>et al.</i> , 2017) (BOLDBAATAR & ŞENDURUR, 2018) (SIERRA <i>et al.</i> , 2016) (LOPEZ <i>et al.</i> , 2016) * (ALHAMMAD & MORENO, 2018) (SOUZA <i>et al.</i> , 2017a) (KOSA <i>et al.</i> , 2016) **
Manter os alunos focados e interessados		(DUARTE & PEARCE, 2017) (JORDINE <i>et al.</i> , 2014) (VINCUR <i>et al.</i> , 2017) (LEE <i>et al.</i> , 2014) (LAW, 2018) (BOLDBAATAR & ŞENDURUR, 2018) *

	<p>(ALHAMMAD & MORENO, 2018) (KOSA <i>et al.</i>, 2016) (JIANG <i>et al.</i>, 2015) (CAULFIELD <i>et al.</i>, 2011) **</p> <p>(VASCONCELOS <i>et al.</i>, 2018) (SOUZA <i>et al.</i>, 2017b) (HERRANZ <i>et al.</i>, 2017) (RAMÍREZ-ROSALES <i>et al.</i>, 2016) (ALBILALI & QURESHI, 2016) (USKOV & SEKAR, 2014) (TELUKUNTA <i>et al.</i>, 2014) (SHABALINA <i>et al.</i>, 2013) ***</p>
Interatividade	<p>(VINCUR <i>et al.</i>, 2017) *</p> <p>(JIANG <i>et al.</i>, 2015) **</p> <p>(MORSCHHEUSER <i>et al.</i>, 2018) (SUESCÚN <i>et al.</i>, 2018) (VALENCIA <i>et al.</i>, 2016) (SU, 2016) (CHAVES <i>et al.</i>, 2015) (UYAGUARI <i>et al.</i>, 2015) (DLAMINI, 2015) ***</p>
Colaboração e competição	<p>(MORALES-TRUJILLO <i>et al.</i>, 2018) (PIEPER <i>et al.</i>, 2017) (SU, 2016) (UNKELOS-SHPIGEL, 2016) (USKOV & SEKAR, 2014) (TELUKUNTA <i>et al.</i>, 2014) (MILOSZ & MILOSZ, 2017) ***</p>
<i>Feedback</i> imediato	<p>(DUARTE & PEARCE, 2017) (LAW, 2018) (BOLDBAATAR & ŞENDURUR, 2018) (CHANG, 2014) *</p> <p>(CALDERÓN <i>et al.</i>, 2017c) (PIEPER <i>et al.</i>, 2017) (RAMÍREZ-ROSALES <i>et al.</i>, 2016) ***</p>
Reflexão	<p>(VASCONCELOS <i>et al.</i>, 2018) (PIEPER <i>et al.</i>, 2017) (HERRANZ <i>et al.</i>, 2017) (LEE, 2016) (VINCUR <i>et</i></p>

		<i>al.</i> , 2017) *
Imersão		(VINCUR <i>et al.</i> , 2017) * (CALDERÓN <i>et al.</i> , 2017c) (LOPES & PETRILLO, 2016) (MAXIM <i>et al.</i> , 2016) (DLAMINI, 2015) ***
Cenário reais		(VINCUR <i>et al.</i> , 2017) (CHANG, 2014) * (CALDERÓN <i>et al.</i> , 2017b) (LOPES & PETRILLO, 2016) (MAXIM <i>et al.</i> , 2016) (NASSAL, 2014) ***
Confiança		(ALHAMMAD & MORENO, 2018) (KOSA <i>et al.</i> , 2016) (CAULFIELD <i>et al.</i> , 2011) ** (GARCIA <i>et al.</i> , 2019) (CALDERÓN <i>et al.</i> , 2017c) (HERRANZ <i>et al.</i> , 2017) (CALDERÓN & RUIZ, 2016) (UNKELOS-SHPIGEL, 2016) (POTTER <i>et al.</i> , 2014) ***
Retenção de conhecimento	de	(VINCUR <i>et al.</i> , 2017) * (ALHAMMAD & MORENO, 2018) (KOSA <i>et al.</i> , 2016) (CAULFIELD <i>et al.</i> , 2011) ** (LARENAS <i>et al.</i> , 2018) (GARCIA <i>et al.</i> , 2019) (HERRANZ <i>et al.</i> , 2017) (CALDERÓN <i>et al.</i> , 2017b) (HERRANZ <i>et al.</i> , 2017) (CALDERÓN & RUIZ, 2016) (UNKELOS-SHPIGEL, 2016) (USKOV & SEKAR, 2014) (POTTER <i>et al.</i> , 2014) ***
Experiência sem risco		(VINCUR <i>et al.</i> , 2017) * (JIANG <i>et al.</i> , 2015) (CAULFIELD <i>et al.</i> , 2011) **

	(LINO <i>et al.</i> , 2015) (NASSAL, 2014) ***
Repetibilidade	(CAULFIELD <i>et al.</i> , 2011) **
Aprender através de falhas	(JIANG <i>et al.</i> , 2015) (CAULFIELD <i>et al.</i> , 2011) ** (CALDERÓN <i>et al.</i> , 2018) (RAMÍREZ-ROSALES <i>et al.</i> , 2016) (ATAL & SUREKA, 2015) (SZABO, 2014) ***

3.5.4 Qual(s) a(s) desvantagem(s) do uso de jogos para o ensino de engenharia de software?

Apesar do grande número de trabalhos analisados, poucos artigos relatavam problemas que envolviam a utilização dos jogos na educação.

Jogos educativos são difíceis de projetar por uma variedade de razões, como: são complexos e multifacetados; visam afetar um comportamento e não apenas entreter, o que implica na compreensão de uma série de atividades psicológicas (motivacional) e requer competências apropriadas da equipe de desenvolvimento (MORSCHHEUSER *et al.*, 2018) ***. Outro problema que envolve a criação de jogos educativos é que se este for muito complexo, os alunos reduzem o tempo gasto na solução de problemas para se concentrar no jogo que está sendo usado. Essa é uma distração indesejada e qualquer jogo usado deve ser de fácil utilização, permitindo que o aluno se concentre em resolver o problema e não em como usar o jogo (BOLDBAATAR & ŞENDURUR, 2018) (CHAVES *et al.*, 2015) (PELLAS, 2014) * (ALHAMMAD & MORENO, 2018) ** (MORSCHHEUSER *et al.*, 2018) (VALENCIA *et al.*, 2016) ***.

O uso limitado de jogos sérios na educação formal pode estar relacionado às questões que cercam o uso de jogos para lazer, como por exemplo, uma visão onde os jogos podem viciar, não são produtivos e podem ensinar conceitos errados. Outro ponto muito relevante em relação a não utilização de jogos como meio de ensino é que os jogadores normalmente aprendem através da repetição, padrões e exploração, o que

contrasta com a aprendizagem de quantidades discretas de informação, como ocorre nas escolas e cursos de graduação de maneira geral (VALENCIA *et al.*, 2016) ***.

Para criar um jogo educativo eficiente não basta usar qualquer tipo de jogo que execute qualquer atividade. Para efetivamente ensinar e reter melhor as práticas, é preciso que os estudantes sejam mais ativos e é preciso usar melhores métodos instrucionais nos quais eles participam de tarefas cognitivas de níveis mais altos (LINO *et al.*, 2015) ***.

Os jogos como ferramentas pedagógicas são bem recebidos pelos jogadores, no entanto, eles não são autossuficientes e devem ser complementados por outros dispositivos de aprendizagem. Ou seja, o jogo sério não consegue sozinho ensinar um assunto, ele deve ser apoiado pela teoria, ele funciona mais como um fortalecimento do conhecimento (PELLAS, 2014) * (BOLDBAATAR & ŞENDURUR, 2018) (PELLAS, 2014) * (MAURICIO *et al.*, 2017) (SOUZA *et al.*, 2017a) (CAULFIELD *et al.*, 2011) ** (SOUZA *et al.*, 2017b) ***. Além do mais eles são apenas uma representação de como o mundo funciona, portanto, é potencialmente perigoso ter jogadores deixando o ambiente de jogo com a crença de que as estratégias que foram efetivamente empregadas no jogo são diretamente transferíveis para o mundo real (LINO *et al.*, 2015) (CAULFIELD *et al.*, 2011) **.

A gamificação também implica em algumas desvantagens. A possível interpretação errônea das informações e a implementação inadequada dos elementos podem gerar valores que não são pretendidos, introduzindo, por exemplo, uma competitividade injusta, um fator que é altamente desmotivador (HERRANZ *et al.*, 2017) ***. A gamificação faz utilização de vários elementos de jogos, entretanto, alguns desses elementos podem trazer problemas se não usados corretamente, como, por exemplo, as tabelas de classificação (SOUZA *et al.*, 2017b) ***, que podem fazer com que os alunos se sintam desmotivados se estiverem com uma pontuação muito baixa.

A gamificação é uma técnica nova e imatura e, por isso, deve-se ter muito cuidado ao tentar gamificar um processo. Muitos aplicativos já deixaram de atender os objetivos do negócio ao serem gamificados de maneira incorreta. Estudos demonstram que quando ela não foi gradualmente incorporada, obteve-se resultados negativos

(PELLAS, 2014) * (ALHAMMAD & MORENO, 2018) ** (SOUZA *et al.*, 2017b) (HERRANZ *et al.*, 2017) (MORSCHHEUSER *et al.*, 2018) ***. A rápida inserção de elementos gamificados podem causar uma sensação de confusão ou frustração que impediu os alunos de compreender e desfrutar do "jogo". Os alunos podem se distrair com novas regras, novos elementos, novas maneiras de aprender, deslocando a experiência de aprendizado como um todo. A gamificação deve ser introduzida gradualmente começando com nenhuma nova atividade e apenas integrando elementos de gamificação em processos existentes (ALHAMMAD & MORENO, 2018) **.

Por fim foi identificado mais uma curiosidade, onde o trabalho de Boldbaatar e Sendurur (BOLDBAATAR & ŞENDURUR, 2018) * demonstrou que, apesar das vantagens oferecidas pela programação visual, a programação baseada em texto obteve melhores resultados em relação a transferência de conhecimento.

3.5.5 Qual(s) a(s) principal(s) característica(s) do jogo(s) utilizado(s)?

A partir desse estudo, foram identificados vários jogos e características que buscavam aumentar a motivação e o engajamento por meio da diversão. Em grande parte dos estudos, compreende-se que esses jogos com essas características são utilizados por usuários com conhecimento mínimo ou inexistente na disciplina a ser lecionada.

Vários jogos foram encontrados com diversas características juntas, entre os principais jogos identificados se destacam: o jogo LightBot, Cargo-Bot e o Robo3, que são jogos para ensinar lógica de programação e possuem características como: nível, progressão de dificuldade, *feedback*, desafios, uso de tarefas similares e demonstram conceitos como: funções, abstração, controle de fluxos, recursão e reutilização (DUARTE & PEARCE, 2017) (LAW, 2018) (LOPEZ *et al.*, 2016) (LEE *et al.*, 2014) (LAW, 2018) (AGALBATO, 2017) *. Outro jogo identificado foi o Lost in Space que inclui, entre outras características, um sistema de regras de jogo, um mecanismo de física e um mecanismo de renderização (SERRANO-LAGUNA *et al.*, 2015) *.

Nessa pesquisa, também, foram identificadas algumas linguagens de programação visuais que fazem uso “blocos” para construção de programas. Elas não são tidas como jogos diretamente, mas utilizam algumas características semelhantes, como por exemplo: o apelo visual, sequencias de etapas de devem ser seguidas em uma fase e interação com o usuário. Esses tipos de linguagens podem ter diversas vantagens, tais como: são independentes de linguagem, o que pode diminuir os erros de sintaxe, principalmente para programadores iniciantes; *feedback* imediato, permite auto regulação das aprendizagens; podem ser utilizadas fora da sala de aula; são intuitivas, simples, fácil de utilizar (VACA-CÁRDENAS, 2015) (CHANDRASHEKAR *et al.*, 2018) *. Vale lembrar que essas linguagens não são jogos, mas algumas delas já possuem jogos criados com bases nelas, como é o caso do Google Blockly e do Scratch.

As duas primeiras linguagens a serem identificadas foram a Alice (LEE *et al.*, 2014) (CHANG, 2014) * e o Scratch (CHANG, 2014) (TABET *et al.*, 2016) (TOPALLI & CAGILTAY, 2018) (CHANDRASHEKAR *et al.*, 2018) (PELLAS & VOSINAKIS, 2017) (MARIMUTHU & GOVENDER, 2018) (PELLAS, 2014) (KALELIOGLU & GÜLBAHAR, 2014) (SÁEZ-LÓPEZ *et al.*, 2016) (VACA-CÁRDENAS *et al.*, 2015) *, que são linguagens de programação visual baseadas em blocos projetadas para promover a manipulação de mídia para novos programadores. A partir dessas linguagens é possível fazer *upload* de projetos de mídia e *scripts*, histórias animadas, jogos, relatórios de livros, cartões de felicitações, vídeos de música, tutoriais, simulações e projetos de arte e música. Outras duas linguagens muito parecidas com as que foram descritas são a StarLogo TNG (BOLDBAATAR & ŞENDURUR, 2018) * e o Droplet (BAU, 2015a) *, que também são linguagem visuais no estilo *drag-and-drop*.

O Greenfoot foi outra ferramenta encontrada, tendo como objetivo ensinar programação orientada a objetos. Além disso, a ferramenta permite que os professores introduzam os conceitos mais importantes e fundamentais da orientação a objetos de uma maneira facilmente compreensível (CHANDRASHEKAR *et al.*, 2018) *. Por fim, a última linguagem visual encontrada se chama Google Blockly (YAMASHITA *et al.*, 2017) *, que é uma biblioteca para construção de editores de programação visual.

A Tabela 3.30 faz uma breve sumarização das características que foram encontradas nos jogos que foram utilizados como técnicas de ensino.

Tabela 3.30: Características utilizadas nos jogos como técnicas de ensino.

Tabelas de classificação	(ALHAMMAD & MORENO, 2018) (MAURICIO <i>et al.</i> , 2017) ** (MAXIM <i>et al.</i> , 2016) (UNKELOS-SHPIGEL, 2016) (DLAMINI, 2015) ***
Pontuação	(DUARTE & PEARCE, 2017) (LEE <i>et al.</i> , 2014) (SERRANO-LAGUNA <i>et al.</i> , 2015) (AGALBATO, 2017) (LAW, 2018) (SIERRA <i>et al.</i> , 2016) (LOPEZ <i>et al.</i> , 2016) * (MAURICIO <i>et al.</i> , 2017) (JIANG <i>et al.</i> , 2015) ** (MATSUBARA & DA SILVA, 2017) (MAXIM <i>et al.</i> , 2016) (UNKELOS-SHPIGEL, 2016) (DLAMINI, 2015) ***
Níveis	(DUARTE & PEARCE, 2017) (LEE <i>et al.</i> , 2014) (SERRANO-LAGUNA <i>et al.</i> , 2015) (AGALBATO, 2017) (LAW, 2018) (SIERRA <i>et al.</i> , 2016) (LOPEZ <i>et al.</i> , 2016) * (MAURICIO <i>et al.</i> , 2017) (JIANG <i>et al.</i> , 2015) ** (CALDERÓN <i>et al.</i> , 2018) (MATSUBARA & DA SILVA, 2017) (RAMÍREZ-ROSALES <i>et al.</i> , 2016) (JIMÉNEZ-HERNÁNDEZ <i>et al.</i> ,

	2016) (CALDERÓN & RUIZ, 2016) (UNKELOS-SHPIGEL, 2016) (ATAL & SUREKA, 2015) ***
<i>Checkpoints</i>	(DUARTE & PEARCE, 2017) (LEE <i>et al.</i> , 2014) (AGALBATO, 2017) (LAW, 2018) (SIERRA <i>et al.</i> , 2016) (LOPEZ <i>et al.</i> , 2016) * (MAURICIO <i>et al.</i> , 2017) ** (JIMÉNEZ-HERNÁNDEZ <i>et al.</i> , 2016) (ATAL & SUREKA, 2015) *****
<i>Status</i>	(HUNICKE <i>et al.</i> , 2004) (SOUZA <i>et al.</i> , 2017b) (UNKELOS-SHPIGEL, 2016) (WOHLIN, 2014) (MAXIM <i>et al.</i> , 2016) ***
Barra de progresso	(ALHAMMAD & MORENO, 2018) (MAURICIO <i>et al.</i> , 2017) ** (HUNICKE <i>et al.</i> , 2004) (SOUZA <i>et al.</i> , 2017b) (UNKELOS-SHPIGEL, 2016) (WOHLIN, 2014) (MATSUBARA & DA SILVA, 2017) (CALDERÓN & RUIZ, 2016) (ATAL & SUREKA, 2015) ***
Competição e colaboração (<i>multiplayer</i>)	(JORDINE <i>et al.</i> , 2014) * (ALHAMMAD & MORENO, 2018) (MAURICIO <i>et al.</i> , 2017) (JIANG <i>et al.</i> , 2015) ** (MORSCHHEUSER <i>et al.</i> , 2018) (MATSUBARA & DA SILVA, 2017)

	(PIEPER <i>et al.</i> , 2017) (HERRANZ <i>et al.</i> , 2017) (FUCHS & WOLFF, 2016) (UYAGUARI <i>et al.</i> , 2015) ***
Janela de diálogos	(ALHAMMAD & MORENO, 2018) ** (LARENAS <i>et al.</i> , 2018) (RAMÍREZ-ROSALES <i>et al.</i> , 2016) ***
Recompensas	(MAURICIO <i>et al.</i> , 2017) ** (MATSUBARA & DA SILVA, 2017) (ATAL & SUREKA, 2015) ***
<i>Feedback</i> constante	(DUARTE & PEARCE, 2017) (LEE <i>et al.</i> , 2014) (SERRANO-LAGUNA <i>et al.</i> , 2015) (AGALBATO, 2017) (LAW, 2018) (LOPEZ <i>et al.</i> , 2016) * (MAURICIO <i>et al.</i> , 2017) (JIANG <i>et al.</i> , 2015) ** (MORSCHHEUSER <i>et al.</i> , 2018) (LARENAS <i>et al.</i> , 2018) (MATSUBARA & DA SILVA, 2017) (HERRANZ <i>et al.</i> , 2017) (RAMÍREZ-ROSALES <i>et al.</i> , 2016) (FUCHS & WOLFF, 2016) (UYAGUARI <i>et al.</i> , 2015) (ATAL & SUREKA, 2015) ***
<i>Boss</i> (chefe da fase)	(ALHAMMAD & MORENO, 2018) (JIANG <i>et al.</i> , 2015) ** (MATSUBARA & DA SILVA, 2017) ***
Simulação	(VINCUR <i>et al.</i> , 2017) *

	<p>(JIANG <i>et al.</i>, 2015) **</p> <p>(CALDERÓN <i>et al.</i>, 2018) (PIEPER <i>et al.</i>, 2017) (DLAMINI, 2015) ***</p>
Cenário real	<p>(VINCUR <i>et al.</i>, 2017) *</p> <p>(JIANG <i>et al.</i>, 2015) **</p> <p>(CALDERÓN <i>et al.</i>, 2018) (PIEPER <i>et al.</i>, 2017) (SU, 2016) (DLAMINI, 2015) ***</p>
Imersão	<p>(VINCUR <i>et al.</i>, 2017) *</p> <p>(DALPIAZ & COOPER, 2018)</p> <p>(LASKOWSKI, 2015) (QU <i>et al.</i>, 2014)</p> <p>(CALDERÓN <i>et al.</i>, 2018) ***</p>
Resultado final	<p>(DUARTE & PEARCE, 2017) (LEE <i>et al.</i>, 2014) (SERRANO-LAGUNA <i>et al.</i>, 2015)</p> <p>(VINCUR <i>et al.</i>, 2017) (LAW, 2018)</p> <p>(SIERRA <i>et al.</i>, 2016) *</p> <p>(JIANG <i>et al.</i>, 2015) **</p>
Tempo	<p>(MAURICIO <i>et al.</i>, 2017) (JIANG <i>et al.</i>, 2015) **</p> <p>(CALDERÓN <i>et al.</i>, 2018) (RAMÍREZ-ROSALES <i>et al.</i>, 2016) ***</p>
Narrativa	<p>(ALHAMMAD & MORENO, 2018) **</p> <p>(NASSAL, 2014) ***</p>

Desafio	(DUARTE & PEARCE, 2017) (JORDINE <i>et al.</i> , 2014) (LEE <i>et al.</i> , 2014) (SERRANO-LAGUNA <i>et al.</i> , 2015) (VINCUR <i>et al.</i> , 2017) (AGALBATO, 2017) (LAW, 2018) (SIERRA <i>et al.</i> , 2016) * (ALHAMMAD & MORENO, 2018) ** (JIMÉNEZ-HERNÁNDEZ <i>et al.</i> , 2016) ***
Dependência entre os conteúdos	(CALDERÓN <i>et al.</i> , 2018) (PIEPER <i>et al.</i> , 2017) (RAMÍREZ-ROSALES <i>et al.</i> , 2016) (DLAMINI, 2015) ***
Gráficos estimulantes	(DUARTE & PEARCE, 2017) (JORDINE <i>et al.</i> , 2014) (LEE <i>et al.</i> , 2014) (SERRANO-LAGUNA <i>et al.</i> , 2015) (VINCUR <i>et al.</i> , 2017) (LOPEZ <i>et al.</i> , 2016) * (MAURICIO <i>et al.</i> , 2017) (JIANG <i>et al.</i> , 2015) ** (LARENAS <i>et al.</i> , 2018) (PIEPER <i>et al.</i> , 2017) (CALDERÓN & RUIZ, 2016) (DLAMINI, 2015) ***
Existência de meta-jogos	(CALDERÓN & RUIZ, 2016) ***
Avatar	(JESUS <i>et al.</i> , 2018) (MAXIM <i>et al.</i> , 2016) (DLAMINI, 2015) ***

Nesse estudo, foram observadas basicamente seis maneiras de se integrar os jogos como estratégia de ensino, dependendo de suas características. A Figura 3.8 faz a sumarização desse conteúdo, com a intenção de facilitar o leitor a conseguir escolher a

técnica que mais se adequa ao seu propósito de ensino, com base nas características que serão utilizadas.


































