

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**EXPLORANDO GAMIFICAÇÃO NO ENSINO DE
TESTE DE SOFTWARE**

GABRIELA MARTINS DE JESUS

ORIENTADOR: PROF. DR. FABIANO CUTIGI FERRARI

São Carlos – SP
Dezembro/2019

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

EXPLORANDO GAMIFICAÇÃO NO ENSINO DE TESTE DE SOFTWARE

GABRIELA MARTINS DE JESUS

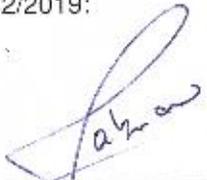
Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Engenharia de software
Orientador: Prof. Dr. Fabiano Cutigi Ferrari

São Carlos – SP
Dezembro/2019

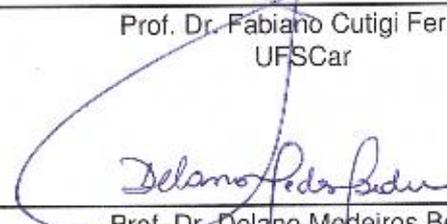


Folha de Aprovação

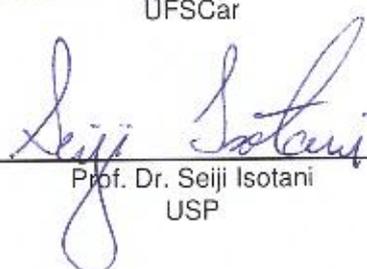
Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Dissertação de Mestrado da candidata Gabriela Martins de Jesus, realizada em 20/12/2019:



Prof. Dr. Fabiano Cutigi Ferrari
UFSCar



Prof. Dr. Delano Medeiros Beder
UFSCar



Prof. Dr. Seiji Isotani
USP

Dedico aos meus.

AGRADECIMENTO

Primeiramente a Deus, meu Tudo, e eternamente à minha família por estar sempre ao meu lado com todo amor, carinho, compreensão e suporte que eu poderia ter.

À minha amiga Sarinha, irmã que escolhi, “my person”, obrigada pela presença e apoio nos dias claros e nublados, e pela amizade tão valiosa.

Aos meus amigos de perto, do cotidiano, da Tokenlab, especialmente ao grupo "Bora se almoçar 3a idade": (Sarinha “my person”, Aninha (que mulher!), Nair (estrupício top), e Rô (maravilhosa). A vocês que sempre convidam para uma pausa no trabalho, seja para um almoço no shopping, *baconzitos*, espeto, ou sorvete, meu muito obrigada por tornar os dias mais gor(dos)tosos e agradáveis <3 .

Aos meus amigos que mesmo distantes fisicamente torcem por mim.

Ao meu orientador pelo apoio, parceria, conselhos, compreensão, paciência e confiança. Agradeço por ter compartilhado a experiência e me ajudado a concluir este mestrado com bons frutos.

Aos colegas do Laboratório de Pesquisa em Engenharia de Software (LAPES) do Departamento de Computação da Universidade Federal de São Carlos (DC-UFSCar) pela colaboração e inúmeras discussões produtivas.

Aos co-autores dos trabalhos publicados como frutos dessa pesquisa, agradeço pela dedicação, colaboração e parceria.

Às quatro instituições de ensino superior que possibilitaram a realização do experimento, aos professores que cederam suas aulas, e aos alunos que livremente participaram.

Ao Ricardo, meu gestor na Tokenlab, por sua compreensão pelos momentos em que precisei dividir o foco do trabalho com o mestrado. Agradeço também aos colegas mais próximos de trabalho que me acompanharam (e aguentaram) nessa reta final.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo financiamento deste projeto.

A todos que direta ou indiretamente contribuíram para a realização (e finalização) deste trabalho.

"Só por mais 5 minutos"

Gabriela Martins de Jesus

RESUMO

Contexto: Teste é uma atividade fundamental no processo de desenvolvimento e tem por objetivo revelar defeitos. Apesar de sua importância, o tempo dedicado ao ensino de teste não tem sido suficiente, resultando na falta de profissionais capacitados. Outros problemas enfrentados são o desalinhamento entre o que se ensina e o que se aplica na indústria, ineficiência de aulas teóricas, aulas pouco atrativas, e falta de atividades práticas. Para resolvê-los, abordagens de ensino alternativas à convencional vêm sendo investigadas, tais como o uso de jogos sérios, *Flipped Classroom*, e gamificação. Gamificação, por sua vez, é uma tendência promissora que usa elementos de jogos em contextos do mundo real a fim de aumentar a motivação e o engajamento das pessoas em tarefas que requerem estímulos externos, especialmente em contextos educacionais.

Objetivos: Contribuir com o ensino de teste de software investigando se o uso da gamificação neste contexto aumenta a motivação e o desempenho dos alunos quando comparado com a abordagem convencional. **Método:** Realizou-se um mapeamento sistemático da literatura caracterizando como a gamificação tem sido explorada no contexto de teste de software. Além disso, identificaram-se alguns dos problemas no ensino de teste por meio de uma busca *ad hoc*. Em seguida, desenvolveu-se uma abordagem gamificada e implementou-se uma plataforma inserindo a abordagem criada. Sessões experimentais foram realizadas, e os resultados coletados em diversos formulários foram analisados. **Resultados:** Em geral, não é possível afirmar que a motivação e o desempenho dos alunos que aprenderam teste de software por meio da abordagem gamificada é maior do que a dos que aprenderam por meio da abordagem convencional, tendo em vista que para cada grupo experimental o resultado foi diferente nas questões avaliadas quando comparado com o grupo de controle. No entanto, considera-se como positiva a experiência de uso desta abordagem alternativa, pois a mesma proporcionou um ambiente mais agradável e divertido, tanto do ponto de vista da pesquisadora quanto dos alunos.