	Serious Games	Game-based learning	Digital game-based learning	Game development based learning	Simulation	Gamification
Definição	Jogo criado com outro propósito que não apenas entretenimento	Qualquer abordagem usando jogos para fins de aprendizado	Qualquer abordagem usando jogos digitais para fins de aprendizado	Prática de desenvolver jogos como experiência de aprendizado	Imitação de um processo do mundo real ao longo de tempo	Uso de elementos de jogos em outros contextos que não sejam apenas lúdicos
Propósito	Aprendizagem, saúde, distribuição de informações, etc	Aprendizagem			Aprendizagem, saúde, treinamento, etc	Usado para aumentar a motivação de um conteúdo
Divertimento						
Game play						
Motivação, engajamento e feedback						
Mundo Virtual						
Aprendizagem pelo fracasso						
Uso de recompensas						
Recursos Digitais						
Desvantagens	Não são autossuficientes, são vistos como um método de ensino secundário que precisam de uma fundamentação teórica antes do uso				Apenas uma representação. É perigoso ter jogadores com a crença de que as estratégias que foram aplicadas são diretamente transferíveis para o mundo real	Algo novo e imaturo, deve-se ter muito cuidado ao tentar gamificar algo para não perder o conceito real da aplicação

Figura 3.8: Sumarização dos achados. Fonte: do autor.

3.5.6 Qual(is) foi(ram) o(s) método(s) de avaliação utilizado(s)?

Várias maneiras de avaliação com pequenas mudanças foram observadas, entretanto, todas giravam em torno de um estudo de caso, onde um grupo de pessoas utilizava o jogo e após esse teste era inserido um questionário de avaliação.

O primeiro e mais utilizado método de avaliação (quase metade dos artigos utilizaram) foi através do uso de um questionário após a utilização do jogo onde os usuários podiam comentar suas interpretações sobre o jogo fazer sugestões para

melhorias do jogo (VINCUR *et al.*, 2017) (BOLDBAATAR & ŞENDURUR, 2018) (SIERRA *et al.*, 2016) (CHANG, 2014) (TABET *et al.*, 2016) (TOPALLI & CAGILTAY, 2018) (CHANDRASHEKAR *et al.*, 2018) (YAMASHITA *et al.*, 2017) * (MORALES-TRUJILLO *et al.*, 2018) (LARENAS *et al.*, 2018) (SUESCÚN *et al.*, 2018) (GARCIA *et al.*, 2019) (AYDAN *et al.*, 2017) (SOUZA *et al.*, 2017b) (CALDERÓN *et al.*, 2017b) (SU, 2016) (RAMÍREZ-ROSALES *et al.*, 2016) (UNKELOS-SHPIGEL, 2016) (LEE, 2016) (ALBILALI & QURESHI, 2016) (LINO *et al.*, 2015) (USKOV & SEKAR, 2014) (DLAMINI, 2015) (ATAL & SUREKA, 2015) (SZABO, 2014) (TELUKUNTA *et al.*, 2014) (SHABALINA *et al.*, 2013) (BERKLING & THOMAS, 2013) (THOMAS & BERKLING, 2013) ***.

Outra maneira de avaliação um pouco parecida foi à utilização de dois grupos de participantes, um primeiro que fez uso da nova estratégia a ser adotada e um segundo que serviu como grupo de controle (LEE *et al.*, 2014) (SERRANO-LAGUNA *et al.*, 2015) * (PIEPER *et al.*, 2017) ***. Com esses dois grupos, foi possível verificar se o jogo estava, realmente, fazendo diferença no meio de ensino. Para verificar qual técnica de ensino era a mais apropriada, um questionário também foi executado para os dois grupos, contendo questões que procuravam verificar o quão eficiente era o jogo (LASKOWSKI, 2015) ***.

Outro método de avaliação foi com a utilização do GQM (Goal Question Metric) para elaborar as métricas para os objetivos, formulando assim as perguntas do questionário. Nesse método, os participantes preencheram um questionário de caracterização sobre a experiência, em seguida, foi dada uma introdução geral sobre o assunto a ser ensinado. Por fim, um role-playing foi realizado, simulando o que deveria ser ensinado (POTTER *et al.*, 2014) ***.

Uma sugestão de organização do questionário a ser respondido é dividi-lo em quatro sessões. A primeira sessão é usada para identificar o participante; a segunda sessão tem como objetivo coletar dados sobre o comportamento e preferências dos participantes relacionados aos jogos; a terceira sessão tem como objetivo conhecer as percepções dos participantes sobre a motivação do mesmo em relação ao trabalho; e a quarta sessão pretende verificar o conhecimento do participante sobre o jogo

(VASCONCELOS *et al.*, 2018) ***. Uma adaptação do modelo sugerido acima é feita com a utilização de 3 questionários: um antes do jogo, testando os conhecimentos do usuário, um questionário enquanto o jogo estava em andamento, para ver a progressão do ensino, e mais um questionário executado no final (CALDERÓN *et al.*, 2017c) ***. Outro meio de organizar o questionário é por meio de 7 fatores: (1) interesse/prazer, (2) competência, (3) esforço, (4) pressão/estresse, (5) escolha, (6) utilidade e (7) conexão. As respostas variam de 1 a 7, onde 1 representa “completamente não” e 7 significa “muito verdadeiro” (JIMÉNEZ-HERNÁNDEZ *et al.*, 2016) ***.

Outra maneira de avaliar os jogos é criando avaliações de usabilidade e de aprendizagem separadas para serem preenchidas após a utilização dos jogos. O preenchimento de um formulário *online* anterior é sugerido para ter um conhecimento prévio da ferramenta. Como parte da avaliação de usabilidade, cada membro da equipe revisa o jogo, preenchendo uma lista de verificação de usabilidade (MAXIM *et al.*, 2016) ***.

Os métodos descritos acima, de uma maneira geral, são muito parecidos com o MEEGA, que visa avaliar a efetividade educacional dos jogos utilizando um questionário pós-jogo. Então, para realizar a avaliação, basta selecionar o jogo, definir as questões adicionais que avaliam o *feedback* de aprendizagem, conduzir uma sessão prática e, após a experiência, coletar os dados através do questionário pós-jogo (CALDERÓN *et al.*, 2018) ***.

3.5.7 Perguntas extras

As perguntas a seguir não apresentam identificadores para demonstrar de onde veio cada uma das referências, pois todas elas foram retiradas do estudo sobre “Ensino de reutilização de software”.

3.5.7.1 Qual(is) foi(ram) a(s) característica(s) utilizada(s) no meio de ensino?

Grande parte dos autores acreditam que a reutilização de software precisa ser ensinada mais precocemente nos cursos de graduação de computação (CASPERSEN &

CHRISTENSEN, 2008) e acreditam que esse ensino pode ser estendido de diversas maneiras possíveis, desde a criação de um novo curso até a utilização de filmes, sons ou até mesmo jogos (CASPERSEN & CHRISTENSEN, 2008) (TEWARI, 1995). Atualmente, muitos professores universitários passam do aprendizado passivo para o aprendizado ativo, para encontrar maneiras melhores de envolver os alunos no processo de aprendizado. Isso significa que, em vez de simplesmente receber informações verbal e visuais, os alunos estão recebendo, participando e fazendo (NIU, 2011).

LISBOA *et al.* (2008) buscaram demonstrar o estado da arte e o estado da prática na reutilização de software, expondo os alunos a um desenvolvimento em grande escala, conforme acontece em equipes dentro de uma organização de software. O curso foi dividido em duas partes, a primeira apresentou uma visão geral sobre reutilização de software, exibindo os conhecimentos necessários para entender os problemas atuais e as tecnologias disponíveis. Nessa primeira parte, algumas questões foram apresentadas, tais como: o que significa reutilização de software; quais aspectos do software são reutilizáveis e; quais são os custos e benefícios da reutilização. O objetivo geral era esclarecer os ceticismos e abordar as dúvidas e mal-entendidos dos alunos (CARDOW & WATSON, 1994). Para a segunda parte do curso, o processo de engenharia de domínio foi discutido em três palestras. Além disso, essa parte foi a execução real do projeto (LISBOA *et al.*, 2008). Foram apresentados conceitos como: análise de domínio, construção de componentes e ferramentas, gerenciamento e construção de bibliotecas e construção de sistemas (CARDOW & WATSON, 1994). Os alunos devem receber uma estrutura ou modelo de um problema do mundo real para iniciar o processo de desenvolvimento do sistema. Processos de ciclo de vida de software que garantam a adesão às melhores práticas profissionais devem ser utilizados, permitindo ainda que os alunos trabalhem em equipes (RAMAKRISHNAN *et al.*, 1994).

Outros exemplos de atividades de aprendizado ativo incluem: um breve exercício escrito, sendo uma boa maneira de revisar materiais e fornecer feedback; uma discussão em aula, que exige que os alunos pensem criticamente sobre o assunto e usem a lógica para avaliar suas posições e a dos outros; um grupo de aprendizagem colaborativa, onde os alunos são designados em grupos de 3-6 pessoas e recebem uma

tarefa para trabalhar juntos; um jogo em aula, representa uma maneira enérgica de aprender, porque não apenas ajuda os alunos a revisar o material do curso, mas também ajuda a aprender sobre um tópico de uma maneira divertida (NIU, 2011).

Outra ideia que foi utilizada para trazer uma realidade maior ao curso foi a nomeação de papéis dentro de projetos fictícios. Os estudantes atuaram como analistas de domínio, engenheiros de software, gerente de projeto, testadores, engenheiros de qualidade de software, arquitetos e consultor para executar o projeto. Devido à equipe reduzida para a construção de uma fábrica de software, os alunos puderam assumir mais de uma função, dependendo da fase do projeto (LISBOA *et al.*, 2008).

Dos quinze artigos que procuravam criar um curso sobre reutilização de software, quase metade deles procuravam criar uma junção com uma matéria de linguagem de programação. Assim, o aluno aprende a reutilizar através de componentes, bibliotecas e heranças, construindo suas próprias aplicações e vendo aos poucos que é mais fácil criar software a partir de componentes já prontos do que criar uma aplicação a partir do zero. A linguagem de programação mais utilizada para criar esses cursos é o C++ (WANG & WANG, 2000), devido ao fato de possuir algumas características que facilitam o ensino como, por exemplo, mecanismos de orientação a objetos. Nessa ideia, conceitos de engenharia de software são apresentados aos alunos ao longo do curso de programação, fazendo o aluno aprender alguns conceitos que ficam muitas das vezes na teoria (TEWARI & FRIEDMAN, 1993).

Com base nessa junção da reutilização com a programação, LATTANZI & HENRY (1996) descreveram quais seriam as etapas que deveriam ser apresentadas aos alunos em um curso com esse enfoque: Fundamentos da engenharia de software; Princípios de análise e *design* orientados a objetos; Fundamentos da reutilização de software; Bibliotecas C++ de componentes reutilizáveis; Divisão da turma em grupos; Elaboração de projetos; Reutilização de códigos para atender aos requisitos de *design*; e Entrega do projeto.

FRAKES (1993) listou a sequência de tópicos que o mesmo julga ser importante e a ordem em que os mesmos devem ser ensinados. A sequência era composta por: Introdução à reutilização; Gestão da reutilização; *Design* para reutilização; Bibliotecas;

Análise de Domínio; Reengenharia; e Qualidade de Ativos. De acordo com o autor, a avaliação desses cursos deve seguir o seguinte padrão, 20% do aprendizado e da nota ficam a cargo de exercícios de fixação, 40% seria cobrado por uma prova escrita na primeira metade do curso, 35% ficaria por conta de um projeto, onde os alunos aprenderiam através da realização, e por fim 5% seria cobrado pela participação em sala de aula. Nesse mesmo curso, foi sugerida uma leitura de um livro e de artigos introdutórios para que o aluno conseguisse aprender em casa, através de uma leitura extra. Entretanto, os alunos deram ao livro uma classificação baixa, em geral, eles acharam os artigos sugeridos muito mais úteis, por se tratarem de leituras mais rápidas e diretas. Já para NIU *et al.* (2011), 48% da avaliação deve ficar em torno de um projeto, onde os alunos devem estar imersos, 30% a cargo de uma prova, 10% deve ser avaliado através de um projeto de pesquisa ou escrita de um artigo e, por fim, 12% fica para as apresentações e participações em sala de aula. A Tabela 3.31 mostra um comparativo entre os dois tipos de avaliação do curso de reutilização encontrado.

Tabela 3.31: Abordagens das avaliações propostas.

Tópicos	Curso 1 (FRANKS, 1993)	Curso 2 (NIU, 2011)
Exercícios de fixação	20%	10%
Prova escrita na primeira metade do curso	40%	30%
Projeto	35%	48%
Participação em sala	5%	12%

Por fim, outro meio de ensino que também foi encontrado no estudo foi a utilização de plataformas online, que foi proposto por LIU *et al.* (2013). Nessa plataforma existiriam exemplificações e visualizações de conceitos e tecnologias para que aluno possa interagir e tirar um maior proveito do ensino. Assim, o aluno poderia aprender mais conceitos ao estar fora da sala de aula apenas com o uso da plataforma.

3.5.7.2 Qual(is) área(s) da reutilização de software foi(ram) ensinada(s)?

De maneira geral, existem duas atividades básicas na reutilização de software: desenvolvimento de componentes *para* reutilização e desenvolvimento de software *com* reutilização (de componentes existentes) (WANG & WANG, 2000). Dos artigos que

foram avaliados, essas duas áreas foram as mais exploradas, a primeira tendo quinze artigos discutindo sobre desenvolvimento baseado em componentes ou através do uso de bibliotecas, já a segunda atividade teve nove artigos sobre a criação de componentes reutilizáveis.

Depois dessas duas grandes atividades, a área da reutilização que teve mais trabalhos relacionados foi a de engenharia de domínio, tendo sete artigos. Em seguida, ficou a área das linhas de produto e criação de repositórios de ativos com quatro artigos, *design* para reutilização com três, criação de ambientes para reutilização e reengenharia com dois artigos. Por fim, tiveram outras áreas com apenas um artigo: métricas, processos de reutilização, qualidade e gerenciamento.

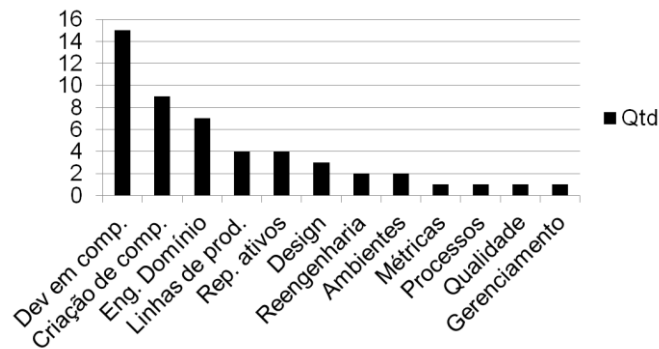


Figura 3.9: Principais áreas de ensino da reutilização de software. Fonte: do autor.

3.6 Considerações Finais

Os estudos exploratórios buscaram identificar algumas informações relevantes sobre a utilização de jogos como meio de ensino de RS e responder as perguntas da Seção 3.1.

Desta forma, foi possível observar quais são as principais dificuldades de ensino de RS sendo as mais constantes, a falta de motivação / engajamento por parte do aluno em relação a disciplina e a falta de oportunidades para colocar em prática o que foi ensinado em sala de aula. Esses estudos também demonstraram que os problemas

listados não se limitam apenas a RS, sendo estes genéricos e afetando diversas disciplinas de uma maneira geral.

Tendo em mente que os problemas listados são genéricos, muito já se foi pesquisado para criar novas abordagens para melhorar o cenário atual de ensino. Dentro dessa pesquisa uma estratégia de que se mostrou relevante foi a utilização de jogos como meio de ensino, levando em consideração que eles poderiam aumentar a motivação do aluno e dar treinamento prático para o mesmo.

Essa estratégia pode se dar de diversas formas, sendo identificadas como as principais o *Game-Based Learning* e a Gamificação. Cada uma dessas técnicas encontradas possui características diferentes e momentos adequados para serem utilizadas com base em suas vantagens e desvantagens. Várias vantagens foram atribuídas a utilização dos jogos como meio de ensino, sendo as principais, o entretenimento que faz com que o aluno consiga se divertir ao estar aprendendo e capacidade de demonstrar conceitos teóricos de uma maneira prática. Apesar das diversas vantagens que foram observadas ao longo da leitura desse capítulo, os jogos como meio de ensino também podem proporcionar algumas desvantagens ao serem aplicados. Cada uma das técnicas possui desvantagens particulares, no entanto, se destacam: a preocupação com distrações ao utilizar o jogo, a interpretação errônea das informações demonstradas e a necessidade do aluno possuir um conhecimento prévio sobre a disciplina ser lecionada para utilizar o jogo.

Fazendo uma comparação entre os principais problemas encontrados em relação ao ensino de RS (**falta de motivação / engajamento do aluno e falta de oportunidades para colocar em prática o que foi ensinado em sala de aula**) e as principais vantagens oferecidas pelos jogos como meio de ensino (**aumentar motivação do aluno e dar treinamento prático**), é possível observar que elas se complementam, tornando essa metodologia uma possível estratégia de ensino para RS.

4 *Game-Based Learning* no ensino de Reutilização de Software

Esse capítulo apresenta duas seções principais, a primeira que apresenta uma abordagem inicial que foi executada para avaliar a utilização de jogos, mais especificamente GBL, como técnica de ensino e outra que descreve o jogo principal desse trabalho.

4.1 Abordagem inicial

Antes da proposta de desenvolvimento do jogo principal desse trabalho era necessário verificar a utilização dos jogos como estratégia de ensino para RS, porque como demonstrado nos estudos exploratórios do Capítulo 3, não foi encontrado nenhum jogo que tenha sido criado com esse propósito específico. Por isso, era necessário que fosse realizada uma abordagem inicial para verificar algumas informações relacionadas a criação de jogos para o ensino de RS, tais como: tipo de jogo mais adequado, experiência proporcionada pelo jogo, satisfação fornecida pelo jogo, entre outras.

4.1.1 Estudo de caso

Nessa abordagem inicial, dois jogos foram criados com base nas características e jogos que foram encontrados nos estudos apresentados no Capítulo 3. Após os desenvolvimentos dos jogos, eles foram avaliados com alunos da graduação do curso de Engenharia de Computação da Universidade Federal do Rio de Janeiro.

Para essa abordagem inicial, procurou-se criar jogos que fossem de simples implementação e que pudessem auxiliar no ensino de RS, para assim avaliar a usabilidade e experiência proporcionada por cada um dos jogos. Como RS contém um material teórico extenso e muitas vezes bastante complexo para entender o seu

propósito, procurou-se criar dois jogos: um para ajudar na memorização das informações da disciplina e outro para ajudar na compreensão da importância da reutilização.

Conforme discutido na Seção 2.2.3, se a finalidade do jogo for memorização, jogos do tipo *quiz* são mais adequados. No entanto, para análise e avaliação de informações, jogos do tipo *puzzle* são mais adequados. Por este motivo, foi escolhido o desenvolvimento de um jogo do tipo *quiz* e outro do tipo *puzzle*. Dentre as *engines*⁹ que existem hoje disponíveis (e.g., Unreal, Game Maker, Construct, Godot, etc), utilizou-se a Unity para a construção de todos os jogos deste trabalho, devido a sua facilidade de apoio ao desenvolvimento.

Com base nas características demonstradas na Tabela 3.30 da Seção 3.5.5 e nas escolhas dos tipos de jogos a serem criados, algumas características foram selecionadas. Essas características podem ser observadas na Tabela 4.1.

Tabela 4.1: Características dos jogos.

Características	ReuseQuiz	CodeBoy
<i>Feedback</i> constante	X	X
Pontuação	X	
Níveis de dificuldade	X	X
Gráficos estimulantes		X
Desafio	X	X
Barra de progresso	X	
Tempo	X	
Dependência entre os conteúdos	X	X
Recompensas		X

O primeiro jogo criado foi um *quiz* com o tema de RS (CASTRO & WERNER, 2019). Ressaltando que o jogo não se limita somente a esse tema, podendo ser utilizado para qualquer outro. Nele foram utilizadas características como: variação do número de

⁹ Engine: Motores para a construção de jogos. Programa de computador e/ou conjunto de bibliotecas, para simplificar e abstrair o desenvolvimento de jogos (LEWIS & JACOBSON, 2002).

perguntas, tempo de resposta, demonstração das respostas certas e erradas, barra de progresso, pontuação e níveis de dificuldade. Ele foi criado para ser totalmente customizável, onde o criador das fases pudesse escolher o conjunto de características de cada fase. A Figura 4.1 mostra algumas telas do jogo Quiz.



Figura 4.1: Jogo Quiz (CASTRO & WERNER, 2019). Fonte: do autor.

Já o segundo jogo, denominado CodeBoy (CASTRO & WERNER, 2019), foi inspirado no jogo Lightbot (GOUWS *et al.*, 2013), que foi encontrado no mapeamento deste trabalho, utilizando mecânicas bem parecidas, sendo integrado com a ideia de árvore de modelagem da linguagem FODA (*Feature-Oriented Domain Analysis*) (KANG *et al.*, 1990) no contexto de RS. Vale lembrar que todas as fases foram criadas pensando em fazer o usuário reutilizar o maior número de movimentos possíveis, para que entenda a necessidade da RS e a importância da reutilização de funções. No CodeBoy, o usuário possui uma árvore de elementos no estilo da linguagem FODA. Do lado esquerdo, é a parte dos elementos obrigatórios, onde o usuário deve programar o boneco para “pegar a estrela” e “passar de fase”. Já o lado direito é a parte opcional, onde o usuário deve “chegar até o baú” e “coletar a moeda bônus” da fase. Na parte de baixo dessa árvore, é mostrada uma função em que o jogador pode “criar um método” que pode ser chamado na árvore principal, ensinando o conceito de reutilização. Nesse

jogo, o usuário pode executar as ações de “quebrar obstáculos”, “andar em quatro direções”, “rotacionar o boneco” para os dois lados, “pular” e “abrir o baú”. A Figura 4.2 mostra o jogo CodeBoy e cada um dos jogos é descrito utilizando os modelos OCR e MDA a partir da Tabela 4.2.

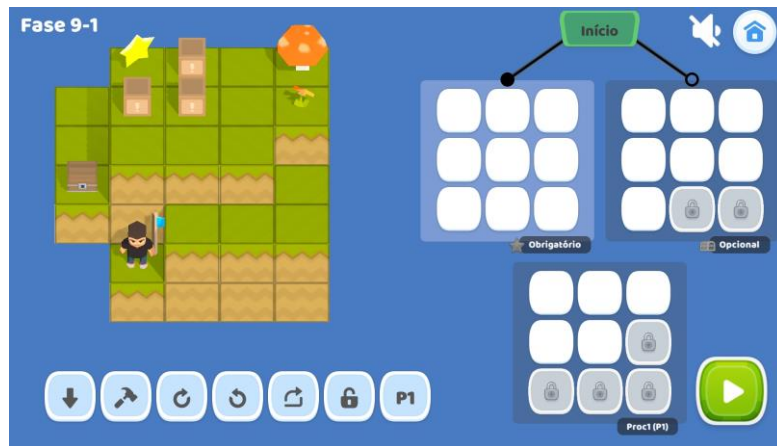


Figura 4.2: Jogo CodeBoy (CASTRO & WERNER, 2019). Fonte: do autor.

Tabela 4.2: Tabela de conceituação dos jogos (CASTRO & WERNER, 2019).

	Quiz	CodeBoy
Objetivo	Ensinar reutilização através de perguntas que o jogador deve responder em cada uma das partidas do jogo. Cada uma dessas perguntas foi formulada retirando informações sobre três livros base sobre reutilização: <i>Confessions of Used Program Salesman</i> , <i>Software Reuse A Standards-Based Guide</i> e <i>Software Reuse Architecture, Process and Organization for Bussiness Success</i> , além de algumas perguntas de concurso que foram retiradas da internet.	Ensinar a importância da reutilização através de uma abordagem onde o jogador em certos pontos do jogo é obrigado a reutilizar pedaços de funções para conseguir passar de fase. Além de ensinar o que são componentes obrigatórios e componentes opcionais através da árvore de modelagem FODA.

Desafio	Conseguir responder a resposta certa dentro do tempo estipulado de um minuto e vinte segundos para cada questão.	Conseguir movimentar o boneco do ponto de partida até a estrela e em seguida até o baú, utilizando a sequência de movimentos corretos.
Recompensa	Conquista estrelas a partir do número de respostas certas dentro de cada fase. A nota de cada fase é realizada a partir do cálculo do número de respostas certas dividido pelo número de perguntas corretas.	Capturar a estrela e o baú com o menor número de movimentos possíveis.
Mecânica	<p>1 - As regras giram em torno do jogador escolher uma resposta dentro das respostas possíveis em uma pergunta.</p> <p>2 - O jogador deve responder uma pergunta dentro do tempo estipulado.</p> <p>3 - Caso o jogador erre, a respostas correta e as erradas são mostradas para o usuário.</p>	<p>1 - Movimentar o personagem</p> <p>2 - Coletar itens (Estrela e baú).</p>
Dinâmica	<p>1 - Pressão (gerado pelo sistema de tempo).</p> <p>2 - Escolha das respostas.</p> <p>3 – Sistema de feedback.</p>	<p>1 - Criar funções com as sequências movimentos.</p> <p>2 – Movimentos diversificados (andar, virar, pular e quebrar).</p>
Estética	<p>1 - Desafio (deve-se conseguir a pontuação para avançar de fase).</p> <p>2 - Submissão (o jogo é visto como um passatempo para o usuário).</p>	<p>1 - Desafio (deve-se conseguir levar o personagem do ponto A ao B).</p> <p>2 - Submissão (o jogo é visto como um passatempo para o usuário).</p> <p>3 - Expressão (o jogador cria sua própria sequência de movimentos).</p>

4.1.2 Estudo de viabilidade

Para avaliar os jogos descritos anteriormente, foi executado um estudo de viabilidade com o objetivo de avaliar a usabilidade e experiência proporcionada por cada um dos jogos em relação ao ensino de RS e, em consequência, capturar novos requisitos para a implementação de futuras melhorias ou de um novo jogo a partir da experiência obtida com esses. Para isso, foram utilizadas duas abordagens de avaliação conjuntas: o questionário MEEGA (PETRI *et al.*, 2016) (removendo as perguntas para jogos *multiplayer*) para ser respondido após a utilização do jogo com o protocolo *think-aloud* (JÄÄSKELÄINEN, 2010) com algumas particularidades, onde a tela do celular foi gravada para análise e, assim, ser possível obter algumas informações extras, tais como: em qual etapa o jogador teve mais dificuldade, quanto tempo o jogador demorou para passar cada uma das fases, entre outras informações. A partir desse protocolo, também, foi possível a captura do som do ambiente, o que auxiliou no entendimento de algumas sensações vivenciadas pelo jogador através de suas reações.

O questionário MEEGA é o modelo de avaliação para jogos educativos que captura informações sobre experiências do jogador e usabilidade (PETRI *et al.*, 2016). Ele possui dois questionários a serem utilizados, o primeiro para preenchimento do aluno que executou a avaliação e outro para utilização do avaliador, de acordo com o que o mesmo observou enquanto o aluno jogava (ANON, 2019). Como a avaliação foi executada pelo mesmo autor do jogo, preferiu-se executar somente o questionário com a perspectiva do aluno, a fim de evitar a captura de informações tendenciosas. Este modelo conta, também, com uma planilha para análise das informações coletadas (ANON, 2019), que foi utilizada com algumas modificações para se adequar ao método utilizado, levando em consideração que algumas perguntas do questionário MEEGA foram excluídas da avaliação por não fazerem sentido para o jogo que estava sendo avaliado.

O estudo piloto foi executado no período do mês de junho de 2019 e contou com a participação de quatro alunos da graduação de Engenharia de Computação da Universidade Federal do Rio de Janeiro em diferentes períodos do curso. Essa amostra foi selecionada para ser dividida em dois grupos, um grupo onde os participantes já

havam cursado a disciplina de RS e outro em que os participantes ainda não haviam cursado. Cada um dos grupos ficou com o total de dois participantes. Vale lembrar que, devido a essa divisão, as perguntas 22, 23, 25 e 26 (ver Apêndice A (Questionário MEEGA)) não puderam ser respondidas por todos os participantes, apenas pelos que realizaram a disciplina, já que as perguntas coletaram informações sobre a aderência do jogo em relação a disciplina lecionada.

Todas as avaliações foram realizadas na versão mobile, devido a facilidade de transporte do jogo para o local onde o mesmo seria testado, e foram executadas no mesmo celular com configuração de 4G de memória, tela de 1920x1080 e processador de 2 GHz 8 Core, para não existir favorecimento de algum jogador quanto ao hardware.

A Figura 4.3 apresenta as informações referentes a cada uma das perguntas que podem ser encontradas no Apêndice A (Questionário MEEGA). Desta forma, é possível observar que em questões de usabilidade os dois jogos foram caracterizados com uma boa estética, possuindo uma interface fácil de ser utilizada. Entretanto, em questões de experiência, foi possível observar que o jogo de *quiz* não manteve o jogador com um alto engajamento, não tendo experiências fortes de desafio, satisfação e diversão.

Além das perguntas coletadas pelo questionário no apêndice, também foram feitas perguntas sobre a existência de aspectos positivos e negativos do jogo, sobre sugestões de melhoria, adição de espaço para comentários adicionais e, por fim, para o participante fazer um comparativo entre os dois jogos utilizados.

A partir dessas perguntas, foi possível observar alguns problemas de *layout* nos dois jogos e capturar algumas sugestões como: o uso de perguntas menores, o acréscimo de tempo para responder, a possibilidade da utilização de uma função que foi construída em uma fase anterior, entre outras. Por fim, foi possível identificar uma alta preferência pelo jogo CodeBoy, que recebeu comentários como: "O jogo está muito bom, eu certamente estaria disposto a baixar na PlayStore", "O jogo é muito atraente", "Com certeza me diverti ao utilizar o jogo", sendo ainda recomendado para ser utilizado para o ensino de abstração e conceitos de reutilização de funções para um público mais jovem. Mas esse também recebeu sugestões e comentários negativos, como: "Novas funções como a habilidade de pular não são introduzidas com clareza para o usuário, poderiam

ser adicionadas no tutorial do início do jogo”, “Seria interessante melhorar a força da sombra para ajudar na percepção de altura”, “Seria legal introduzir o tutorial incrementalmente”.

Em contrapartida, o primeiro jogo foi considerado monótono, recebendo comentários como: “A proposta de gamificação através de um questionário pode ser perigosa, uma vez que se não for explorado com cautela, a proposta de ensinar pode ser deixada de lado para, no lugar, obter sucesso no jogo”, “A proposta de fazer um *quiz* é interessante, mas pressupõe que o indivíduo conhece o conteúdo, caso contrário, é como pegar um livro e pular direto para os exercícios e aprender pelo gabarito”, “Fácil de usar com uma interface simples”, “Seria interessante mostrar o motivo da resposta correta, como se tivesse uma explicação”, “O jogo precisa possuir uma curva de aprendizado mais suave”, “Acho que seria uma ideia interessante o jogo permitir um banco de tempo acumulado, onde o usuário pode gastar esse tempo em perguntas mais longas”, “Em geral o jogo me lembrou bastante o jogo Duolingo”.

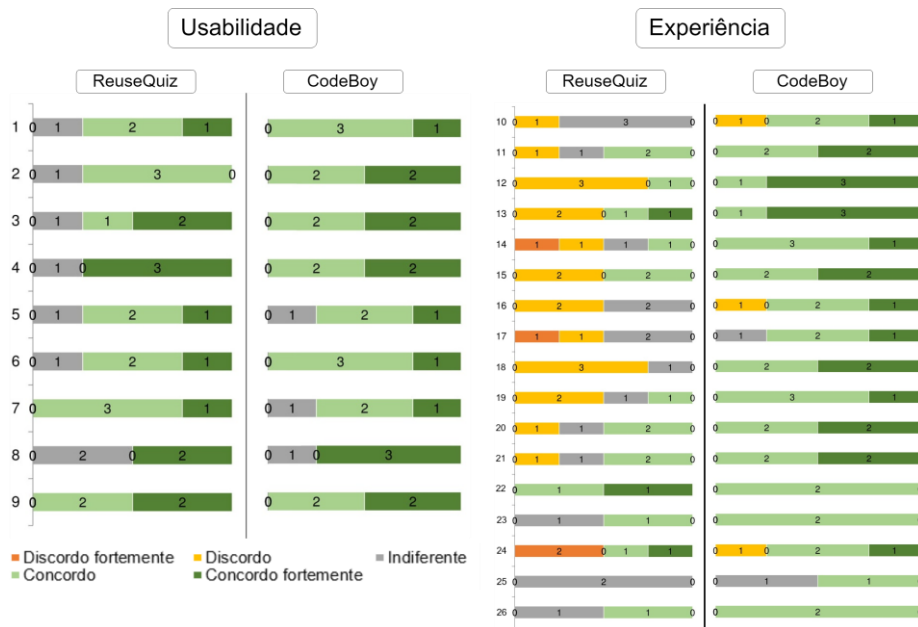


Figura 4.3: Avaliação de usabilidade e experiência dos jogos criados, adaptado de PETRI *et al.* (2019)

Apesar da avaliação inicial contar com uma amostra de apenas quatro alunos, sendo esta relativamente pequena para obter resultados mais confiáveis, foi possível obter as informações necessárias que estavam sendo buscadas e descobrir que a experiência transmitida pela utilização do jogo poderia auxiliar no ensino de reutilização de software, abrindo portas para a proposta de criação de um novo jogo com este propósito educativo.

4.2 Reuse Blocks

A partir da experiência inicial com os jogos descritos anteriormente, construiu-se a solução final desse trabalho, que foi a criação de um jogo para motivar o ensino da reutilização de software por meio da criação de componentes através de uma IDE visual.

4.2.1 Contexto

Conforme mencionado nos estudos exploratórios realizados (Capítulo 3) foram observadas diversas ferramentas com características muito semelhantes a jogos. Essas são denominadas de ferramentas para programação visual, que visam ensinar programação para desenvolvedores iniciantes com a utilização de blocos de código. Essas linguagens, por sua vez, possuem uma forte base de reutilização, onde o usuário cria programas por meio da junção de blocos (i.e., componentes pré-produzidos). Pode-se, portanto, entender os “blocos” de código da programação visual como componentes de software, sendo os dois (componentes e blocos) vistos como trechos de código que exercem uma determinada funcionalidade e devem ser reutilizados para produzir um programa novo. No entanto, apesar das ferramentas de programação visual em blocos introduzirem conceitos de reutilização, nenhuma delas foi criada com o propósito específico de ensinar reutilizar, mas sim de ensinar conceitos introdutórios de programação.

Outra informação que também foi coletada nos estudos exploratórios do Capítulo 3 foi que o tema da reutilização de software que é mais ensinado é o

desenvolvimento baseado em componentes. Conforme discutido anteriormente este tema tem bastante semelhança com o que é aplicado na utilização de linguagens de programação visual, onde o usuário deve juntar “blocos” para a construção de programas. Além disso, os estudos também mostraram que a estratégia mais utilizada para ensinar reutilização, ao menos de código, é por meio de aulas de programação, deixando o aluno criar seu próprio componente.

Na Seção 3.5.1, foi observado que as duas maiores dificuldades de ensino de RS são questões relacionadas ao engajamento/motivação do aluno e dificuldades de colocar em prática o que foi ensinado em sala de aula. Na Seção anterior, dois jogos foram criados para verificar algumas informações relacionadas a criação de jogos como estratégia de ensino para RS. Por meio da avaliação desses jogos, foi possível observar que a questão de engajamento e motivação podem ser tratadas através da utilização dos jogos, restando ainda a questão da prática.

Na Seção 2.2.3, é discutido que para desenvolver jogos que garantam mais atividades práticas para um usuário é necessário criar jogos com características de simulação ou construção para explorar a criatividade do aluno, o que acaba sugerindo a possibilidade de se utilizar da mecânica da programação visual, onde o usuário deve construir seus próprios componentes de software com base nos blocos do código. Com isso, nesta dissertação propomos uma abordagem que utiliza um jogo com mecânica de construção, onde o usuário pode desenvolver seus componentes de software com base nos blocos pré-produzidos, que são disponibilizados pelo jogo ou criados pelo jogador, e assim aprender conceitos básicos de desenvolvimento baseado em componentes.

4.2.2 Métricas para componentes

Uma métrica é uma medição de um atributo de uma determinada entidade. Elas servem para demonstrar e medir indícios de características específicas de uma entidade visando a melhoria de possíveis problemas, podendo ser utilizadas em diferentes dimensões, como esforço, tamanho, complexidade, dentre outras (FENTON & BIEMAN, 2014). Nessa dissertação, são utilizadas métricas para medir o nível de

reutilização de um componente de software, visando a construção de componentes melhores, de modo que os códigos criados sejam avaliados e pontuados de acordo com a qualidade gerada de acordo com as métricas utilizadas.