Palavras-chave: Gamificação, teste de software, teste funcional, particionamento de equivalência, análise de valor limite, educação

ABSTRACT

Context: Testing is a fundamental activity in the development process, and aims at revealing faults. Despite its importance, the time devoted to testing education has not been enough, resulting in the lack of trained professionals. Other problems faced are misalignment between what is taught and what is applied in the industry, inefficiency of lectures, unattractive classes, and lack of practical activities. Aiming at solving them, alternative educational approaches have been investigated, such as serious games, flipped classroom, and gamification. Gamification, in turn, is a promising trend that uses game elements in real-world contexts in order to increase people's motivation and engagement in tasks that require external stimuli, especially in educational contexts. **Objectives:** Contributing to software testing education by investigating whether the use of gamification in this context increases students' motivation and performance when compared to the conventional approach. **Method:** A systematic literature mapping was performed aiming at characterizing how gamification has been explored in the software testing context. In addition, some of the problems faced by testing education were identified through an *ad-hoc* search. Then, a gamified approach was developed, and a platform was implemented containing the created approach. Experimental sessions were run, and the collected results were analysed. **Results:** In general, it is not possible to state that the motivation and performance of students who learned through the gamified approach is higher than those who learned through the conventional approach, given that, for each experimental group, the results were different when compared with the control group. However, the experience of having used this alternative approach is considered positive, as it provided a more enjoyable and funny environment, both from the researcher's and students' point of view.

Keywords: Gamification, software testing, functional testing, equivalence partitioning, boundary value analysis, education

LISTA DE FIGURAS

2.1	Gráfico de interesse dos jogadores	23
2.2	Separação de gamificação e outros conceitos.	25
2.3	Hierarquia dos elementos de jogos	26
2.4	Elementos de jogos utilizados no aplicativo Duolingo	28
3.1	Etapas seguidas durante a realização do Mapeamento Sistemático	33
3.2	Número de estudos que relacionam elementos de gamificação a objetivos da gamificação.	50
4.1	Tabela de definição de atividades e implementação	61
4.2	Planilha de definição do ambiente gamificado	62
4.3	Ambiente gamificado - Aba <i>Admin</i>	63
4.4	Ambiente gamificado - Aba <i>AlunoX</i>	64
4.5	Ambiente gamificado - Aba <i>Avatar</i>	65
4.6	Ambiente gamificado - Aba <i>Loja</i>	66
4.7	Ambiente gamificado - Aba <i>Fórum</i>	66
4.8	Abordagem Metodológica	70
5.1	Resultados da Motivação - GC vs GE1	78
5.2	Resultado dos testes de conhecimento - GC vs GE1	79
5.3	Boxplot com os valores Δ s dos testes de conhecimento - GC vs GE1	79
5.4	Resultado do teste de normalidade da distribuição dos dados no pré- e pós-teste - GC e GE1	80
5.5	Teste de Mann-Whitney para o pré- e pós-teste - GC vs. GE1	80
5.6	Eficácia dos alunos no <i>quiz 1</i> - GC, GE1, GE2	81
5.7	Resultado do teste de normalidade da distribuição dos dados no Quiz 1 - GC e GE1	82
5.8	Teste de Mann-Whitney para o Quiz 1 - GC vs. GE1	82
5.9	Eficácia dos alunos no <i>quiz 2</i> - GC, GE1, GE2	83
5.10	Resultado do teste de normalidade da distribuição dos dados no Quiz 2 - GC e GE1	84
5.11	Teste de Mann-Whitney para o Quiz 2 - GC vs. GE1	84
5.12	Resultados da Motivação - GC vs GE2	85
5.13	Resultado dos testes de conhecimento - GC vs GE2	86
5.14	Boxplot com os valores Δ s dos testes de conhecimento - GC vs GE2	87
5.15	Resultado do teste de normalidade da distribuição dos dados no pré- e pós-teste - GC e GE2	87
5.16	Teste de Mann-Whitney para o pré- e pós-teste - GC vs. GE2	88
5.17	Resultado do teste de normalidade da distribuição dos dados no Quiz 1 - GC e GE2	89
5.18	Teste de Mann-Whitney para o Quiz 1 - GC vs. GE2	89
5.19	Resultado do teste de normalidade da distribuição dos dados no Quiz 2 - GC e GE2	90

5.20	Teste de Mann-Whitney para o Quiz 2 - GC vs. GE2	90
5.21	Resultados da Motivação - GC vs (GE1 + GE2)	91
5.22	Resultado dos testes de conhecimento - GC vs (GE1 + GE2)	92
5.23	Boxplot com os valores Δ s dos testes de conhecimento - GC vs (GE1 + GE2)	93
5.24	Resultado do teste de normalidade da distribuição dos dados no pré- e pós-teste - GC e (GE1 + GE2)	93
5.25	Teste de Mann-Whitney para o pré- e pós-teste - GC vs (GE1 + GE2)	94
5.26	Eficácia dos alunos no <i>quiz</i> 1 - GC vs (GE1 + GE2)	94
5.27	Resultado do teste de normalidade da distribuição dos dados no Quiz 1 - GC e (GE1 + GE2)	95
5.28	Teste de Mann-Whitney para o Quiz 1 - GC vs (GE1 + GE2)	95
5.29	Eficácia dos alunos no <i>quiz</i> 2 - GC vs (GE1 + GE2)	96
5.30	Resultado do teste de normalidade da distribuição dos dados no Quiz 2 - GC e (GE1 + GE2)	96
5.31	Teste de Mann-Whitney para o Quiz 2 - GC vs. (GE1 + GE2)	97