A maioria das métricas encontradas na literatura em relação a RS estão diretamente relacionadas ao custo-benefício do uso da reutilização. Elas fazem referência ao quanto de código pode ser gerado por uma equipe em um intervalo de tempo, ao custo de manutenção de código e ao custo por linha de código (BARNES *et al.*, 1988) (GAFFNEY & DUREK, 1989) (FRAKES & TERRY, 1995).

No entanto, este trabalho visa descrever métricas que servem como indicadores quantitativos para que seja possível medir o nível de reutilização que foi utilizado no desenvolvimento de um componente e o quão reutilizável esse componente seria, deixando de lado a relação de custo. Para identificação das métricas apresentadas na Tabela 4.3 foram estudados alguns trabalhos (REDOLFI *et al.*, 2004) (CALDIERA & BASILI, 1991) (GONÇALVES *et al.*, 2006) e utilizado o próprio conhecimento prévio do autor desta dissertação sobre o tema.

Tabela 4.3: Métricas para componentes reutilizáveis

ID	Nome	Definição	Fórmula
M1	Casos de teste	Número de casos de teste que o componente executou corretamente. Dividir o número de casos de teste corretos (NC) pelo número total de casos de teste (N) (REDOLFI <i>et al.</i> , 2004).	$X = NC / N.$
M2	Quantidade de esforço	A quantidade de esforço para gerar um componente pode ser derivada das contagens de operadores (n1) e operandos (n2) distintos e da frequência total de operadores (N1) e operandos (N2) (CALDIERA & BASILI, 1991) (HALSTEAD & OTHERS, 1977).	$N = (N1 + N2)$ $n = (n1 + n2)$ $Z = N * \log_2 n$ $Y = (n1/2) * (N2/n2)$ $X = Z * Y$
M3	Complexidade Ciclométrica	Complexidade Ciclométrica, também conhecida como Métrica de McCabe, basicamente, consiste em contar os testes de fluxo (TF) de um método (if, for, while, case, catch) (GONÇALVES <i>et al.</i> , 2006) (MCCABE, 1976).	$X = \sum TF$
M4	Incentivo de Reutilização	Caso o usuário utilize pelo menos um componente criado anteriormente em uma fase posterior, +1 ponto será acrescentado na pontuação total. Incentivando o usuário a reutilizar componentes.	$X = \sum Comp$

4.2.3 Descrição do jogo

O jogo proposto, denominado Reuse Blocks, foi inspirado no ambiente de programação Scratch (KALELIOGLU & GÜLBAHAR, 2014) por ser uma das principais linguagens visuais atuais e já possuir diversos usuário que fazem uso dessa plataforma de desenvolvimento (MALONEY *et al.*, 2010). O Reuse Blocks faz uso de uma interface e mecânica bem parecida com o Scratch, utilizando programação em blocos. No entanto, o Scratch foi desenvolvido com o intuito de ensinar programação para usuário iniciantes (BURNETT, 1999b). O jogo desenvolvido nesse trabalho foi criado com o intuito de ensinar reutilização por meio de componentes de software e ser utilizado por alunos de graduação.

Como já dito anteriormente, a mecânica desse jogo foi escolhida para dar mais experiência prática para o usuário e, com isso, todas as características que foram implementadas foram baseadas nas informações dos estudos realizados (Capítulo 3), sendo escolhidas características, como: desafio, pontuação, níveis, feedback constante, gráficos estimulantes, cenário real, simulação, dependência entre os conteúdos e o uso de recompensas. As características de simulação e cenário foram inseridas pensando que o usuário estará produzindo um código real baseado em um algoritmo real, simulando uma IDE visual. Deve-se ressaltar que o Reuse Blocks foi definido como um jogo com base nas características que o mesmo possui. No entanto, como a definição de jogos é algo um pouco controverso, alguns autores podem vir a discordar dessa definição, preferindo chamar de simulador ou outro termo.

Nesse jogo, o usuário deve criar os componentes que são requisitados em cada fase com base nos componentes / funções que são previamente disponibilizados pelo jogo, ou a partir de componentes que foram criados em fases anteriores (requisito elaborado com base em uma das sugestões feita por um dos avaliadores do jogo CodeBoy – Seção 4.1). No final de cada fase, o componente criado pelo jogador é avaliado com base nas métricas que foram definidas na Seção anterior, e a partir dos resultados, uma pontuação é gerada para o jogador, verificando assim o nível de reutilização do componente criado. A pontuação final é dada através da soma das

pontuações de cada uma das métricas, onde a métrica M1 pode alcançar até quatro pontos, a M2 três pontos, a M3 três pontos e a M4 pode somar um ponto extra.

Outra informação encontrada nos estudos realizados (Capítulo 3) foi em relação a importância do *feedback* para os alunos na hora do aprendizado. O jogo conta com o auxílio de dois sistemas de *feedback*, sendo o primeiro provido pela apresentação da pontuação para jogador e um segundo para auxiliar o jogador em tempo real, onde a cada novo bloco que fosse concatenado no código já produzido, o jogo executa uma pré-avaliação desse código para avaliar a sintaxe e verificar a possível futura ação do jogador. Por exemplo, caso o jogador insira um bloco *while*, o jogo irá avaliar o comando e recomendar que ele insira uma condição de parada nesse *while* e quando essa condição for preenchida, o jogo irá recomendar para que o *while* seja finalizado com uma chave, dando dicas sobre o próximo passo que deve ser executado pelo aluno.

A Figura 4.4 mostra um *blueprint* do jogo Reuse Blocks com a interface do jogo seguida de uma breve descrição de cada uma de suas partes. A primeira parte é a IDE (*Integrated Development Environment*) de blocos que é a parte onde o usuário deve criar seus componentes através do *drag and drop* de componentes primitivos (*if*, *else*, *while*, ...). A segunda parte é a IDE visual, onde os códigos são gerados através da decodificação dos componentes criados através dos blocos (dando *feedback* do que está sendo construído de maneira codificada). Por fim, a última parte é a área de resultado que provê o *feedback* para o usuário, mostrando mensagens de erro e pontuação. A Tabela 4.4 apresenta uma descrição do Reuse Blocks com base nos modelos OCR (GUARDIOLA, 2016) e MDA (HUNICKE *et al.*, 2004).

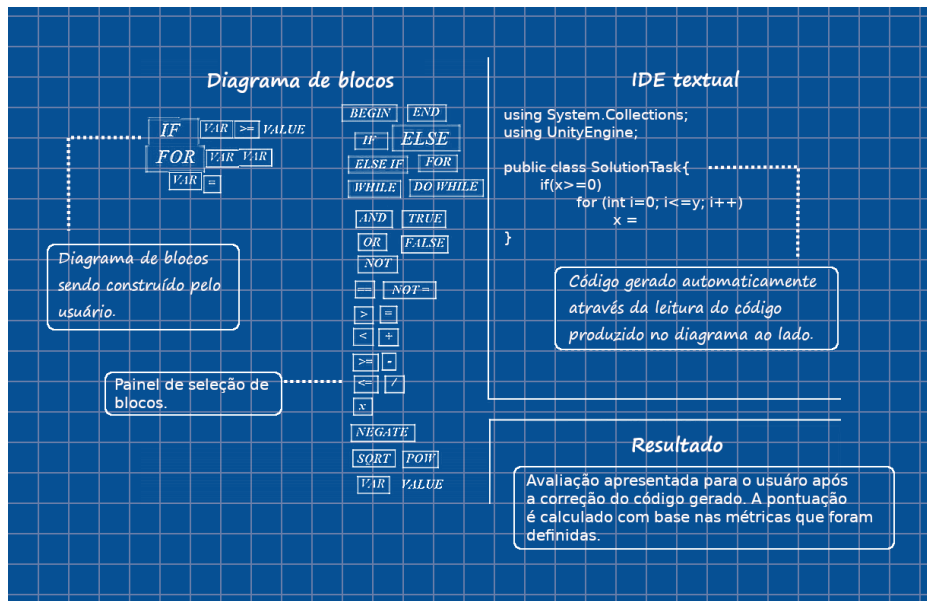


Figura 4.4: *Blueprint* do jogo Reuse Block. Fonte: do autor.

Tabela 4.4: Tabela de conceituação do jogo Reuse Blocks.

	Reuse Blocks
Objetivo	Ensinar reutilização por meio do desenvolvimento de componentes de software.
Desafio	Conseguir desenvolver os algoritmos propostos em cada uma das fases.
Recompensa	A nota de cada fase é realizada a partir do cálculo das métricas da Tabela 4.3.
Mecânica	A base do jogo é por meio da construção de componentes, onde o jogador deve arrastar e soltar blocos de códigos para resolver o algoritmo da fase. Vale lembrar que esses componentes criados podem ser salvos e reutilizados em fases posteriores.
Dinâmica	Após a criação do algoritmo, esse será validado por meio de métricas e será calculada uma nota para o jogador, incentivando que o usuário crie programas melhores para alcançar notas maiores
Estética	O usuário deve desenvolver seu próprio componente (expressão) com base no algoritmo descrito na fase, para que assim consiga uma nota adequada para avançar para a próxima fase (Desafio). Além disso, o jogo ainda pode ser visto como um passatempo para o jogador.

4.2.4 Comandos básicos

Todo o jogo funciona por meio de atalhos no teclado, onde existe uma série de comandos que o usuário precisa dominar para programar no Reuse Blocks, a Tabela 4.5 apresenta uma sumarização desses comandos.

Tabela 4.5: Comandos do jogo Reuse Blocks.

Ação	Comando
Criar bloco	Clicar duas vezes com o botão esquerdo do mouse sobre o bloco.
Mover o bloco	Segurar w - move bloco para cima; Segurar s - move bloco para baixo; Segurar a - move bloco para a esquerda; Segurar d - move bloco para a direita.
Destruir o bloco	Clicar com o botão esquerdo do mouse sobre o bloco para selecioná-lo + Digitar "delete" para destruir o bloco selecionado.
Configurar uma variável	Segurar v + clicar sobre o bloco da variável (Bloco Var). Preencher os campos referentes à configuração e, depois, digitar "Enter".
Configurar uma constante	Segurar v + clicar sobre o bloco da constante (Bloco Value). Preencher os campos referentes à configuração e, depois, digitar "Enter"
Para adicionar ";"	Segurar P + digitar "Enter".
Finalizar o bloco IF	Segurar h + digitar "Enter".
Finalizar o bloco ELSE	Segurar b + digitar "Enter".
Finalizar o bloco ELSE IF	Segurar j + digitar "Enter".
Finalizar o bloco WHILE	Segurar g + digitar "Enter"
Finalizar o bloco DO WHILE	Segurar z + digitar "Enter".
Finalizar o bloco FOR	Segurar c + digitar "Enter".
Finalizar o bloco POW	Segurar o + digitar "Enter".
Finalizar o bloco SQRT	Segurar i + digitar "Enter".
Bloco COMP	Arrastar o bloco COMP e concatenar os outros blocos.
Finalizar o bloco COMP	Segurar T e, depois, digitar "Enter".
Finalizar a fase:	Segurar f + digitar "Enter".
Calcular os valores das métricas	Apertar o botão "Calculate Level Score".

4.2.5 Exemplo prático

O Reuse Blocks foi construído com quatro fases e, por se tratar de um jogo a ser utilizado por pessoas iniciantes em programação, as fases construídas são algoritmos que não possuem uma complexidade muito alta, podendo ser resolvidas com poucas linhas de código. A Tabela a seguir descreve cada uma das fases do jogo. Uma imagem com uma possível solução para cada uma das fases será apresentada mais adiante.

Tabela 4.6: Fases do jogo Reuse Blocks.

Fase	Algoritmo	Explicação
1	Fibonacci	Gerar os primeiros 40 números da série de Fibonacci, sequência de números inteiros, começando por 0 e 1, na qual, cada termo subsequente corresponde à soma dos dois anteriores.
2	Ordenação	Dado um array X, deve-se ordenar o mesmo em ordem crescente.
3	Maior número	Dado um array X, deve-se retornar o maior número do array.
4	Somar array	Dado um array X, deve-se retornar um novo array onde o elemento n do array resultante é igual ao enésimo número do array original somado ao (tamanho do array - n)ésimo elemento. Exemplo: dado um array X igual a [2, 5, 4, 4, 5, 6], o algoritmo deve retornar o array [8, 10, 8].

As Figuras 4.5 e 4.6 mostram duas soluções para o problema de Fibonacci construídas com os operandos *for* e *while*. Com base nessas figuras, é possível observar que, ao realizar o problema com o operador *for*, o código fica menos eficiente em relação as métricas de quantidade de esforço (M3) e complexidade Ciclométrica (M4), mostrando que o uso de métrica ajuda na busca de uma solução algorítmica mais eficiente.

Block Diagram Interface

Task Level 1 Info: Build a solution for the Fibonacci series problem. It must be able to return the first forty numbers, attributing the latter to the predefined array variable named z.

```

BEGIN END AND OR TRUE FALSE SORT POW
IF WHILE DO WHILE == NOT = NEGATE
ELSE ELSE IF FOR NOT > >= < <=
VAR VALUE + - * / =
SORT
FOR VAR VALUE
IF VAR == VALUE OR VAR == VALUE
VAR = VALUE
ELSE
VAR = VAR + VAR
VAR = VAR
VAR = VAR

```

Code Interface

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class SolutionCodeTask {

    public int[] z = new int[100];

    private int i = 0;
    private int x = 1;
    private int y = 1;

    public void Main() {

        for (int i = 0; i <= 39; i++)
        {
            if (i == 0 || i == 1)
            {
                z[i] = 1;
            }
            else
            {
                z[i] = x + y;
                x = y;
                y = z[i];
            }
        }
    }
}

```

Output Interface

Necessary Effort Metric Result: 1
Cyclomatic Complexity (McCabe) Metric Result: 2
Final Score Result: 3

Figura 4.5: Reuse Blocks, algoritmo de Fibonacci implementado com FOR.

Fonte: do autor.

Block Diagram Interface

Task Level 1 Info: Build a solution for the Fibonacci series problem. Attribute its numbers to the predefined array variable named z.

```

BEGIN END AND OR TRUE FALSE SORT POW
IF WHILE DO WHILE == NOT = NEGATE
ELSE ELSE IF FOR NOT > >= < <=
VAR VALUE + - * / =
SORT
VAR = VALUE
VAR = VALUE
WHILE VAR VALUE
VAR = VAR + VAR
VAR = VAR
VAR = VAR

```

Code Interface

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class SolutionCodeTask {

    public int[] z = new int[40];

    private int i = 0;
    private int x = 1;
    private int y = 0;

    public void Main() {

        z[0] = 1;
        i = 1;

        while (i <= 39)
        {
            z[i] = x + y;
            y = x;
            x = z[i];
            i++;
        }
    }
}

```

Output Interface

Test Case Metric Result: 4
Necessary Effort Metric Result: 3
Cyclomatic Complexity (McCabe) Metric Result: 3
Final Score Result: 10

Figura 4.6: Reuse Blocks, algoritmo de Fibonacci implementado com WHILE.

Fonte: do autor.

A Figura 4.7 mostra o algoritmo do *Bubble Sort* implementado no Reuse Blocks. Com base nessa imagem, é possível observar que o algoritmo resolve o problema de ordenar o array, no entanto, em relação a métrica de quantidade de esforço ele não é tão eficiente, recebendo uma nota baixa.

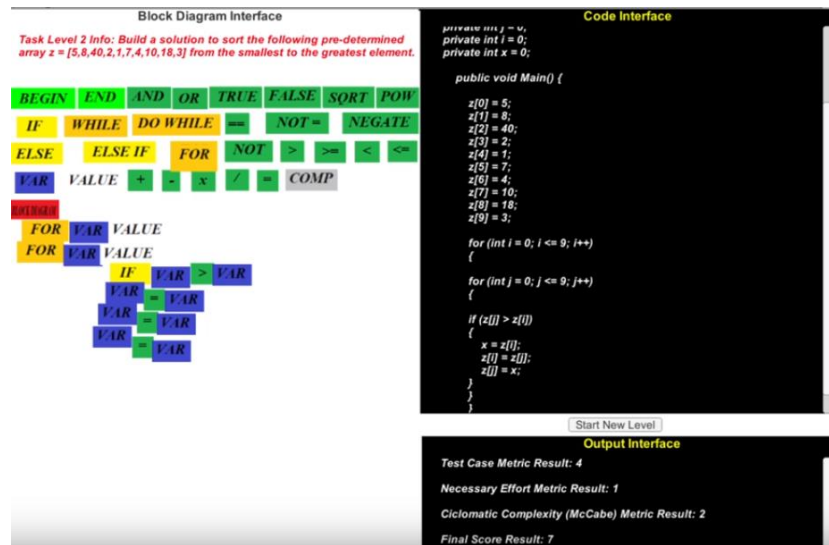


Figura 4.7: Reuse Blocks, algoritmo de ordenação. Fonte: do autor.

A Figura 4.8 mostra a solução para o algoritmo que, dado um array X, deve-se retornar o maior número dele. O algoritmo construído nessa fase ordena o array Z, e depois o percorre para pegar o último elemento. A partir dessa imagem, é possível observar o bloco COMP sendo utilizado, esse bloco representa um componente de software que pode ser reutilizado. Nesta imagem, o algoritmo que está na Figura 4.7 (fase 2) foi transformado em componente e reutilizado na fase 3, dando um ponto extra para o jogador.

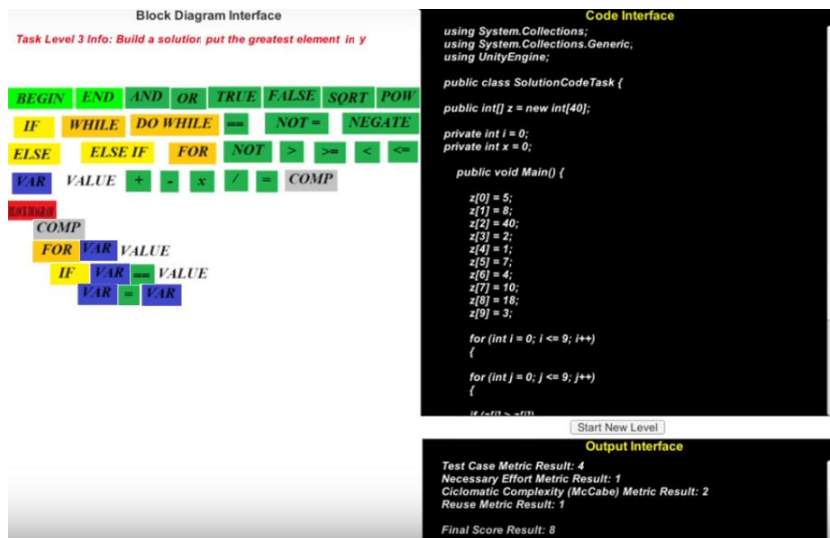


Figura 4.8: Reuse Blocks, algoritmo de retornar maior número. Fonte: do autor.

A Figura 4.9 mostra uma possível solução para o problema da fase 4 com pontuação máxima para todas as métricas, no entanto, como era um problema de simples implementação, não foi utilizado nenhum componente de fases anteriores, não recebendo a pontuação extra.

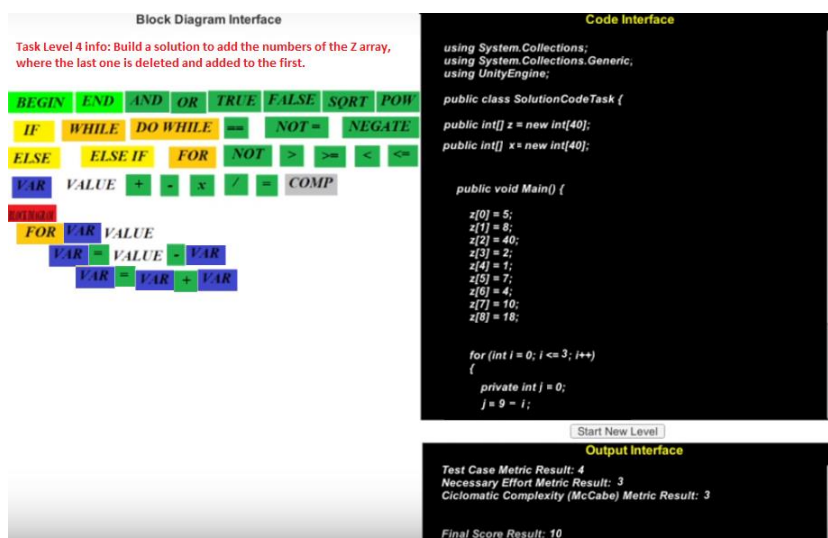


Figura 4.9: Reuse Blocks, algoritmo de soma de array. Fonte: do autor.

Por fim, a Reuse Blocks possui alguns blocos com algumas sub-ações que devem ser executadas pelo jogador. A Figura 4.10 mostra esses comandos - do lado esquerdo, é apresentado como definir uma variável e, do lado direito, como definir um valor para atribuição.



Figura 4.10: Definição de variáveis e valores no Reuse Blocks. Fonte: do autor.

4.3 Considerações Finais

Este capítulo teve como objetivo apresentar os jogos que foram criados e avaliados para serem utilizados como técnica de ensino de RS. A primeira etapa da produção desses jogos ocorreu pela execução de uma abordagem inicial, que contou com a construção de dois jogos. O primeiro deles sendo um *quiz* com perguntas sobre RS, contendo características como: *feedback* constante, pontuação, níveis de dificuldade, desafio, barra de progresso, tempo e dependência entre os conteúdos. Já o segundo jogo era do tipo *puzzle* e contava com características de: *feedback* constante, níveis de dificuldade, gráficos estimulantes, desafio, tempo, dependência entre os conteúdos e recompensas.

Esses dois jogos foram construídos para avaliar algumas informações relacionadas a utilização de jogos para o ensino de RS e para tentar solucionar os problemas dessa disciplina que, de acordo com o Capítulo 3, as principais são a falta de engajamento e motivação do aluno e a falta de prática. Após a avaliação de cada um deles, observou-se que o jogo do tipo *puzzle*, denominado CodeBoy foi melhor avaliado que o jogo do tipo *quiz*, no entanto, foi possível observar satisfação em relação a utilização dos dois jogos e ainda um aumento da motivação e do engajamento dos alunos que utilizaram os jogos.

A partir do indicativo de que os jogos poderiam melhorar o problema de engajamento e motivação do ensino de RS, o jogo Reuse Blocks foi criado para tentar apoiar na solução da questão prática. O jogo possui como principal mecânica a ideia de construção, a partir da observação feita na Seção 4.2.2 de que essa mecânica deve ser utilizada para explorar atividades práticas para o aluno. Além disso, o mapeamento do Capítulo 3 foi utilizado para escolher as principais características que se encaixavam na mecânica escolhida, com isso, as características de: desafio, pontuação, níveis, *feedback* constante, gráficos estimulantes, cenário real, simulação, dependência entre os conteúdos e recompensas foram utilizadas.

5 Avaliação

Neste capítulo, é apresentado um estudo com foco qualitativo sobre a utilização do jogo Reuse Blocks. Descrições sobre o planejamento, participantes, procedimento e resultados são apresentadas.

5.1 Planejamento

Na busca por indícios sobre a usabilidade e experiência proporcionada pelo jogo Reuse Blocks, foi executado um estudo com o objetivo de analisar a utilização do jogo para o ensino de RS com o propósito de avaliar a experiência proporcionada em relação aos ganhos de engajamento, motivação, divertimento e prática.

Nos estudos exploratórios do Capítulo 3, foram observados diversos métodos de avaliação, no entanto, a maioria deles giravam em torno de deixar o usuário utilizar a ferramenta e aplicar um questionário posterior. Na Seção 0, foi utilizado o questionário MEEGA (PETRI *et al.*, 2016) para a avaliação dos jogos iniciais. Esse questionário se mostrou bastante eficiente para uma avaliação pós jogo e, por isso, também foi utilizado nessa etapa do trabalho.

O procedimento de avaliação ocorreu no período de 13/02/2020 a 21/02/2020 e contou com uma avaliação piloto no dia 12/02/2020 com um participante para verificar se o jogo não havia problemas que fossem afetar a avaliação e também o tempo de execução que um candidato levaria para realizar o experimento, após a execução do piloto foi possível identificar que o procedimento demorava em média 30 minutos, não apresentando problemas que fossem afetar o experimento.

Devido ao período de execução e ao tempo relativamente alto do experimento optou-se pela execução remota do estudo, onde um executável e um e-mail com instruções básicas foi disponibilizado para os possíveis avaliadores. Além disso, o questionário MEEGA (PETRI *et al.*, 2016) foi reescrito e disponibilizado através do Google Forms.

5.2 Participantes

Para o estudo piloto utilizou-se apenas um aluno da graduação para validação do jogo. Para o estudo principal foram selecionadas duas amostras de alunos. A primeira amostra foi selecionada a partir de alunos de pós-graduação das universidades UERJ, UFRJ e ISCTE-IUL que haviam experiência de ensino, principalmente do grupo de pesquisa de Reutilização de Software da Coppe, da Universidade Federal do Rio de Janeiro (UFRJ). Essa população foi rotulada como a de especialistas. A segunda população foi extraída a partir de uma população de alunos de variadas universidades, tais como: UFRJ, UERJ, UFFRJ e UFF. A execução remota colaborou para essa segunda amostra, fazendo com que mais avaliadores tivessem acesso ao experimento.

5.3 Procedimento

Como dito anteriormente, o estudo ocorreu remotamente através da disponibilização do material necessário para a execução estudo, as etapas principais desse estudo são:

- **Descrição do jogo:** Essa etapa aconteceu por meio da disponibilização de um e-mail com as principais informações necessárias para a execução da avaliação, como: procedimento de instalação do jogo, comandos básicos do jogo e urls para o questionário e executável.
- **Caracterização dos participantes:** Um questionário de caracterização foi disponibilizado para cada um dos participantes. O questionário pode ser encontrado no Apêndice A desta dissertação.
- **Execução do jogo:** Os participantes desenvolveram o algoritmo no jogo Reuse Blocks.
- **Preenchimento do questionário qualitativo:** O questionário MEEGA (PETRI *et al.*, 2016) (removendo as perguntas para jogos *multiplayer*) foi

disponibilizado para cada um dos participantes. O questionário utilizado pode ser encontrado no Apêndice A.

- **Preenchimento do questionário educacional:** O questionário educacional foi disponibilizado para cada um dos participantes que foram considerados como especialistas (i.e., pessoas com experiência didática).

Como o estudo foi executado remotamente só foi possível executar o protocolo *think-aloud* (JÄÄSKELÄINEN, 2010) com apenas 7 participantes com o auxílio do Skype. Esse protocolo foi utilizado para tentar obter algumas informações extras, tais como: em qual etapa o jogador teve mais dificuldade, quanto tempo o jogador demorou para passar cada uma das fases, entre outras informações. A partir desse protocolo, também, foi possível a captura do som do ambiente, o que auxiliou no entendimento de algumas sensações vivenciadas pelo jogador através de suas reações, como: reivindicações, suspiros e demonstrações de cansaço ou estresse.

5.4 Resultados da avaliação

5.4.1 Caracterização dos participantes

O estudo contou com a participação de 29 avaliadores, sendo estes categorizados em 3 grupos: especialista, que foram definidos como usuários de pós-graduação com experiência didática; alunos de graduação remotos e alunos de graduação observados que fizeram uso do protocolo *think-aloud* (JÄÄSKELÄINEN, 2010).

Como o estudo foi executado remotamente, sendo disponibilizado para diversos alunos, não foi possível saber exatamente quem iria responder o questionário. Por isso, antes da avaliação foi necessário realizar uma filtragem e caracterização dos participantes para descobrir em qual grupo cada participante se encaixava. Nessa filtragem, foi observado que 3 alunos de pós-graduação sem experiência didática

avaliaram o jogo, como esses não se encaixavam em nenhum dos grupos pré-determinados, foram excluídos da avaliação.

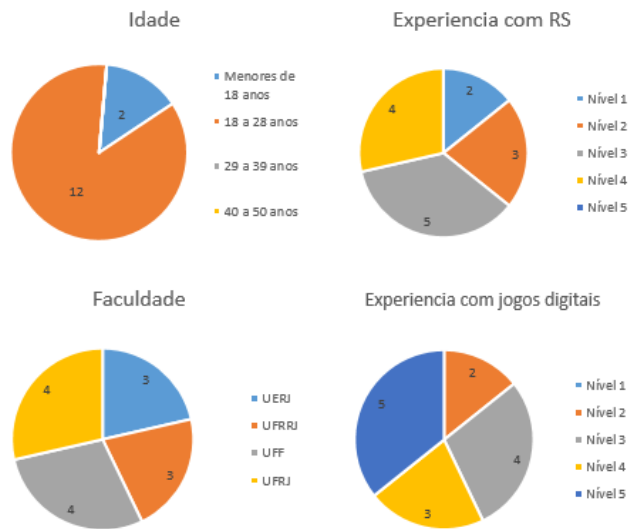


Figura 5.1: Caracterização dos participantes remotos. Fonte: do autor.

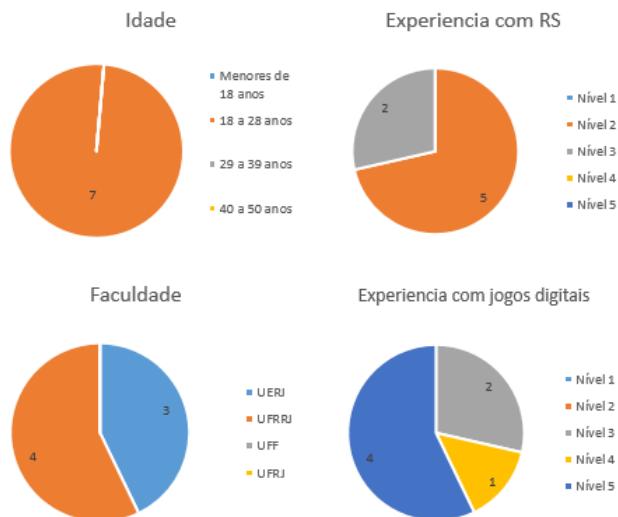


Figura 5.2: Caracterização dos participantes observados. Fonte: do autor.

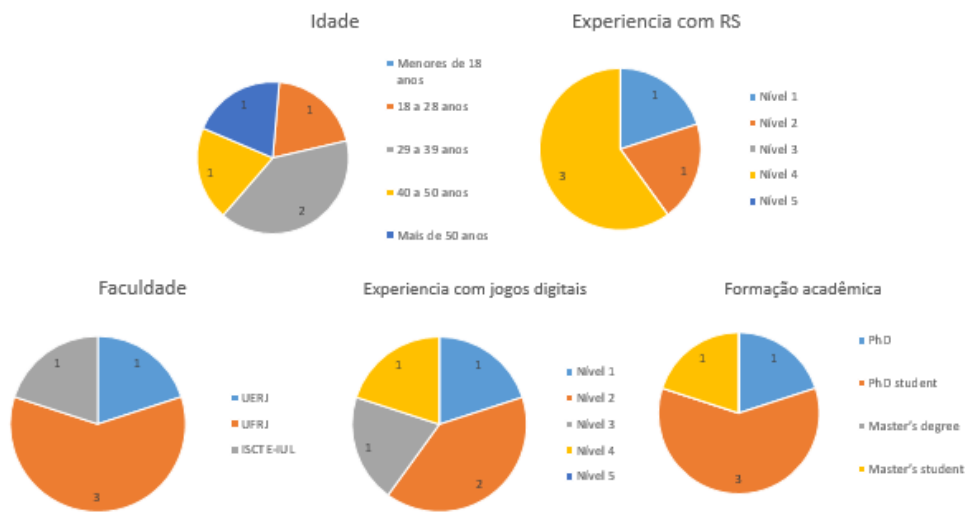


Figura 5.3: Caracterização dos participantes especialistas. Fonte: do autor.

As Figuras 5.1, 5.2 e 5.3 mostram as caracterizações dos participantes de cada um dos grupos. Os grupos de alunos remotos e observados só contaram com alunos de graduação. O grupo de alunos remotos contou com catorze participante sendo doze deles com idades entre 18 e 28 anos e apenas dois com menos de 18. Em questões de experiência em RS e jogos digitais, metade possuíam uma boa experiência. Essa amostra contou com participantes das faculdades UERJ, UFRRJ, UFF e UFRJ. O grupo de alunos observados contou com sete participantes das faculdades UERJ e UFRJ com idades entre 18 e 28 anos, cinco participantes apresentaram nível 2 em RS e dois com nível 3. Em relação a experiência com jogos digitais, mais da metade apresentou nível máximo. Por fim, o grupo de especialistas contou com 5 participantes das Faculdades UERJ, UFRJ e ISCTE-IUL. Três deles eram alunos de doutorado, um possuía pós-doutorado e um era aluno de mestrado, todos possuindo experiências didáticas. No entanto, em relação a experiência com jogos digitais, esse grupo contou com apenas dois participantes com uma boa experiência na área, em RS quatro participantes responderam possuir uma boa experiência na área.

5.4.2 Análise dos resultados

As Figuras 5.4 e 5.5 apresentam as informações referentes a cada uma das perguntas da avaliação. A partir da primeira coluna delas, que é em relação aos usuários remotos, é possível observar que em questões de usabilidade o jogo recebeu uma avaliação mediana, existindo melhorias a serem feitas. Um questionamento bastante recorrente foi em relação a movimentação dos blocos no jogo, grande parte dos usuários pediram por uma nova versão onde o mouse fosse utilizado. Outro ponto observado foi em relação a interface do jogo, muitos usuários levantaram questionamentos sobre a mesma, comentando sobre a necessidade de uma interface mais elaborada para chamar a atenção do aluno. Por fim, a grande quantidade de comandos e a complexidade do jogo também foi apontada como um ponto negativo, fazendo com que muitos usuários tivessem dificuldades de utilizar o jogo, essa preocupação relacionada a complexidade do jogo foi identificada no Capítulo 3 e acabou por se manifestar nesse estudo. Em questões de experiência, o jogo também foi avaliado como mediano, deixando metade da amostra engajada, tendo experiências fortes de desafio, satisfação e diversão. O principal problema apontado foi em relação a experiência de utilização, que devido à grande quantidade de comandos do jogo muitos usuários o definiram como complexo.

Um problema bastante recorrente entre os usuários remotos foi em relação a inicialização do ambiente do jogo. Era necessário a criação de uma pasta temporária no computador para guardar os scripts do jogo. Embora este ponto tenha sido informado por e-mail, muitas pessoas acabaram não se atentando a esse requisito. Outro problema vivenciado pelos usuários remotos foi em relação a utilização do jogo, devido ao mesmo possuir muitos comandos, vários usuários precisaram ficar alternando entre o jogo e a lista de instruções.

Os problemas destacados acima foram vivenciados com menor intensidade pela população que foi observada, devido a não enfrentar o problema de criar o ambiente e pela indicação do autor em requisitar que todos os usuários tivessem em mãos o documento de comandos do jogo. Isso pode ter feito com que as pontuações desses usuários fossem maiores do que as pontuações de quem utilizou por conta própria, conforme pode ser observado na segunda coluna das Figuras 5.4 e 5.5. No entanto,

devido ao protocolo *think-aloud* foi possível perceber bastante insatisfação dos usuários em relação a movimentação dos blocos, essa tarefa se mostrou bastante cansativa.

A última coluna das Figuras 5.4 e 5.5 mostram as pontuações enviadas pelos especialistas. A partir dela é possível observar uma avaliação um pouco mais negativa tanto em questões de usabilidade e experiência, apesar dessa ser a amostra com menor número de avaliadores, foi possível perceber um questionamento em relação a complexidade e utilização do jogo. Vale lembrar que os especialistas também executaram o experimento remotamente, passando pelas mesmas dificuldades da população de alunos remota.

Apesar dos problemas destacados pelos avaliadores, grande parte deles, contando com 19 participantes, gostaram da iniciativa em utilizar jogos para o ensino de reutilização e do jogo Reuse Blocks. Os principais pontos destacados foram em relação a iniciativa, da qualidade de transcrição do código em bloco para o textual, do uso das métricas para pontuação, da programação visual e dos *feedbacks* providos pelo jogo, principalmente em relação a construção da sintaxe do código.

Dado que a avaliação do Reuse Blocks não ocorreu em sala de aula, o questionário para os especialistas também contou com algumas perguntas extras para entender os principais problemas vivenciados no ensino da disciplina de reutilização e tentar descobrir se o jogo proposto poderia auxiliar a contornar esses problemas. O questionário educacional pode ser observado no Apêndice B (Questionário Educacional).

Dentre os cinco avaliadores, quatro deles já utilizaram jogos dentro de sala e acreditam que essa seja uma estratégia muito eficiente para ensinar. Dentre os jogos que já foram utilizados, destacam-se o Kahoot, o Mip App Inventor e o Second Life. Dentre os problemas de ensino de RS listados pelos especialistas, destacam-se a falta de ferramentas instrucionais para apoiar o ensino, a falta de exemplos complexos próximos da vida real e a falta de prática, sendo estes alguns dos problemas que o Reuse Blocks visa resolver.

A última pergunta desse questionário procurava entender se os avaliadores recomendariam o uso do Reuse Blocks dentro de sala ou como instrumento de apoio, um dos avaliadores não soube opinar e os outros quatro responderam que sim, afirmando que o jogo era didático e que poderia ajudar no aprendizado. No entanto, precisaria de algumas modificações, tais como, melhoria do layout e utilização do mouse como ferramenta de movimentação dos blocos. Por fim, um dos avaliadores acrescentou dizendo que aprovaria o uso de blocos de reutilização em sala de aula se expandisse o jogo atual e apresentasse mais conceitos, como: Design Patterns, Arquitetura e Inversão de Controle.

A partir da avaliação foi possível encontrar melhorias a serem aplicadas ao jogo proposto, entre elas se destacam a melhoria da interface do jogo, visando ter um *layout* mais atrativo para o jogador, a minimização dos comandos do jogo, tentando melhorar a usabilidade do mesmo, deixando-o menos complexo e por fim utilização do *mouse* para o movimento de *drag and drop* dos blocos, melhoria essa que já foi realizada.

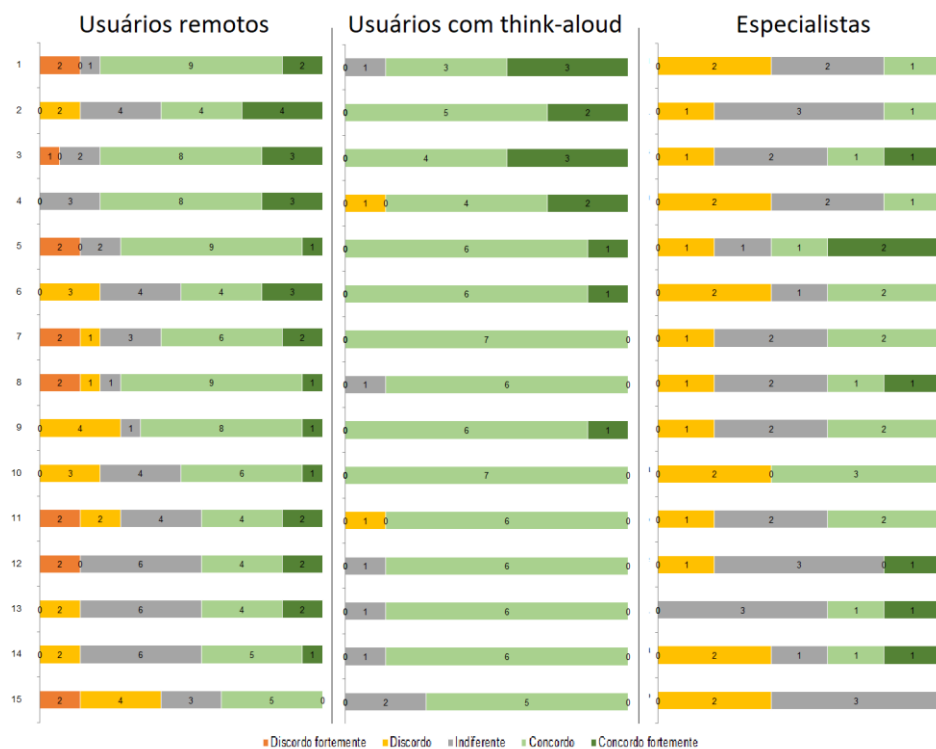


Figura 5.4: Avaliação de experiência, adaptado de PETRI *et al.* (2019).