LISTA DE TABELAS

1.1	Quantidade de estudos relacionados a gamificação em teste de software por ano.	3
3.1	Número de estudos em relação aos métodos de pesquisa	36
3.2	Estudos Incluídos e Inclusores.	37
3.3	Estudos selecionados através do Mapeamento Sistemático.	38
3.4	Resumo dos conceitos principais dos elementos de jogos.	39
3.5	Resumo dos conceitos principais sobre os objetivos da gamificação.	39
3.6	Resumo dos conceitos principais dos contextos de aplicação.	40
3.7	Resumo dos conceitos principais sobre as técnicas de teste.	40
3.8	Resumo dos conceitos principais sobre os níveis de teste.	40
3.9	Resumo dos conceitos principais sobre as fases do processo de teste.	41
3.10	Resultado da classificação dos estudos na categoria <i>Elementos de Jogos</i>	42
3.11	Resultado da classificação dos estudos na categoria <i>Objetivos do uso da Gamificação</i>	43
3.12	Resultado da classificação dos estudos na categoria <i>Contextos de Aplicação de Teste</i>	43
3.13	Resultado da classificação dos estudos na categoria <i>Técnicas de Teste</i>	44
3.14	Resultado da classificação dos estudos na categoria <i>Níveis de Teste</i>	44
3.15	Resultado da classificação dos estudos na categoria <i>Fases de Teste</i>	44
4.1	Problemas no Ensino de Teste de Software	57
4.2	Atividades gamificadas para motivação dos comportamentos desejáveis	59
4.3	Conquistas e suas recompensas	62
4.4	Dados Demográficos.	68
5.1	Questões de Pesquisa e Hipóteses	74
5.2	Resultados do feedback dos alunos	97

LISTA DE ABREVIATURAS E SIGLAS

- ES - Engenharia de Software
- GFC - Grafo de Fluxo de Controle
- GOAL - *Gamification on Application Lifecycle Management*
- HALO - *Highly Addictive, socialLy Optimized*
- IDE - *Integrated Development Environment*
- IMI - *Intrinsic Motivation Inventory*
- MMORPG - *(Massively Multiplayer Online Role Playing Game*
- MS - Mapeamento Sistemático
- PBL - Pontos, Selos, Quadro de classificação
(do inglês *Points, Badges, Leaderboards*)
- QA - *Quality Assurance*
- SDT - *Selfdetermination Theory*
- WReSTT-CyLE - *Web-Based Repository of Software Testing Tutorials
a- Cyberlearning Environment*

SUMÁRIO

CAPÍTULO 1 INTRODUÇÃO	1
1.1 Contexto e Motivação	1
1.2 Problemática	4
1.3 Objetivos	5
1.4 Organização	5
CAPÍTULO 2 FUNDAMENTAÇÃO TEÓRICA	7
2.1 Considerações Iniciais	7
2.2 Teste de Software	7
2.2.1 Técnicas e Critérios	10
2.2.1.1 Teste Funcional	10
2.2.1.2 Teste Estrutural	11
2.2.1.3 Teste Baseado em Defeitos	12
2.2.2 Níveis de Teste e Estratégias de Teste	13
2.2.3 Fases do Processo de Teste	15
2.2.4 Ensino de Teste de Software	17
2.3 Gamificação	20
2.3.1 Conceitos sobre Jogos	21
2.3.2 Conceitos sobre Gamificação	24
2.3.3 Elementos de Jogos	25
2.3.4 Exemplo de Aplicação da Gamificação	27
2.3.5 A Importância da Motivação	29
2.4 Considerações Finais	30
CAPÍTULO 3 REVISÃO DA LITERATURA	32
3.1 Considerações Iniciais	32
3.2 Processo e Planejamento da Pesquisa	33
3.2.1 Protocolo do Mapeamento Sistemático	33
3.3 Condução da Pesquisa e Resultados das Buscas	36
3.4 Extração dos Dados e Visão Geral dos Estudos Seleccionados	38
3.5 Análise e Discussão das Classificações	42
3.5.1 Classificação: <i>Contexto de aplicação</i>	42
3.5.2 Classificação: <i>Elementos de jogos utilizados</i>	45
3.5.3 Classificação: <i>Objetivos da gamificação</i>	46
3.5.4 Classificação: <i>Técnicas de teste de software</i>	46
3.5.5 Classificação: <i>Níveis de teste de software</i>	47
3.5.6 Classificação: <i>Fases do processo de teste de software</i>	47
3.6 Discussões Adicionais	48
3.7 Revisitando a Questão de Pesquisa	49
3.8 Limitações e Trabalhos Relacionados	50
3.8.1 Limitações	51
3.8.2 Trabalhos Relacionados	51
3.9 Considerações Finais	52