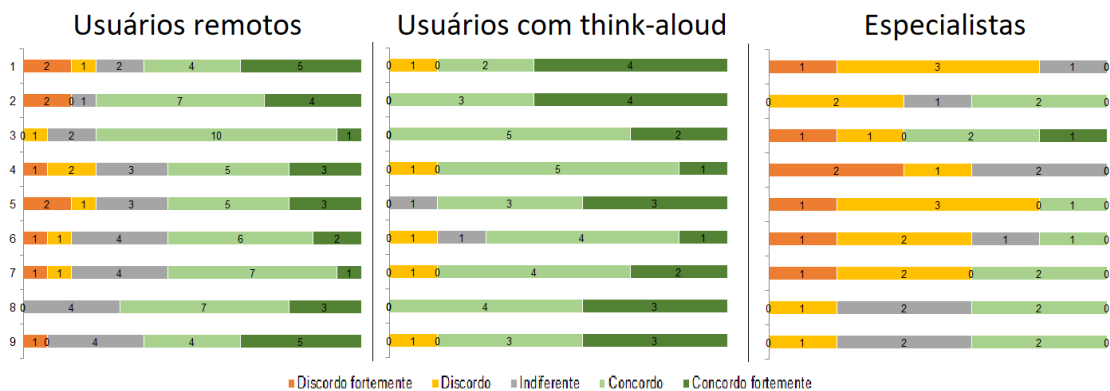


Figura 5.5: Avaliação de usabilidade, adaptado de PETRI *et al.* (2019).

5.4.3 Testes estatísticos

O questionário MEEGA (PETRI *et al.*, 2016) visa avaliar um jogo em questões de usabilidade e experiência vivenciada, no entanto, ele não faz nenhuma hipótese dos relacionamentos entre os atributos da população com os resultados obtidos pelas perguntas do questionário. Buscando testar algumas hipóteses entre as respostas do questionário e as respostas do questionário de caracterização, mais uma análise dos dados foi feita, dessa vez utilizando testes estatísticos. As hipóteses testadas podem ser observadas a seguir:

1. Existe alguma interferência da experiência em reutilização de software do avaliador em relação as respostas obtidas no questionário MEEGA?
2. Existe alguma interferência da experiência em jogos digitais do avaliador em relação as respostas obtidas no questionário MEEGA?
3. Existe alguma interferência da formação acadêmica do avaliador em relação as respostas obtidas no questionário MEEGA?
4. Existe alguma interferência da idade do avaliador em relação as respostas obtidas no questionário MEEGA?

5. Existe alguma interferência da instituição de ensino do avaliador em relação as respostas obtidas no questionário MEEGA?
6. Existe alguma interferência do grupo da amostra que o avaliador participou em relação as respostas obtidas no questionário MEEGA?

Um teste estatístico é um procedimento que permite tomar uma decisão entre duas ou mais hipóteses, utilizando os dados observados de um experimento. Existem diversos tipos de testes que podem ser feitos, no entanto, a escolha do teste fica atrelada com tipo de informação que se deseja obter e com o tipo de informações que foram utilizadas / obtidas no experimento (HALSTEAD & OTHERS, 1977). Nesse trabalho, os testes estatísticos foram aplicados para encontrar as dependências entre as informações sobre a população e as respostas obtidas.

Na avaliação do Reuse Blocks, foi utilizada a escala Likert como padrão de respostas das questões. Essa escala se caracteriza por valores não contínuos, o que dificulta a aplicação desses testes (AGUIAR *et al.*, 2011). Para resolver esse problema foi utilizado um método aditivo, que soma todas as respostas de um determinado candidato e observa o resultado geral, transformando uma escala ordinal em uma escala contínua (CAMELO *et al.*, 2016). Essa abordagem foi utilizada para facilitar os cálculos que seriam feitos pelos testes estatísticos. Uma escala ordinal é limitada a oferecer resultados em porcentagem, mediana e moda, não sendo possível descobrir algumas medidas como, por exemplo, o desvio e a média. Utilizando o método aditivo é possível criar dados contínuos que facilitam a aplicação de algum teste estatístico.

Para a execução dos testes foi utilizada uma ferramenta estatística da IBM denominada *Statistical Package for the Social Sciences (SPSS)* (FIELD, 2013). A primeira etapa para execução da análise era verificar a normalidade dos dados, como os dados da escala Likert não são contínuos, aplicou-se o método aditivo para somar todas as respostas de cada um dos candidatos. Como o questionário contava com 26 perguntas, com padrão de resposta da escala likert dentro do intervalo de [-2 e 2], é possível descobrir os valores que o método aditivo deve retornar, basta multiplicar o número de

perguntas pelo máximo e mínimo do intervalo, resultando em valores entre [-52 e 52]. Uma vez que a escala contínua foi construída foi possível aplicar o teste de normalidade. As Figuras 5.6 e 5.7 mostram o teste de normalidade.

Testes de Normalidade

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estatística	gl	p valor	Estatística	gl	p valor
resultado	,137	26	,200*	,877	26	,005

* Este é um limite inferior da significância verdadeira.

	Estatística	Estatística do teste Padrão
resultado Média	13,9615	3,76522
95% de Intervalo de Confiança para Média	Limite inferior	6,2069
	Limite superior	21,7161
5% da média aparada	15,2094	
Mediana	17,0000	
Variância	368,598	
Erro Padrão	19,19892	
Mínimo	-30,00	
Máximo	35,00	
Amplitude	65,00	
Amplitude interquartil	24,00	
Assimetria	-1,069	,456
Curtose	,417	,887

Figura 5.6: Teste de normalidade, adaptado de FIELD (2013).

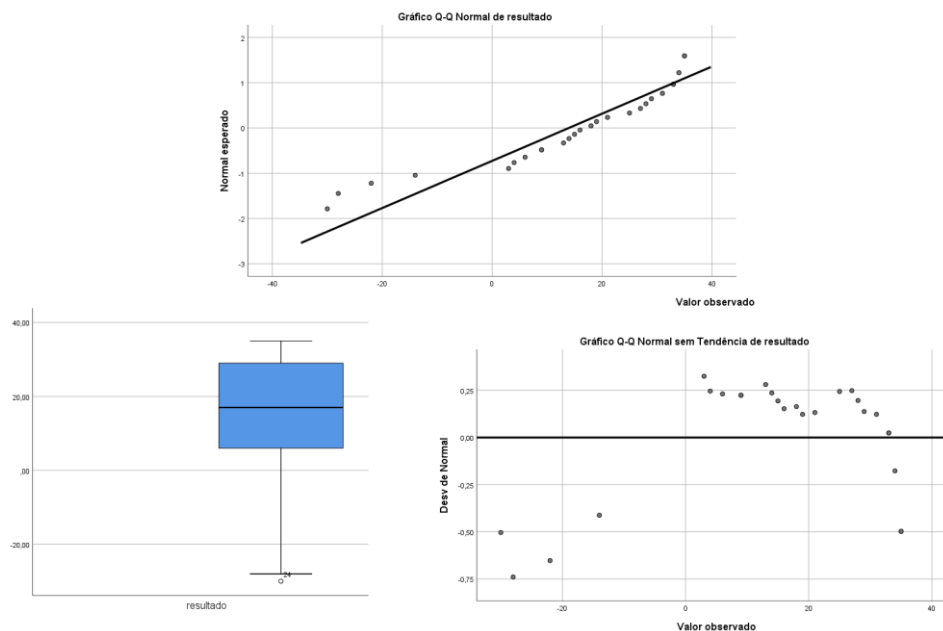


Figura 5.7: Gráficos de normalidade, adaptado de FIELD (2013).

A partir da análise dos gráficos e dos testes de Shapiro Wilk (LOPES *et al.*, 2013) e Kolmogorov-Smirnov (LOPES *et al.*, 2013) que são os mais convencionais para teste de normalidade, é possível observar que os dados não seguem uma distribuição normal, sendo recomendado que sejam aplicados testes não paramétricos¹⁰. O teste de Kruskal-Wallis (KW) é um teste não paramétrico recomendado para ser utilizado para comparar três ou mais populações e é usado para verificar hipóteses relacionadas a uma determinada população (BRESLOW, 1970). Em outras palavras, esse teste foi utilizado para verificar se os atributos do questionário de caracterização possuem influência em relação as respostas obtidas pelo questionário de avaliação. A Figura 5.8 mostra o teste KW (BRESLOW, 1970) para cada uma das seis hipóteses levantadas.

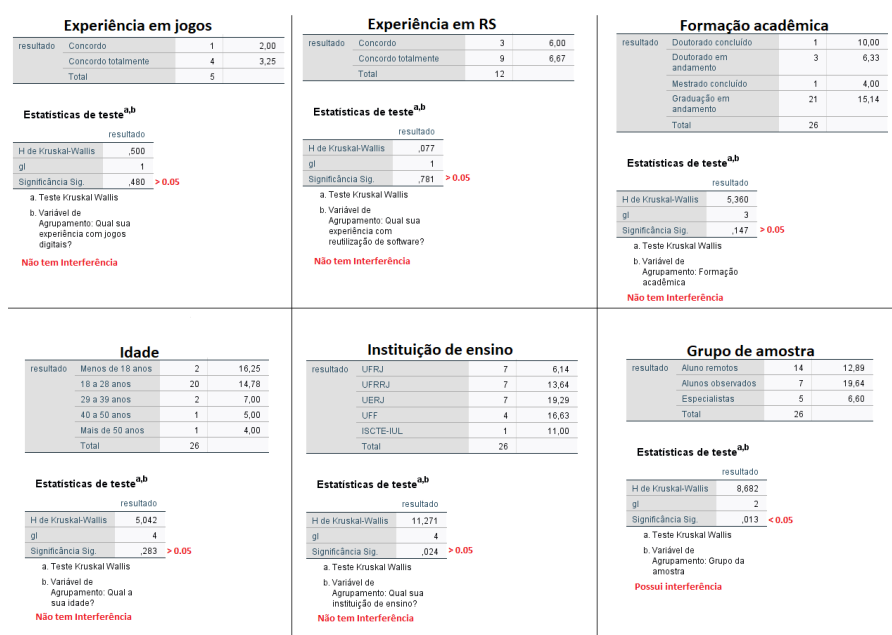


Figura 5.8: Teste de Kruskal-Wallis (BRESLOW, 1970), adaptado de FIELD (2013).

¹⁰ Testes não paramétricos: Testes para uma amostra que não tem dependência de parâmetros, cuja interpretação não depende da população, cabendo assim qualquer distribuição parametrizada.

Por convenção, foi utilizado uma margem de erro de 5%, ou seja, valores de significância menores que 5% correspondem a atributos da amostra que interferem no resultado do questionário de avaliação. Devido a alguns padrões na amostra utilizada, alguns fatores não foram considerados relevantes para o teste. Por exemplo, como mais da metade dos participantes tinham idade entre 18 e 28 anos, o atributo idade acabou tendo um valor maior do que a margem de erro. Como o fator de experiência em jogo e reutilização acaba ficando a critério de uma autoanálise do avaliador, houve muitos candidatos que acabaram colocando valores medianos, o que acabou gerando um valor maior que a margem de erro para o teste. O atributo universidade já era esperado que não fosse ser um valor que fosse interferir nas respostas dos candidatos, mas mesmo assim foi avaliado. A formação acadêmica acabou sendo descartada pelo teste devido à grande parte da amostra estar na graduação. Por fim, o teste acabou indicando que o atributo de grupo de amostra interferia nos resultados do questionário, o que já era esperado. No entanto, é importante ressaltar que devido a amostra possuir apenas 29 participantes, os testes estatísticos mostraram que alguns fatores, como idade e nível da formação acadêmica, não foram considerados relevantes, sendo que se o estudo fosse executado com uma amostra maior esse resultado poderia ser diferente.

Como foi comprovado pelo teste de KW que o atributo do grupo de amostra tinha interferência sobre os resultados da pesquisa, o teste de Bonferroni (BLAND & ALTMAN, 1995) foi aplicado. Este teste foi utilizado para verificar a significância estatística de cada um dos valores do grupo de amostra testados dois a dois. A Figura 5.9 mostra o teste de Bonferroni, com base na coluna de diferença média, é possível observar que os dois valores mais controversos são os alunos observados em relação aos especialistas, o que indica que a maior variação de respostas está entre esses dois grupos, demonstrando que algum fator interferiu nos resultados dessas duas equipes. Outra informação que é possível compreender é que a segunda maior variação de respostas está entre os alunos remotos e os alunos observados. A partir desses dados, é possível confirmar que os alunos observados tiveram uma experiência melhor com o jogo por não enfrentarem alguns problemas que foram enfrentados pela população remota (alunos remotos e especialistas).

Comparações múltiplas

Variável dependente: resultado
Bonferroni

(I) Grupo da amostra	(J) Grupo da amostra	Diferença média (I-J)	Estatística do teste Padrão	Sig.	Intervalo de Confiança 95%	
					Limite inferior	Limite superior
Aluno remotos	Alunos observados	-15,00000	7,92547	,213	-35,4637	5,4637
	Especialistas	13,77143	8,91983	,409	-9,2597	36,8026
Alunos observados	Aluno remotos	15,00000	7,92547	,213	-5,4637	35,4637
	Especialistas	28,77143*	10,02502	,026	2,8867	54,6562
Especialistas	Aluno remotos	-13,77143	8,91983	,409	-36,8026	9,2597
	Alunos observados	-28,77143*	10,02502	,026	-54,6562	-2,8867

*. A diferença média é significativa no nível 0.05.

Figura 5.9: Teste de Bonferroni (BLAND & ALTMAN, 1995), adaptado de FIELD (2013).

5.4.4 Considerações Finais

A partir do desenvolvimento e avaliação do jogo Reuse Blocks foi possível observar que apesar dos problemas de *layout* e usabilidade do jogo, o mesmo foi considerado como didático, oferecendo desafios engajadores para os participantes, sendo recomendado para ser utilizado em sala de aula para ensinar RS para alunos em formação.

A partir dos testes estatísticos, foi possível observar que os atributos da amostra não acabaram influenciando no resultado final da análise, sendo impactada somente pelo atributo do grupo da amostra (alunos remotos, alunos observados e especialistas). Por este motivo, é possível inferir que a avaliação poderia ter sido feita com apenas dois grupos, um remoto e outro com observação.

No Capítulo 3, foi demonstrado que os jogos para ensino são utilizados principalmente como métodos que apoiam o ensino. Isso também pode ser observado através da avaliação e dos testes estatísticos do jogo Reuse Blocks, uma vez que o grupo de alunos observados demonstrou uma satisfação maior em relação ao jogo.

Outro ponto observado foi em relação ao grau de experiência dos participantes, como essas respostas ficam a critério de uma autoavaliação do avaliador, muitos usuários acabaram colocando respostas medianas ou baixas para experiências em relação a RS, de um total de vinte e seis participantes, sete responderam ter experiência nível três e doze responderam ter pouca experiência (nível um e dois). Já em relação a experiência com jogos digitais, quinze dos vinte e seis participantes responderam ter uma experiência alta na área, sendo considerado como alto, os níveis 4 e 5. Como a maioria dos participantes responderam ter uma experiência média ou baixa em RS e o jogo acabou sendo considerado como bom no resultado final, é possível inferir que o jogo apesar dos problemas encontrados na avaliação alcançou seu objetivo que era apoiar o ensino de RS de maneira engajadora e prática para alunos iniciantes.

6 Conclusão

Esta dissertação procurou demonstrar a utilização dos jogos para o ensino de RS. Para que isso fosse possível, um estudo exploratório foi executado para entender o que já havia sido produzido na área. O estudo revelou informações importantes, tais como: quais são as vantagens de se utilizar jogos para o ensino, quais as principais características utilizadas nesses jogos educativos, quais são as vantagens e desvantagens desses jogos, além de indicar as principais dificuldades no ensino de RS.

A partir das dificuldades no ensino de RS (motivação / engajamento e prática) e das vantagens oferecidas pela utilização de jogos no ensino de ES, em geral (maior motivação / engajamento e prática), foi possível perceber que o uso de jogos no ensino de RS poderia ser uma boa estratégia a ser pesquisada. No entanto, nenhum jogo havia sido criado para o ensino dessa disciplina. Como não foi encontrado nenhum jogo com o propósito de ensino de RS, optou-se pelo desenvolvimento de dois jogos iniciais para avaliar a usabilidade e experiência proporcionada por cada um dos jogos como estratégia de ensino para RS. Os jogos criados foram construídos e avaliados, demonstrando bons resultados para o ensino de reutilização em relação a usabilidade e experiência proporcionada e ainda apresentando o aumento da motivação e engajamento dos alunos como uma das suas principais vantagens.

Com a demonstração dos bons resultados obtidos com o uso dos jogos para o ensino de RS e do possível tratamento das questões de engajamento e motivação a partir da construção dos jogos iniciais, restaria ainda a questão da prática. Nos estudos exploratórios realizados, foram encontradas ferramentas com características muito semelhantes a jogos. Essas ferramentas utilizavam linguagens visuais, onde o usuário deveria escrever um programa apenas arrastando blocos. Uma importante subárea da reutilização é o desenvolvimento baseado em componentes, onde o desenvolvedor deve criar programas a partir de trechos de código pré-produzidos. Fazendo uma comparação entre o desenvolvimento baseado em componentes e as linguagens visuais, é possível perceber semelhanças, na medida em que as duas abordagens visam criar programas a partir de “blocos” ou “componentes” pré-produzidos.

Na tentativa de tratar as questões relacionadas ao ensino de RS, inclusive a de prática, mais um jogo foi criado, dessa vez fazendo uso de programação visual. O jogo criado foi denominado Reuse Blocks e visava ensinar RS de maneira prática e divertida por meio do uso de componentes de software. O jogo foi avaliado por vinte e seis participantes, sendo cinco deles considerados como especialistas. Por meio dessa avaliação, foi possível encontrar alguns problemas relacionados ao *layout* do jogo e a usabilidade. No entanto, o jogo foi considerado como didático e divertido, apresentando experiências fortes de desafio e satisfação para os usuários.

6.1 Contribuições

Dentre as principais contribuições alcançadas por este trabalho, destacam-se:

- Identificação das principais dificuldades no ensino de reutilização de software (Revisão de Literatura);
- Identificação dos principais recursos do uso de jogos educacionais com ênfase no ensino de engenharia de software (Revisão de Literatura);
- Identificação dos principais jogos e suas características utilizadas no ensino de engenharia de software (Revisão de Literatura);
- Identificação de jogos de programação, que podem ser utilizados no ensino de conceitos de reutilização de software, mais especificamente no desenvolvimento baseado em componentes (Revisão de Literatura);
- Implementação e avaliação dos jogos iniciais, i.e., CodeBoy e Reuse Quiz, que visavam identificar algumas informações relacionadas a criação de jogos para o ensino de RS, tais como: tipo de jogo mais adequado, experiência proporcionada pelo jogo, satisfação fornecida pelo jogo, entre outras;
- Implementação e avaliação do jogo Reuse Blocks, que visou o ensino de reutilização de software de uma maneira prática e divertida, por meio da avaliação de componentes de software criados através de uma IDE visual.