CAPÍTULO 4 PROCEDIMENTOS E MÉTODOS	54
4.1 Considerações Iniciais	54
4.2 Referencial Metodológico	54
4.3 Procedimentos	55
4.3.1 Revisão da Literatura	55
4.3.2 Desenvolvimento da Abordagem Gamificada	57
4.3.3 Desenvolvimento da Plataforma Gamificada	63
4.3.4 Material Experimental	65
4.3.5 Experimento	68
4.4 Considerações Finais	71
CAPÍTULO 5 ANÁLISE DOS DADOS	72
5.1 Considerações Iniciais	72
5.2 Estudo Piloto	72
5.2.1 Lições Aprendidas	73
5.3 Experimento	74
5.3.1 Design e Procedimento	75
5.3.2 Análise Quantitativa – Observações Gerais	75
5.3.3 Análise Quantitativa 1 - (GC vs GE1)	77
5.3.4 Análise Quantitativa 2 - (GC vs GE2)	85
5.3.5 Análise Quantitativa 3 - (GC vs (GE1 + GE2))	91
5.3.6 Percepção dos Alunos	95
5.3.7 Análise Qualitativa	98
5.4 Revisitando as Questões de Pesquisa	100
5.5 Ameaças à Validade	102
5.6 Considerações Finais	104
CAPÍTULO 6 CONCLUSÃO	105
6.1 Contribuições	106
6.2 Limitações e Trabalhos Futuros	107
6.3 Publicações	108
REFERÊNCIAS	110
APÊNDICE A VISÃO GERAL DOS ESTUDOS SELECIONADOS	117
APÊNDICE B INTRINSIC MOTIVATION INVENTORY - IMI (INGLÊS)	126
APÊNDICE C INTRINSIC MOTIVATION INVENTORY - IMI (PORTUGUÊS)	128

Capítulo 1

INTRODUÇÃO

1.1 Contexto e Motivação

Atualmente os sistemas de software têm controlado o funcionamento de grande parte das atividades cotidianas, seja nos negócios, educação, saúde, lazer, ou vida doméstica (Graham et al., 2008; Myers et al., 2011; Spillner et al., 2014). Jones e Bonsignour (2011), por exemplo, afirmaram que 65% da população norte americana utilizava software incorporado em diversos tipos de dispositivos, e boa parte os utilizavam diariamente. No entanto, muitos desses sistemas funcionam em ambientes considerados críticos, podendo, inclusive, colocar em risco a vida de pessoas ou a sobrevivência financeira de empresas. Dessa forma, garantir a qualidade e aumentar a confiabilidade dos produtos de software não é apenas uma opção, mas uma necessidade.

Ao longo dos anos, diversos avanços na área de desenvolvimento de software levaram à criação de novas tecnologias, modelos e técnicas para alcançar a qualidade de software desejada, mas apesar do progresso feito até hoje, desenvolver software confiável permanece um desafio. Neste cenário, as atividades de teste de software desempenham um papel fundamental em contribuir com a melhoria da qualidade que tem sido, inclusive, um fator essencial de sucesso para as organizações (Pressman, 2010; Spillner et al., 2014) .

A atividade de teste se concretiza em um conjunto de técnicas, critérios, níveis, e fases de um processo que, em geral, consiste em executar o software (em qualquer modelo abstrato executável) utilizando dados de entrada específicos com a intenção de revelar seus defeitos (Myers et al., 2011). Dessa forma, verifica-se se o comportamento observado está em conformidade com o comportamento esperado. No entanto, mesmo que os testes possam revelar defeitos existentes, não se pode provar a ausência de defeitos sem uma execução exaustiva do software, com todos os possíveis dados de entrada (Myers et al., 2011).

Embora o teste exaustivo seja impraticável, o teste sistemático pode contribuir em aumentar a confiança de que o software funcione conforme o esperado (Harrold, 2000). Porém, há alguns fatores que tornam as atividades de teste tediosas e desmotivadoras. Por exemplo, o teste é um fator de alto consumo de tempo e esforço dentro do processo de desenvolvimento de software, podendo representar mais de 40% do orçamento do projeto, conforme anualmente reportado por uma pesquisa global com gerentes de Tecnologia da Informação (Capgemini Group et al., 2016). Além disso, a abordagem convencional¹ utilizada no ensino de teste de software não colabora em atrair a atenção dos alunos que, conseqüentemente, não aprendem e não se interessam pela área. Destaca-se também que o tempo dedicado ao ensino de seus conceitos nem sempre é suficiente nos cursos de computação resultando, assim, na falta de profissionais de teste capacitados (Bell et al., 2011; Fraser, 2017; Rojas e Fraser, 2016a; Yujian Fu e Clarke, 2016). Como consequência, a indústria carece de profissionais e os projetos produzem software de baixa qualidade, ou fracassam excedendo o tempo e/ou custo planejados (Dubois e Tamburrelli, 2013).

O fracasso em projetos de software também pode ocorrer devido a fatores humanos presentes ao longo do processo de desenvolvimento de software. Estes fatores humanos são os recursos principais dentro deste processo (García et al., 2017), e sua desmotivação traz riscos para o sucesso de um projeto, uma vez que a qualidade do produto de software é impactada também pela motivação e comprometimento (Dal Sasso et al., 2017; Koscianski e dos Santos Soares, 2007). Dessa forma, motivação e disciplina têm se tornado elementos cruciais para um bom desenvolvimento de software (Dubois e Tamburrelli, 2013), e uma forma de introduzir e manter estes dois elementos é utilizando a *gamificação*.

Gamificação é um assunto que tem sido amplamente discutido em outras áreas além da Engenharia de Software (Hamari et al., 2014), e se define pelo uso de elementos de jogos em contextos do mundo real (Deterding et al., 2011). Essa estratégia é indicada em contextos cujas tarefas são consideradas (por alguns) desmotivadoras por serem repetitivas, entediantes, longas, ou complexas, necessitando assim de estímulos para gerar uma mudança de comportamento, ou aumentar a motivação e o engajamento das pessoas para realizá-las (Werbach e Hunter, 2012). Dessa forma, tarefas desagradáveis à equipe de desenvolvimento, tais como a escrita de testes de unidade, podem ser estimuladas com recompensas e diversão através da gamificação. Em um contexto educacional, a competitividade inserida por meio de elementos de jogos como, por exemplo, *quadro de classificação*, pode instigar e motivar os alunos a permanecerem atentos e se aplicarem com maior esforço na resolução de problemas.

¹Refere-se à *abordagem convencional* como o modelo de ensino no qual o professor explica os conceitos e os alunos assistem passivamente.

Werbach e Hunter (2012) apresentaram o exemplo de uma iniciativa da companhia *Volkswagen* em uma cidade na Suécia. A empresa simulou um piano nas escadas do metrô onde, à medida que as pessoas tocavam os degraus para subir ou descer, os sons eram emitidos. Os autores destacaram que, mesmo as pessoas sabendo da importância da prática de exercícios físicos, os evitam em situações simples preferindo o uso da escada rolante. Porém, adicionando a diversão à prática de subir/descer uma escada, perceberam um aumento de 66% no uso das escadas tradicionais. Ou seja, às vezes o que as pessoas precisam para realizarem suas atividades é de um estímulo motivacional. Dessa forma, a gamificação tem sido usada como forma de inserir a diversão para servir a objetivos do mundo real (Werbach e Hunter, 2012) em diversas áreas do conhecimento nos últimos anos, inclusive no ensino teste de software.

Dado este contexto, um Mapeamento Sistemático (MS) da literatura (apresentado no Capítulo 3) foi conduzido com o objetivo de caracterizar como a gamificação tem sido utilizada para apoiar as atividades de teste de software. A caracterização foi realizada de acordo com seis perspectivas, sendo elas: contexto de aplicação da gamificação, elementos de jogos utilizados, objetivos da gamificação, técnicas de teste, níveis de teste, e fases do processo de teste abordados. Com as estratégias de busca automática e *backward snowballing*, 817 estudos foram encontrados, mas apenas 15 estudos primários foram selecionados para compor a base de discussão deste trabalho (Jesus et al., 2018).

De acordo com Deterding et al. (2011), o uso da gamificação se tornou amplamente adotado no final de 2010. No entanto, especificamente para a área de teste de software, somente em 2016 um número maior de trabalhos foi publicado, conforme identificado no MS e mostrado na Tabela 1.1.

Tabela 1.1: Quantidade de estudos relacionados a gamificação em teste de software por ano.

Ano	#	Estudos
2017	6	(Clegg et al., 2017; Dal Sasso et al., 2017; García et al., 2017; Laurent et al., 2017; Liechti et al., 2017; Rojas et al., 2017)
2016	4	(Parizi, 2016; Rojas e Fraser, 2016a,b; Yujian Fu e Clarke, 2016)
2015	1	(Anderson et al., 2015)
2014	0	
2013	1	(Dubois e Tamburrelli, 2013)
2012	1	(Sheth et al., 2012)
2011	2	(Bell et al., 2011; Passos et al., 2011)

Fonte: Dados da pesquisa

Percebeu-se nestes 15 trabalhos selecionados que as abordagens de gamificação são, quase todas, aplicáveis tanto a contextos educacionais quanto industriais. Não houve, com exceção dos trabalhos voltados ao teste baseado em defeitos, um foco específico em gamificar determinada técnica, nível, ou fase do processo de teste. Além

disso, os elementos básicos de gamificação conhecidos como a tríade PBL (do inglês *Points, Badges e Leader board*) estão ainda entre os mais utilizados. Outro ponto importante a se destacar é que todos os trabalhos selecionados que realizaram experimentos apresentaram resultados positivos, atraindo assim a atenção e o interesse dos pesquisadores em utilizar a gamificação como estratégia para motivação e desempenho de alunos em seu processo de aprendizagem.

Os desafios presentes no contexto educacional (por exemplo, atrair a atenção dos alunos) e os benefícios motivacionais e comportamentais que a gamificação tem apresentado motivaram este trabalho de mestrado, que focou na criação de uma abordagem gamificada contendo um conjunto mais completo de elementos de jogos visando proporcionar resultados mais profundos (Werbach e Hunter, 2012), como uma maior motivação e desempenho dos alunos no contexto de ensino de teste de software.

1.2 Problemática

Diante do exposto anteriormente, apesar do profundo impacto da qualidade do software, a educação no contexto de teste de software está atrasada: não há ênfase suficiente no ensino das habilidades de teste necessárias na indústria de software. A maior parte de um currículo típico de graduação em Ciência da Computação é dedicada às atividades de desenvolvimento (The Joint Task Force on Computing Curricula, 2015), e o teste de software é frequentemente negligenciado em favor das atividades de *design* e implementação. Para piorar a situação, um equívoco comum entre estudantes e profissionais é pensar que o teste de software é algo desinteressante em comparação com outras atividades de caráter construtivo.