6.2 Limitações

Limitações podem ser identificadas ao realizar uma análise crítica sobre este trabalho de pesquisa. Dentre as principais limitações, destacam-se:

- Os estudos exploratórios realizados utilizaram alguns filtros de busca visando reduzir o escopo, o que pode ter reduzido os trabalhos encontrados apesar do método de *snowballing* (WOHLIN, 2014).
- O jogo Reuse Blocks não foi avaliado em sala de aula.
- Devido ao período em que o experimento foi executado, optou-se pela avaliação do jogo Reuse Blocks remotamente, o que acabou resultando negativamente na experiência dos alunos.
- A avaliação do Reuse Blocks contou com apenas 26 participantes, sendo apenas 5 deles considerados especialistas, o que foi uma amostra relativamente pequena.

6.3 Trabalhos Futuros

Trabalhos futuros foram identificados no decorrer da realização deste trabalho de pesquisa. Dentre os principais trabalhos futuros, destacam-se:

- Como foi observado na Seção 4.2.2, outra mecânica que possibilita mais aprendizagem prática é a utilização de jogos com simulação. Por isso, poder-se-ia criar um jogo com simulação para verificar o potencial de ensino dessa mecânica para o ensino de RS.
- Os jogos construídos como parte desta dissertação não seguiram um processo de construção bem definido, sendo desenvolvido com base na experiência do autor. Isso ocorreu devido a não ser encontrado um processo amplamente utilizado para a construção de jogos educativos, sendo uma área de pesquisa promissora.

- Na avaliação do Reuse Blocks, foram identificados alguns problemas, com destaque para a movimentação e *layout* do jogo. Pretende-se, no futuro, continuar com o desenvolvimento do jogo para melhorar os problemas apontados.

Referências

- ADAMS, E., DORMANS, J., 2012, "Game mechanics: advanced game design". In: *New Riders*.
- AGALBATO, F., 2017, "Design and implementation of Robo3: an applied game for teaching introductory programming". In: . pp. 98–102.
- AGALBATO, F., LOIACONO, D., 2018. "Robo 3: A Puzzle Game to Learn Coding". In: *2018 IEEE Games, Entertainment, Media Conference (GEM)*. Galway, Ireland: IEEE. 2018. pp. 359–366.
- AGUIAR, B., CORREIA, W., CAMPOS, F., 2011, "Uso da escala likert na análise de jogos". In: *Salvador: SBC-Proceedings of SBGames Anais*. v. 7.
- ALBILALI, A.A., QURESHI, R.J., 2016, "Proposal to Teach Software Development Using Gaming Technique". In: *International Journal of Modern Education and Computer Science*. v. 8, pp. 21–27.
- ALHAMMAD, M.M., MORENO, A.M., 2018, "Gamification in software engineering education: A systematic mapping". In: *Journal of Systems and Software*. v. 141, pp. 131–150,.
- ATAL, R., SUREKA, A., 2015. "Anukarna: A Software Engineering Simulation Game for Teaching Practical Decision Making in Peer Code Review.". In: *QuASoQ/WAWSE/CMCE@ APSEC*. S.l.: s.n. 2015. pp. 63–70.
- AYDAN, U., YILMAZ, M., CLARKE, P.M., et al., 2017, "Teaching ISO/IEC 12207 software lifecycle processes: a serious game approach". In: *Computer Standards & Interfaces*. v. 54, pp. 129–138.
- BAKER, A., NAVARRO, E.O., HOEK, A.V.D., 2005, "An experimental card game for teaching software engineering processes". In: *Journal of Systems and Software*. v. 75, pp. 3–16,.
- BARNES, B., DUREK, T., GAFFNEY, J., et al., 1988, *A framework and economic foundation for software reuse*. . S.l., Software reuse: emerging technology.
- BAU, D., 2015a, "Droplet, a blocks-based editor for text code". In: *Journal of Computing Sciences in Colleges*. v. 30, pp. 138–144.
- BAU, D.A., 2015b. "Integrating Droplet into Applab—Improving the usability of a blocks-based text editor". In: *2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond)*. NW Washington, DC United States: IEEE. 2015. pp. 55–57.

BERKLING, K., THOMAS, C., 2013. "Gamification of a Software Engineering course and a detailed analysis of the factors that lead to it's failure". In: *Interactive Collaborative Learning (ICL), 2013 International Conference on*. S.l.: IEEE. 2013. pp. 525–530.

BLAND, J.M., ALTMAN, D.G., 1995, "Multiple significance tests: the Bonferroni method". In: *Bmj*. v. 310, pp. 170.

BOLDBAATAR, N., ŞENDURUR, E., 2018, "Developing Educational 3D Games With StarLogo: The Role of Backwards Fading in the Transfer of Programming Experience". In: *Journal of Educational Computing Research*. v. 57, pp. 1468--1494.

BOLLER, S., KAPP, K., 2017, *Play to learn: Everything you need to know about designing effective learning games*. . S.l., Association for Talent Development.

BRESLOW, N., 1970, "A generalized Kruskal-Wallis test for comparing K samples subject to unequal patterns of censorship". In: *Biometrika*. v. 57, pp. 579–594.

BURNETT, M.M., 1999, "Visual programming". In: *Wiley Encyclopedia of Electrical and Electronics Engineering*.

CAILLOIS, R., 2015, *Les jeux et les hommes. Le masque et le vertige*. . S.l., Editions Gallimard.

CALDERÓN, A., RUIZ, M., 2016. "Coverage of ISO/IEC 12207 software lifecycle process by a simulation-based serious game". In: *International Conference on Software Process Improvement and Capability Determination*. Cham, Germany: s.n. 2016. pp. 59–70.

CALDERÓN, A., RUIZ, M., O'CONNOR, R.V., 2017a. "Coverage of ISO/IEC 29110 project management process of basic profile by a serious game". In: *European Conference on Software Process Improvement*. Cham, Germany: s.n. 2017. pp. 111–122.

CALDERÓN, A., RUIZ, M., O'CONNOR, R.V., 2017b. "Coverage of the ISO 21500 Standard in the Context of Software Project Management by a Simulation-Based Serious Game". In: *International Conference on Software Process Improvement and Capability Determination*. Cham, Germany: s.n. 2017. pp. 399–412.

CALDERÓN, A., RUIZ, M., O'CONNOR, R.V., 2018, "A serious game to support the ISO 21500 standard education in the context of software project management". In: *Computer Standards & Interfaces*. pp. 80–92.

CALDERÓN, A., RUIZ, M., ORTA, E., 2017c. "Integrating serious games as learning resources in a software project management course: the case of ProDec". In: *Proceedings of the 1st International Workshop on Software Engineering Curricula for Millennials*. Buenos Aires, Argentina: s.n. 2017. pp. 21–27.

CALDIERA, G., BASILI, V.R., 1991, "Identifying and qualifying reusable software components". In: *Computer*. v. 24, pp. 61–70.

DALPIAZ, F., COOPER, K.M., 2018, "Games for Requirements Engineers: Analysis and Directions". In: *IEEE Software*.

CAMELO, H.N., LUCIO, P.S., JUNIOR, J., 2016, "Modelagem de média mensal de velocidade do vento para região litorânea no nordeste Brasileiro através do método aditivo Holt-Winters com vias a previsão de geração eólica". In: *Revista Brasileira de Energias Renováveis*. v. 5, pp. 587–604.

CARDOW, C.J.E., WATSON, JR., W.D., 1994, "A practical approach to teaching software reuse". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. v. 750 LNCS, pp. 517–525.

CASPERSEN, M.E., CHRISTENSEN, H.B., 2008. "Frameworks in teaching". In: *LNCS*. S.l.: s.n. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 190–205,.

CASTRO, D., WERNER, C., 2019. "Uso de jogos como estratégia de ensino para reutilização de software". In: *SBGames 2019 - Educação* (). Rio de janeiro, Brazil: s.n. Outubro 2019.

CASTRO, D., WERNER, C., FERNANDES, F., 2019. "A tertiary mapping on the use of games for teaching software engineering". In: *SBGames 2019 - Educação*. Rio de janeiro, Brazil: s.n. Outubro 2019. pp. 1120–1123.

CAULFIELD, C., XIA, J.C., VEAL, D., et al., 2011, "A systematic survey of games used for software engineering education". In: *Modern Applied Science*. v. 5, pp. 28– 43,.

CEDDIA, J., TUCKER, S., CLEMENCE, C., et al., 2001. "W.I.E.R - Implementing artifact reuse in an educational environment with real projects-work in progress". In: *Proceedings - Frontiers in Education Conference*. United States, USA: s.n. 2001. pp. 11–16.

CHANDRASHEKAR, H., KIRAN, A.G., UMA, B., et al., 2018, "Introducing Programming using ‘Scratch’ and ‘Greenfoot,’". In: *Journal of Engineering Education Transformations*. pp. 17–27.

CHANG, C.-K., 2014, "Effects of using Alice and Scratch in an introductory programming course for corrective instruction". In: *Journal of Educational Computing Research*. v. 51, pp. 185–204.

CHAVES, R.O., WANGENHEIM, C.G., FURTADO, J.C.C., et al., 2015, "Experimental evaluation of a serious game for teaching software process modeling". In: *IEEE Transactions on Education*. v. 58, pp. 289– 296,.

CHRISTENSEN, C.M., HORN, M.B., STAKER, H., 2013, *Is K-12 Blended Learning Disruptive? An Introduction to the Theory of Hybrids*. . S.l., Clayton Christensen Institute for Disruptive Innovation.

DICHEVA, D., DICHEV, C., AGRE, G., et al., 2015, "Gamification in education: A systematic mapping study". In: *Journal of Educational Technology & Society*. v. 18, pp. 75–88.

DIESEL, A., BALDEZ, A.L.S., MARTINS, S.N., 2017, "Os princípios das metodologias ativas de ensino: uma abordagem teórica". In: *Revista Thema*. v. 14, pp. 268–288,.

DJAOUTI, D., ALVAREZ, J., JESSEL, J.-P., 2011. "Classifying serious games: the G/P/S model". In: *Handbook of research on improving learning and motivation through educational games: Multidisciplinary approaches*. S.l.: IGI Global. pp. 118–136.

DLAMINI, R.S., 2015, "Examining the effectiveness of 2D and 3D online environment in enhancing students' learning of software engineering". In: *International Journal of Innovation and Learning*. v. 17, pp. 425–452,.

DUARTE, E.V., PEARCE, J.L., 2017, "A cross-cultural review of lightbot for introducing functions and code reuse". In: *Journal of Computing Sciences in Colleges*. v. 33, pp. 100–105.

FENTON, N., BIEMAN, J., 2014, *Software metrics: a rigorous and practical approach*. . S.l., CRC press.

FIELD, A., 2013, *Discovering statistics using IBM SPSS statistics*. . S.l., sage.

FRAKES, W., TERRY, C., 1995, "Software reuse and reusability metrics and models". In: . pp. 415–435.

FRAKES, W.B., 1993. "A Graduate Course on Software Reuse, Domain Analysis, and Re-engineering". In: *Proceedings of the Sixth Annual Workshop for Institutionalizing Software Reuse*. Owego, NY: s.n. 1993. pp. 55–65.

FRAKES, W.B., POLE, T.P., 1994, "An empirical study of representation methods for reusable software components". In: *IEEE Transactions on Software Engineering*. v. 20, pp. 617–630.

FUCHS, M., WOLFF, C., 2016. "Improving programming education through gameful, formative feedback". In: *Global Engineering Education Conference (EDUCON), 2016 IEEE*. S.l.: IEEE. 2016. pp. 860–867.

GAFFNEY, J.E., Jr, DUREK, T.A., 1989, "Software reuse—key to enhanced productivity: some quantitative models". In: *Information and Software Technology*. v. 31, pp. 258–267.

- GANNOD, G.C., BURGE, J.E., HELMICK, M.T., 2008. "Using the inverted classroom to teach software engineering". In: *Proceedings of the 30th international conference on Software engineering*. Leipzig, Germany: s.n. 2008. pp. 777–786.
- GARCIA, I., PACHECO, C., LEÓN, A., et al., 2019, "Experiences of using a game for improving learning in software requirements elicitation". In: *Computer Applications in Engineering Education*. pp. 249–265.
- GOLOVIN, D., 2017, "OutSystems as a Rapid Application Development Platform for Mobile and Web Applications". In: *Lahden ammattikorkeakoulu*.
- GONÇALVES, J.J., DE OLIVEIRA, T.M.A., DE OLIVEIRA, K.M., 2006. "Métricas de Reusabilidade para Componentes de Software". In: *VI Workshop de Desenvolvimento Baseado em Componentes (WDBC 2006)*. Campinas, São Paulo, Brazil: s.n. 2006. pp. 14–21.
- GOUWS, L.A., BRADSHAW, K., WENTWORTH, P., 2013. "Computational thinking in educational activities: an evaluation of the educational game light-bot". In: *Proceedings of the 18th ACM conference on Innovation and technology in computer science education*. Canterbury, England, UK: ACM. 2013. pp. 10–15.
- GSCHLIESSER, C., 2019. "The Future Use of LowCode/NoCode Platforms by Knowledge Workers—An Acceptance Study". In: *Knowledge Management in Organizations: 14th International Conference, KMO 2019, Zamora, Spain, July 15–18, 2019, Proceedings*. Zamora, Spain: Springer. 2019. pp. 445–454.
- GUARDIOLA, E., 2016. "The Gameplay Loop: a Player Activity Model for Game Design and Analysis". In: *ACE*. Osaka, Japan: ACM. 2016. pp. 23–30.
- HADJERROUIT, S., 2005, "Learner-centered web-based instruction in software engineering". In: *IEEE Transactions on Education*. v. 48, pp. 99– 104,.
- HALSTEAD, M.H., OTHERS, 1977, *Elements of software science*. . S.l., Elsevier New York.
- HERRANZ, E., COLOMO-PALACIOS, R., AL-BARAKATI, A., 2017. "Deploying a Gamification Framework for Software Process Improvement: Preliminary Results". In: *European Conference on Software Process Improvement*. Ostrava, Czech Republic: s.n. 2017. pp. 231–240.
- HUIZINGA, J., 1949, *Homo ludens: A study of the play element in culture*. . S.l., Taylor & Francis.
- HUNICKE, R., LEBLANC, M., ZUBEK, R., 2004. "MDA: A formal approach to game design and game research". In: *Proceedings of the AAAI Workshop on Challenges in Game AI*. Menlo Park, California: s.n. 2004. pp. 1722.

JÄÄSKELÄINEN, R., 2010, "Think-aloud protocol". In: *Handbook of translation studies*. v. 1, pp. 371–374.

JAIME, M.P., KOLLER, M.R.T., GRAEML, F.R., 2015, *La aplicación de Flipped Classroom en el curso de Dirección Estratégica*. . S.l., s.n.

JESUS, G.M., FERRARI, F.C., DE PAULA PORTO, D., et al., 2018. "Gamification in Software Testing: A Characterization Study". In: *Proceedings of the III Brazilian Symposium on Systematic and Automated Software Testing*. S.l.: ACM. 2018. pp. 39–48.

JIANG, S., ZHANG, H., GAO, C., et al., 2015. "Process simulation for software engineering education". In: *Proceedings of the 2015 International Conference on Software and System Process*. Torino, Italy: s.n. 2015. pp. 147–156.

JIMÉNEZ-HERNÁNDEZ, E.M., OKTABA, H., PIATTINI, M., et al., 2016. "Methodology to construct educational video games in software engineering". In: *Software Engineering Research and Innovation (CONISOFT), 2016 4th International Conference in*. Puebla, Mexico: s.n. 2016. pp. 110–114.

JORDINE, T., LIANG, Y., IHLER, E., 2014. "A mobile-device based serious gaming approach for teaching and learning Java programming". In: *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*. Madrid, Spain: IEEE. 2014. pp. 1–5.

KALELIOGLU, F., GÜLBAHAR, Y., 2014, "The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners' Perspective". In: *Informatics in Education*. v. 13, pp. 33– 50,.

KANG, K.C., COHEN, S.G., HESS, J.A., et al., 1990. *Feature-oriented domain analysis (FODA) feasibility study*. S.l. Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst.

KEELE, S., 2007. *Guidelines for performing systematic literature reviews in software engineering*. Technical report, Ver. 2.3 EBSE Technical Report. S.l. EBSE.

KITCHENHAM, B., 2004, "Procedures for performing systematic reviews". In: *Keele, UK, Keele University*. v. 33, pp. 1–26.

KOSA, M., YILMAZ, M., O'CONNOR, R., et al., 2016, "Software engineering education and games: a systematic literature review". In: *Journal of Universal Computer Science*. v. 22, pp. 1558–1574.

KRUEGER, C.W., 1992, "Software reuse". In: *ACM Computing Surveys*. v. 24, pp. 131– 183,.

LACERDA, D.P., DRESCH, A., PROENÇA, A., et al., 2013, "Design Science Research: método de pesquisa para a engenharia de produção". In: *Gestão & produção*. v. 20, pp. 741–761.

- LARENAS, F., MARÍN, B., GIACHETTI, G., 2018. "Classutopia: A Serious Game for Conceptual Modeling Design". In: *The 30th International Conference on Software Engineering and Knowledge Engineering*. San Francisco, California: s.n. 2018. pp. 116–154,.
- LASKOWSKI, M., 2015. "Implementing gamification techniques into university study path-A case study". In: *Global Engineering Education Conference (EDUCON), 2015*. Tallinn, Estonia: IEEE. 2015. pp. 582–586.
- LATTANZI, M.R., HENRY, S.M., 1996, "Teaching the Object-Oriented Paradigm and Software Reuse: Notes from an Empirical Study". In: *Computer Science Education*. v. 7, pp. 99– 10,.
- LAW, R., 2018. "Introducing Novice Programmers to Functions and Recursion Using Computer Games". In: *European Conference on Games Based Learning*. Antibes, France: Academic Conferences International Limited. 2018. pp. 325–334.
- LEE, E., SHAN, V., BETH, B., et al., 2014. "A structured approach to teaching recursion using cargo-bot". In: *Proceedings of the tenth annual conference on International computing education research*. Glasgow, Scotland, United Kingdom: ACM. 2014. pp. 59–66.
- LEE, W.L., 2016, *SCRUM-X: An interactive and experiential learning platform for teaching scrum*. . S.I., Elsevier.
- LEGAULT, L., GREEN-DEMERS, I., PELLETIER, L., 2006, "Why do high school students lack motivation in the classroom? Toward an understanding of academic amotivation and the role of social support.". In: *Journal of educational psychology*. v. 98, pp. 567–582.
- LEWIS, M., JACOBSON, J., 2002, "Game engines". In: *Communications of the ACM*. v. 45, pp. 27–37.
- LINO, J.E.N., PALUDO, M.A., BINDER, F.V., et al., 2015. "Project management game 2D (PMG-2D): A serious game to assist software project managers training". In: *Frontiers in Education Conference (FIE), 2015 IEEE*. S.I.: IEEE. 2015. pp. 1–8.
- LISBOA, L.B., NASCIMENTO, L.M., DE ALMEIDA, E.S., et al., 2008. "A case study in software product lines: An educational experience". In: *Software Engineering Education Conference, Proceedings*. S.I.: s.n. 2008. pp. 155–162.
- LIU, X., RAJ, R., REICHLMAYR, T., et al., 2013. "Incorporating service-oriented programming techniques into undergraduate CS and SE curricula". In: *Proceedings - Frontiers in Education Conference*. Oklahoma City, Oklahoma,: s.n. 2013. pp. 1369–1371,.

LOPES, F.J.R., PETRILLO, F., 2016. "SimKan: Training Kanban Practices Through Stochastic Simulation". In: *Brazilian Workshop on Agile Methods*. S.l.: Springer. 2016. pp. 110–121.