Além do tempo dedicado ao ensino de teste não ser suficiente, resultando na falta de profissionais capacitados, há outros problemas como o desalinhamento entre o que se ensina e o que se aplica na indústria, ineficiência de aulas teóricas, aulas pouco atrativas, e falta de atividades práticas. Tais problemas podem gerar como consequência o desinteresse, desatenção e desmotivação dos alunos.

Visando mitigar tais problemas, abordagens de ensino alternativas à convencional vêm sendo investigadas, tais como o uso de jogos sérios (Rojas e Fraser, 2016b), *Flipped Classroom* (Paschoal et al., 2017), e a gamificação (Yujian Fu e Clarke, 2016). Porém, estas abordagens emergentes ainda têm sido pouco discutidas e investigadas (Smith et al., 2012).

Frente aos desafios e problemas enfrentados no ensino de teste de software, os benefícios apresentados nos estudos analisados em um mapeamento sistemático (Jesus et al., 2018) inspiraram este trabalho de mestrado onde investigou-se se seria possível

atrair mais a atenção dos alunos com o uso da gamificação no ensino de teste de software de forma a aumentar sua motivação e desempenho quando comparado com a abordagem convencional. Dessa forma, com base no que fora apresentado até o momento, a partir dessa questão geral definiram-se duas subquestões de pesquisa, sendo a primeira relacionada à motivação, e a segunda ao desempenho dos alunos, conforme apresentado a seguir:

Subquestão de Pesquisa 1: A motivação dos alunos que aprendem teste de software por meio de uma abordagem gamificada é maior do que a motivação dos alunos que aprendem por meio da abordagem convencional?

Subquestão de Pesquisa 2: O desempenho dos alunos que aprendem teste de software por meio de uma abordagem gamificada é maior do que o desempenho dos alunos que aprendem por meio da abordagem convencional?

1.3 Objetivos

Este trabalho teve como objetivo definir uma abordagem de ensino gamificada como alternativa à convencional, visando mitigar os problemas identificados no contexto de educacional de teste de software, a fim de aumentar a motivação e desempenho dos alunos.

Os objetivos específicos deste trabalho foram:

1. Desenvolver uma abordagem gamificada, definindo o conjunto de elementos de jogos a serem utilizados, o sistema de recompensas, conquistas e pontos;
2. Implementar o ambiente gamificado contendo a abordagem desenvolvida;
3. Criar os materiais experimentais empregados nas sessões experimentais, sendo eles a base para a extração e a análise dos dados;
4. Planejar e executar as sessões experimentais a fim de responder as questões de pesquisa deste trabalho; e
5. Analisar e discutir os resultados obtidos, e propor melhorias necessárias.

1.4 Organização

Este trabalho de mestrado está organizado em seis capítulos, sendo este primeiro uma introdução contendo a contextualização e motivação, problemática e objetivos

deste trabalho. No Capítulo 2 são apresentados os fundamentos de teste de software e gamificação, seguido do Capítulo 3, no qual se apresenta a revisão da literatura com os trabalhos relacionados ao uso da gamificação no contexto de teste de software. Em seguida, no Capítulo 4, são apresentados os procedimentos e métodos utilizados para o desenvolvimento deste trabalho. No Capítulo 5 apresentam-se as análises dos dados extraídos das atividades experimentais e a discussão dos resultados. Por fim, no Capítulo 6 conclui-se apresentando as descobertas, contribuições, limitações, trabalhos futuros e publicações realizadas ao longo deste trabalho de mestrado.

Capítulo 2

FUNDAMENTAÇÃO TEÓRICA

2.1 Considerações Iniciais

São discutidos neste capítulo os conceitos principais sobre *teste de software* e *gamificação*, a fim de proporcionar uma melhor compreensão deste trabalho no decorrer da leitura. Em relação a teste de software, na Seção 2.2 são apresentados os conceitos básicos e na subseção 2.2.1 abordam-se as técnicas de teste e seus critérios. Os diferentes níveis e estratégias de teste são apresentados na subseção 2.2.2, seguidos das fases do processo de teste na subseção 2.2.3. O ensino de teste de software, por fim, é abordado na subseção 2.2.4.

Em seguida, na Seção 2.3, são abordados os fundamentos relacionados a gamificação, como os conceitos de jogos na subseção 2.3.1, e os conceitos de gamificação na subseção 2.3.2. Em seguida, os elementos de jogos são apresentados na subseção 2.3.3, seguido de exemplos de aplicação da gamificação na subseção 2.3.4. Por fim, alguns conceitos sobre motivação serão brevemente apresentados na subseção 2.3.5. Encerra-se o capítulo com as considerações finais na Seção 2.4.

2.2 Teste de Software

O teste de software é um processo “destrutivo” que tem por objetivo revelar defeitos e desempenhando, assim, um papel crítico em aumentar a confiabilidade e prover evidência da qualidade dos sistemas (Myers et al., 2011). As atividades de teste devem ocorrer durante todo o ciclo de desenvolvimento, iniciando o quanto antes a

fim de revelar os defeitos desde as menores unidades do software, facilitando-se a depuração (Graham et al., 2008).

Teste de software é considerado uma das técnicas de *verificação* e *validação*, as quais visam garantir que o software foi implementado corretamente, e se o software correto foi implementado, respectivamente (Pressman, 2010). As atividades de teste pertencem ao grupo das técnicas de validação. Em geral, o teste possui um comportamento dinâmico (Spillner et al., 2014) e consiste na execução do software em qualquer modelo abstrato executável com o uso de dados de entrada específicos; após a execução, verifica-se se o comportamento observado está em conformidade com o comportamento esperado. O software deve funcionar apenas conforme está especificado. Assim, com as atividades de teste, verifica-se tanto se o software faz o que deve ser feito, quanto se ele não faz o que não deve ser feito (Myers et al., 2011).

Alguns termos como *defeito*, *erro*, *falha*, e *engano* podem ser utilizados de formas distintas pelas pessoas, mas para uma melhor compreensão durante a leitura deste trabalho, serão utilizadas as definições segundo o *Institute of Electrical and Electronics Engineers (IEEE)*. De acordo com o IEEE (1990), engano (do inglês *mistake*) é uma ação humana incorreta. Um engano produz um defeito (do inglês *fault*), que é uma imperfeição (por exemplo, passos, processos, definições de dados incorretos) em algum artefato de software. Erro (do inglês *error*) é um estado¹ inconsistente do programa, e pode ser gerado por um defeito no software. O erro pode causar uma falha (do inglês *failure*), que é a percepção desta inconsistência de forma externa à estrutura do software, ou seja, observa-se que o resultado obtido não está de acordo com o resultado esperado (dependendo da criticidade, uma falha pode causar a parada do funcionamento do sistema). Além destes, é comum no jargão dos desenvolvedores e testadores utilizar o termo *bug* que, segundo, Spillner et al. (2014), pode ser usado como sinônimo de *defeito*. Portanto, um engano produz um defeito que pode gerar um erro que, por sua vez, pode produzir uma falha.

Destaca-se que defeito é considerado estático, pois não depende da execução do programa para existir. Por outro lado, erro e falha são dinâmicos, ou seja, não ocorrem nem são perceptíveis sem que o programa (ou modelo) seja executado; além disso, outros fatores além de defeitos podem causar uma falha, como por exemplo, condições do ambiente, ou erros humanos durante o uso do sistema (Graham et al., 2008).

Uma falha ocorre dependendo de valores de entrada específicos, ou seja, nem todo valor de entrada gerará uma falha quando o programa defeituoso for executado (Koscianski e dos Santos Soares, 2007). É possível que um artefato de software contenha defeitos (imperfeições) que nunca serão revelados (Graham et al., 2008);

¹Valor das variáveis do programa na memória e do apontador de instruções em um determinado momento de execução do programa (Delamaro et al., 2017)

porém, uma descoberta tardia (com o sistema já em produção²) pode causar impactos consideráveis e até catastróficos. Por isso é de fundamental importância a definição de casos de teste úteis e eficazes para revelar os defeitos, especialmente em sistemas críticos como os utilizados nas áreas da saúde, militar, aeronáutica, agronomia, ou tantos outros que podem colocar a vida de seres humanos ou a sobrevivência financeira de empresas em risco.

O saber escrever bons casos de teste (isto é, com alta probabilidade de revelar defeitos) é um conhecimento essencial para os testadores, além de ser um desafio considerando a variada e complexa tecnologia que abrange as atividades diárias (Myers et al., 2011). Mas o que são casos de teste? De acordo com Carrington (1996), um caso de teste é um par composto por um dado de teste que será utilizado como entrada no programa em teste, e uma saída esperada como resultado do uso deste dado de teste. Para melhor compreensão, conforme apresentado por Delamaro et al. (2017), um programa P possui um domínio de entrada $D(P)$ que contém todos os possíveis valores de entrada para a sua execução. Um *dado de teste* d é um elemento (valor de entrada) específico pertencente a $D(P)$, ou seja, $d \in D(P)$. Dessa forma, um caso de teste é formado pelo par (d, r) , onde r é o resultado esperado da execução de P utilizando d . Os casos de teste projetados para a execução de P são conhecidos como *conjunto de casos de teste* C , ou *conjunto de testes* (Frankl e Weyuker, 2000; Graham et al., 2008; Myers et al., 2011).

Para a criação de um conjunto de casos de teste adequado, é necessário definir quais elementos de software deverão ser executados. Estes elementos de software são chamados *requisitos de teste*, e são definidos baseando-se em alguns *critérios de teste* (apresentados na próxima seção) (Frankl e Weyuker, 2000).

Se a execução de C *passar* (ou seja, a execução ocorreu com sucesso), então entende-se que o resultado obtido está de acordo com o resultado esperado, não revelando defeitos; mas se a execução de C *falhar*, então um defeito foi revelado e o caso de teste que o revelou obteve sucesso em seu objetivo (isto é, revelar defeitos) (Myers et al., 2011).

O responsável por verificar e decidir após a execução de um programa em teste se o resultado obtido está de acordo com o resultado esperado (isto é, se o caso de teste *passou* ou *falhou*) é conhecido como *oráculo*. Esse papel pode ser desempenhado pelo próprio testador ou por alguma ferramenta de teste automatizado. Além disso, especificações do software podem ser usadas como base para apoiar a decisão (Mathur, 2013).