LOPES, M., BRANCO, V.T.F.C., SOARES, J.B., 2013, "Utilização dos testes estatísticos de Kolmogorov-Smirnov e Shapiro-Wilk para verificação da normalidade para materiais de pavimentação". In: *Transportes*. v. 21, pp. 59–66.

LOPEZ, A.M., VIDAL DUARTE, E., CASTRO GUTIERREZ, E., et al., 2016. "Teaching Abstraction, Function and Reuse in the first class of CS1: A Lightbot Experience". In: *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*. Arequipa, Peru: ACM. 2016. pp. 256–257.

MACKAY, D., 2017, *The fantasy role-playing game: A new performing art*. . S.l., McFarland.

MALONEY, J., RESNICK, M., RUSK, N., et al., 2010, "The scratch programming language and environment". In: *ACM Transactions on Computing Education (TOCE)*. v. 10, pp. 16–31.

MARIMUTHU, M., GOVENDER, P., 2018, "Perceptions of Scratch programming among secondary school students in KwaZulu-Natal, South Africa". In: *African Journal of Information and Communication*. v. 21, pp. 51–80.

MATSUBARA, P.G.F., DA SILVA, C.L.C., 2017. "Game elements in a software engineering study group: a case study". In: *Proceedings of the 39th International Conference on Software Engineering: Software Engineering and Education Track*. S.l.: IEEE Press. 2017. pp. 160–169.

MAURICIO, R. DE A., VEADO, L., MOREIRA, R.T., et al., 2017, "A systematic mapping study on game-related methods for software engineering education". In: *Information and Software Technology*.

MAXIM, B.R., KAUR, R., APZYNSKI, C., et al., 2016. "An agile software engineering process improvement game". In: *Frontiers in Education Conference (FIE), 2016 IEEE*. Erie, PA, USA, USA: s.n. 2016. pp. 1–4.

MCCABE, T.J., 1976, "A complexity measure". In: *IEEE Transactions on software Engineering*. pp. 308–320.

MCCAUGHERTY, D., 1991, "The theory-practice gap in nurse education: its causes and possible solutions. Findings from an action research study". In: *Journal of advanced nursing*. v. 16, pp. 1055– 1061,.

PETRI, G., VON WANGENHEIM, C.G., BORGATTO, A.F., 2016, "MEEGA+ A model for evaluating educational games". In: . Disponível em: <<http://www.gqs.ufsc.br/meega-a-model-for-evaluating-educational-games/>>. Acessado em: 2 Agosto 2019.

- MIŁOSZ, M., MIŁOSZ, E., 2017. "Developing and implementation of decision-making games for business education of engineering students". In: *Global Engineering Education Conference (EDUCON), 2017 IEEE*. Athens, Greece: s.n. 2017. pp. 10–16.
- MORALES-TRUJILLO, M.E., GARCÍA-MIRELES, G.A., MASLOVA, P., 2018. "What Can Go Wrong in a Software Project? Have Fun Solving it". In: *2018 Federated Conference on Computer Science and Information Systems (FedCSIS)*. Poznań, Poland: s.n. 2018. pp. 995–1004.
- MORSCHHEUSER, B., HASSAN, L., WERDER, K., et al., 2018, "How to design gamification? A method for engineering gamified software". In: *Information and Software Technology*. v. 95, pp. 219–237.
- NASSAL, A., 2014. "A general framework for software project management simulation games". In: *Information Systems and Technologies (CISTI), 2014 9th Iberian Conference on*. NW Washington, DC United States: s.n. 2014. pp. 1–5.
- NAVARRO, E.O., BAKER, A., HOEK, A.V.D., 2004. "Teaching software engineering using simulation games". In: *ICSIE'04: Proceedings of the 2004 International Conference on Simulation in Education*. San Diego, CA,: s.n. 2004.
- NIU, N., 2011, *Reuse a "software Reuse" Course*. . S.I., ASEE Annual Conference and Exposition, Conference Proceedings.
- PELLAS, N., 2014, "The development of a virtual learning platform for teaching concurrent programming languages in the Secondary Education: The use of Open Sim and Scratch4OS". In: *Journal of e-Learning and Knowledge Society*. v. 10.
- PELLAS, N., VOSINAKIS, S., 2017. "Learning to think and practice computationally via a 3D simulation game". In: *Interactive Mobile Communication, Technologies and Learning*. S.I.: Springer. 2017. pp. 550–562.
- PERROTTA, C., FEATHERSTONE, G., ASTON, H., et al., 2013, *Game-based learning: Latest evidence and future directions*. . Slough, NFER.
- PETRI, G., VON WANGENHEIM, C.G., BORGATTO, A.F., 2016, "MEEGA+: an evolution of a model for the evaluation of educational games". In: *INCoD/GQS*. v. 3.
- PETTICREW, M., ROBERTS, H., 2009, *systematic reviews in the social sciences: A practical guide*. . Blackwell, Malden, John Wiley & Sons.
- PIEPER, J., LUETH, O., GOEDICKE, M., et al., 2017. "A case study of software engineering methods education supported by digital game-based learning: Applying the SEMAT Essence kernel in games and course projects". In: *Global Engineering Education Conference (EDUCON), 2017 IEEE*. Athens, Greece: s.n. 2017. pp. 1689–1699.

- PIVEC, M., DZIABENKO, O., SCHINNERL, I., 2003. "Aspects of game-based learning". In: *3rd International Conference on Knowledge Management*. Graz, Austria: s.n. 2003. pp. 216–225.
- POTTER, H., SCHOTS, M., DUBOC, L., et al., 2014. "InspectorX: A game for software inspection training and learning". In: *Software Engineering Education and Training (CSEE&T), 2014 IEEE 27th Conference on*. S.l.: IEEE. 2014. pp. 55–64.
- QU, W.-Q., ZHAO, Y.-F., WANG, M., et al., 2014. "Research on teaching gamification of software engineering". In: *Computer Science & Education (ICCSE), 2014 9th International Conference on*. S.l.: IEEE. 2014. pp. 855–860.
- RAMAKRISHNAN, S., MINGINS, C., HENDERSON-SELLERS, B., et al., 1994. "Planned software reuse in object-oriented software engineering education". In: *Software Education Conference, 1994*. Dunedin, New Zealand: Proceedings. 1994. pp. 250–254.
- RAMÍREZ-ROSALES, S., VÁZQUEZ-REYES, S., VILLA-CISNEROS, J.L., et al., 2016. "A serious game to promote object oriented programming and software engineering basic concepts learning". In: *Software Engineering Research and Innovation (CONISOFT), 2016 4th International Conference in*. Puebla, Puebla, Mexico: s.n. 2016. pp. 97–103.
- REDOLFI, G., DE ARAUJO SPAGNOLI, L., BASTOS, R.M., et al., 2004, "Especificando Informações para Componentes Reutilizáveis". In: *Porto Alegre, Abril*.
- RITTERFELD, U., CODY, M., VORDERER, P., 2009, *Serious games: Mechanisms and effects*. . 1. New York, Routledge.
- ROSA, M., 2008, *A Construção de identidades online por meio do Role Playing Game: relações com o ensino e aprendizagem de matemática em curso à distância*. . 1. S.l., Bolema: Boletim de Educação Matemática.
- ROUSE, M., 2018. Disponível em: <<https://searchsoftwarequality.techtarget.com/definition/low-code-no-code-development-platform>>. Acessado em: 5 Janeiro 2020.
- SÁEZ-LÓPEZ, J.-M., ROMÁN-GONZÁLEZ, M., VÁZQUEZ-CANO, E., 2016, "Visual programming languages integrated across the curriculum in elementary school: A two year case study using “Scratch” in five schools". In: *Computers & Education*. v. 97, pp. 129–141.
- SALEN, K., TEKINBAŞ, K.S., ZIMMERMAN, E., 2004, *Rules of play: Game design fundamentals*. . S.l., MIT press.
- SAMETINGER, J., 1997, *Software engineering with reusable components*. . S.l., Springer Science & Business Media.

SCHMITZ, E.X., 2016, *Sala de Aula Invertida: uma abordagem para combinar metodologias ativas e engajar alunos no processo de ensino-aprendizagem*. . S.l., Universidade Federal de Santa Maria.

SERRANO-LAGUNA, Á., TORRENTE, J., IGLESIAS, B.M., et al., 2015, "Building a Scalable Game Engine to Teach Computer Science Languages". In: *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*. v. 10, pp. 253–261.

SHABALINA, O., SADOVNIKOVA, N., KRAVETS, A., 2013. "Methodology of teaching software engineering: Game-based learning cycle". In: *Engineering of Computer Based Systems (ECBS-EERC), 2013 3rd Eastern European Regional Conference on the*. S.l.: IEEE. 2013. pp. 113–119.

SIERRA, A., ARIZA, T., FERNÁNDEZ-JIMÉNEZ, F., et al., 2016. "Educational resource based on games for the reinforcement of engineering learning programming in mobile devices". In: *2016 Technologies Applied to Electronics Teaching (TAE)*. S.l.: IEEE. 2016. pp. 1–6.

SINDRE, G., KARLSSON, E.-A., STÅLHANE, T., 1992. "Software reuse in an educational perspective". In: *Software Engineering Education*. Berlin, Heidelberg: Springer Berlin Heidelberg. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics. pp. 99–114,.

SOUZA, M.R., CONSTANTINO, K.F., VEADO, L.F., et al., 2017a. "Gamification in software engineering education: An empirical study". In: *Software Engineering Education and Training (CSEE&T), 2017 IEEE 30th Conference on*. Savannah, GA, USA Cite this publication: s.n. 2017. pp. 276–284.

SOUZA, M.R., VEADO, L., MOREIRA, R.T., et al., 2017b. "Games for learning: bridging game-related education methods to software engineering knowledge areas". In: *Proceedings of the 39th International Conference on Software Engineering*. Buenos Aires Argentina: Software Engineering and Education Track. 2017. pp. 170–179.

STEINMACHER, I., CHAVES, A.P., GEROSA, M.A., 2013, "Awareness support in distributed software development: A systematic review and mapping of the literature". In: *Computer Supported Cooperative Work (CSCW)*. v. 22, pp. 113–158,.

SU, C.-H., 2016, "The effects of students' motivation, cognitive load and learning anxiety in gamification software engineering education: a structural equation modeling study". In: *Multimedia Tools and Applications*. v. 75, pp. 10013–10036,.

SUESCÚN, E., 2018, "SimulES-W: A Collaborative Game to Improve Software Engineering Teaching". In: *Computación y Sistemas*. v. 22.

- SUESCÚN, E., CAMAPIO DO PRADO LEITE, J.C., WERNECK, V., et al., 2018, "SimulES-W: A Collaborative Game to Improve Software Engineering Teaching". In: *Computación y Sistemas*. v. 22.
- SZABO, C., 2014. "Evaluating GameDevTycoon for teaching software engineering". In: *Proceedings of the 45th ACM technical symposium on Computer science education*. Atlanta Georgia, USA: s.n. 2014. pp. 403–408.
- TABET, N., GEDAWY, H., ALSHIKHABOBAKR, H., et al., 2016. "From alice to python. Introducing text-based programming in middle schools". In: *Proceedings of the 2016 ACM Conference on innovation and Technology in Computer Science Education*. Arequipa, Peru: ACM. 2016. pp. 124–129.
- TELUKUNTA, S., KOTA, M.S.K., POTTI, M.S., et al., 2014, *StrateJect: An Interactive Game for Project Management Experiential Learning*. . S.l., PMP Conference, PMI Bangalore Chapter. PMPC.
- TEWARI, R., 1995, "Software Reuse and Object-Oriented Software Engineering in the Undergraduate Curriculum". In: *ACM SIGCSE Bulletin*. v. 27, pp. 253– 257,.
- TEWARI, R., FRIEDMAN, F.L., 1993, "A Framework for Incorporating Object-Oriented Software Engineering in the Undergraduate Curriculum". In: *Computer Science Education*. v. 4, pp. 45– 62,.
- THOMAS, C., BERKLING, K., 2013. "Redesign of a gamified software engineering course". In: *Interactive Collaborative Learning (ICL), 2013 International Conference on*. S.l.: IEEE. 2013. pp. 778–786.
- TOPALLI, D., CAGILTAY, N.E., 2018, "Improving programming skills in engineering education through problem-based game projects with Scratch". In: *Computers & Education*. v. 120, pp. 64–74.
- UNKELOS-SHPIGEL, N., 2016. "Ace That Game: Educating Students to Gamified Design Thinking.". In: *STPIS@ CAiSE*. S.l.: s.n. 2016. pp. 75–82.
- USKOV, V., SEKAR, B., 2014. "Gamification of software engineering curriculum". In: *Frontiers in Education Conference (FIE), 2014 IEEE*. Madrid, Spain: s.n. 2014. pp. 1–8.
- UYAGUARI, F.U., INTRIAGO, M., JÁCOME, E.S., 2015. "Gamification Proposal for a Software Engineering Risk Management Course". In: *New Contributions in Information Systems and Technologies*. S.l.: Springer. pp. 795–802.
- VACA-CÁRDENAS, L.A., 2015. "Coding with Scratch: The design of an educational setting for Elementary pre-service teachers". In: *2015 International Conference on Interactive Collaborative Learning (ICL)*. Florence, Italy: s.n. 2015. pp. 1171–1177.

- VACA-CÁRDENAS, L.A., BERTACCHINI, F., TAVERNISE, A., et al., 2015. "Coding with Scratch: The design of an educational setting for Elementary pre-service teachers". In: *2015 International Conference on Interactive Collaborative Learning (ICL)*. Florence, Italy: IEEE. 2015. pp. 1171–1177.
- VALENCIA, D., VIZCAÍNO, A., GARCIA-MUNDO, L., et al., 2016. "GSDgame: A serious game for the acquisition of the competencies needed in GSD". In: *Global Software Engineering Workshops (ICGSEW), 2016 IEEE 11th International Conference on*. Orange County, CA, USA: s.n. 2016. pp. 19–24.
- VASCONCELOS, L.E.G., OLIVEIRA, L.B., GUIMARÃES, G., et al., 2018. "Gamification Applied in the Teaching of Agile Scrum Methodology". In: *Information Technology-New Generations*. S.l.: Springer. pp. 207–212.
- VINCUR, J., KONOPKA, M., TVAROZEK, J., et al., 2017. "Cubely: virtual reality block-based programming environment". In: *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*. Gothenburg, Sweden: ACM. 2017. pp. 84–88.
- WANG, J., WANG, Y.-A., 2000. "Teaching software reuse with JavaBeans". In: *Proceedings - Frontiers in Education Conference*. Kansas city, Missouri: s.n. 2000. pp. 2–8.
- WOHLIN, C., 2014. "Guidelines for snowballing in systematic literature studies and a replication in software engineering". In: *Proceedings of the 18th international conference on evaluation and assessment in software engineering*. London, England United Kingdom: s.n. 2014. pp. 38–48.
- XEXÉO, G., CARMO, A., ACIOLI, A., et al., 2017, *O Que Sao Jogos*. . S.l., Rio de Janeiro. Disponível em:<<https://ludes.cos.ufrj.br/wp-content/uploads/2016/07/LJP1C01-O-que-sao-jogos-v2.pdf>>.
- XIE, F., BROWNE, J.C., 2006. "Verification of component-based software application families". In: *International Symposium on Component-Based Software Engineering*. Västerås, Sweden: Springer. 2006. pp. 50–66.
- YAMASHITA, S., TSUNODA, M., YOKOGAWA, T., 2017. "Visual Programming Language for Model Checkers Based on Google Blockly". In: *International Conference on Product-Focused Software Process Improvement*. Wolfsburg, Germany: Springer. 2017. pp. 597–601.
- YAO, L., RABHI, F.A., PEAT, M., 2014. "Supporting data-intensive analysis processes: a review of enabling technologies and trends". In: *Handbook of Research on Architectural Trends in Service-Driven Computing*. S.l.: IGI Global. pp. 481–508.

Apêndice A (Questionário MEEGA)

Questionário de caracterização

Por favor, responda as questões abaixo com base na experiência que obteve ao utilizar o jogo. Todos os dados coletados são apenas para melhoria da pesquisa e serão publicadas de forma totalmente anônima, não comprometendo o participante.

Nome:

Formação acadêmica:

- Doutorado concluído
- Doutorado em andamento
- Mestrado em andamento
- Mestrado concluído
- Graduação concluída
- Graduação em andamento

Faixa etária:

- Menos de 18 anos
- 18 a 28 anos
- 29 a 39 anos
- 40 a 50 anos
- Mais de 50 anos

Com que frequência você costuma jogar jogos digitais? (Considere 1 o menor grau e 5 o maior)

1	2	3	4	5
---	---	---	---	---

Com sua experiência com reutilização de software? (Considere 1 o menor grau e 5 o maior)

1	2	3	4	5
---	---	---	---	---

Questionário de avaliação

Usabilidade					
	Discordo totalmente	Discordo	Nem discordo, nem concordo	Concordo	Concordo totalmente
O <i>design</i> do jogo é atraente (tabuleiro, cartas, interfaces, gráficos, etc.).	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Os textos, cores e fontes combinam e são consistentes.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eu precisei aprender poucas coisas para poder começar a jogar o jogo.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Aprender a jogar este jogo foi fácil para mim.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eu acho que a maioria das pessoas aprenderiam a jogar este jogo rapidamente.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eu considero que o jogo é fácil de jogar.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
As regras do jogo são claras e compreensíveis.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
As fontes (tamanho e estilo) utilizadas no jogo são legíveis.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
As cores utilizadas no jogo são compreensíveis.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Experiência do Jogador					
	Discordo totalmente	Discordo	Nem discordo, nem concordo	Concordo	Concordo totalmente
A organização do conteúdo me ajudou a estar confiante de que eu iria aprender com este jogo.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Este jogo é adequadamente desafiador para mim.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
O jogo oferece novos desafios (oferece novos obstáculos, situações ou	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

variações) com um ritmo adequado.					
O jogo não se torna monótono nas suas tarefas (repetitivo ou com tarefas chatas).	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Completar as tarefas do jogo me deu um sentimento de realização.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
É devido ao meu esforço pessoal que eu consigo avançar no jogo.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Me sinto satisfeito com as coisas que aprendi no jogo.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eu recomendaria este jogo para meus colegas.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eu me diverti com o jogo.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Houve algo interessante no início do jogo que capturou minha atenção.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eu estava tão envolvido no jogo que eu perdi a noção do tempo.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eu esqueci sobre o ambiente ao meu redor enquanto jogava este jogo.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
O conteúdo do jogo é relevante para os meus interesses.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
É claro para mim como o conteúdo do jogo está relacionado com a disciplina.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eu prefiro aprender com este jogo do que de outra forma (outro método de ensino).	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
O jogo contribuiu para a minha aprendizagem na disciplina.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
O jogo foi eficiente para minha aprendizagem, em comparação com outras atividades da disciplina.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

De acordo com sua opinião, foi identificado algum aspecto positivo / negativo da utilização do jogo? Se sim, qual(ais)?

Você possui alguma sugestão para melhoria do jogo? Em caso positivo, por favor, especifique-a.

Este espaço é reservado para quaisquer comentários adicionais (dificuldades, críticas e/ou sugestões) a respeito do estudo executado. Contamos com sua contribuição para que o trabalho seja aprimorado.

Obrigado pela sua colaboração!

Pesquisador responsável:

Diego Cardoso Borda Castro

Professor pesquisador responsável:

Cláudia Maria Lima Werner

Apêndice B (Questionário Educacional)

Dentro de sala de aula, você já utilizou jogos para o ensino? Comente sobre a utilização.

No seu ponto de vista, descreva as vantagens e desvantagens de se utilizar jogos em sala de aula.

No seu ponto de vista, qual a maior dificuldade de ensino de Reutilização de Software?

Você considera que a utilização do jogo Reuse Blocks em sala de aula pode favorecer o aprendizado?