Embora o oráculo de teste desempenhe o papel de decisor, é importante observar que deve ser considerado um limite de tempo e esforço praticável. Além disso, mesmo

²Produção é o ambiente de funcionamento real do usuário onde o sistema é colocado em funcionamento

que o processo de teste possa revelar defeitos existentes, não é possível garantir sua ausência (Graham et al., 2008; Spillner et al., 2014); para isso, deveriam ser criados casos de testes para todas as possíveis combinações de valores do domínio de entrada do programa. Isto é conhecido como *teste exaustivo* e é algo impraticável, mesmo para pequenos programas (Graham et al., 2008; Myers et al., 2011; Spillner et al., 2014).

Além da impossibilidade de praticar o teste exaustivo, é um desafio para gerentes de teste obter os resultados desejados possuindo recursos muitas vezes limitados. Neste cenário, concentrar os esforços nas partes mais críticas do software e priorizar o que deve ser testado é essencial. Dessa forma, estes (e outros) fatores tornam fundamental a definição de um processo em diferentes níveis de teste utilizando técnicas e critérios adequados a cada realidade (Spillner et al., 2014).

2.2.1 Técnicas e Critérios

Embora seja impraticável testar um software exaustivamente, é possível testá-lo sistematicamente delimitando-se finitamente o domínio de entrada D em subdomínios S_n . Para isso, técnicas e critérios de teste são utilizados para gerar casos de teste eficazes (Spillner et al., 2014). As técnicas de teste são estilos de se criar os casos de teste (Carrington, 1996), e diferem entre si pelo tipo de informação utilizada para gerar os subdomínios S_n que, por sua vez, contêm elementos d (valores de entrada) derivados do domínio de entrada D de um programa P . Além disso, cada técnica possui critérios de teste que são regras utilizadas para definir a qual dos S_n determinado d pertence (Delamaro et al., 2017).

2.2.1.1 Teste Funcional

A técnica de *teste funcional*, também conhecida como *teste caixa-preta* (do inglês *black-box testing*), é utilizada para verificar as funcionalidades do programa em teste. Os casos de teste são criados sem que se conheça a estrutura interna do software, baseando-se apenas nas especificações de requisitos (cuja qualidade é fundamental); assim, esta técnica é aplicável em qualquer nível de teste. Os principais critérios funcionais para a criação dos S_n são (Myers et al., 2011):

1. *Particionamento de Equivalência;*
2. *Análise de Valor Limite;*
3. *Grafo de Causa-Efeito;*
4. *Error-Guessing.*

No Particionamento de Equivalência, o $D(P)$ é particionado em S_n que representam as classes de dados equivalentes válidas (isto é, seus valores de entrada são válidos) e inválidas (isto é, seus valores de entrada são inválidos), de onde podem-se gerar os casos de teste escolhendo-se qualquer d pertencente a cada um dos S_n . Segundo Myers et al. (2011), como no Particionamento de Equivalência diversos S_n são gerados para representar as classes de equivalência válidas e inválidas, os casos de teste devem ser criados combinando-se todos os valores das classes de equivalência válidas, assim obtendo-se casos de teste positivos. Já para casos de teste negativos, deve-se combinar um valor de entrada de uma classe de equivalência inválido com valores de entrada das classes de equivalência válidas.

Na Análise de Valor Limite, os S_n do critério anterior são utilizados para a criação de novos S_n com o intuito de revelar os defeitos presentes nas “fronteiras” das classes de equivalência, que são onde, em geral, encontram-se a maioria dos defeitos (Spillner et al., 2014). Diferentemente do critério anterior, na Análise de Valor Limite não são selecionados quaisquer elementos dos S_n . Os $d \in S_n$ serão os valores limite inferiores e superiores (tanto dos dados de entrada quanto das saídas da execução) dos elementos de cada subdomínio das classes de equivalência (Myers et al., 2011).

De acordo com Myers et al. (2011), os dois critérios anteriores não exploram os efeitos que combinações de entradas podem causar no funcionamento do software. Nesse sentido, o *Grafo de Causa-Efeito (GCE)* realiza essas combinações. Outra vantagem do *GCE* é a possibilidade de identificar ambiguidades e incompletudes nas especificações. Os casos de teste neste critério são obtidos a partir de uma tabela de decisão que é derivada de um grafo booleano; o grafo, por sua vez, é originado pela combinação das possíveis condições de entrada (causas) e seus possíveis resultados (efeitos). A criação do grafo, no entanto, pode ser complexa (Myers et al., 2011).

Por fim, o critério *Error Guessing* não possui uma metodologia a ser seguida (é considerado *ad hoc*); em geral, depende-se da experiência e intuição do testador. Os casos de teste são criados baseando-se nas características das funcionalidades e no $D(P)$. Além disso, considera-se uma lista de possíveis situações de falhas que poderiam ser observadas (Myers et al., 2011).

2.2.1.2 Teste Estrutural

A técnica de *Teste Estrutural*, também conhecida como *Teste Caixa-Branca* (do inglês *White-Box Testing*), considera o fluxo do programa em teste. Os casos de teste são criados com base na estrutura interna do software. Alguns dos critérios estruturais para a criação dos S_n de $D(P)$ são (Spillner et al., 2014):

1. *Cobertura de Comandos*
2. *Cobertura de Decisão*;

3. Cobertura de Múltipla-Condição.

Estes critérios são baseados em fluxo de controle e geralmente são representados por um *Grafo de Fluxo de Controle (GFC)* da unidade em teste; o *GFC* possui nós e arestas representando os blocos de código e os possíveis caminhos lógicos, respectivamente.

De acordo com Spillner et al. (2014), o critério Cobertura de Comandos define que todos os comandos do programa devem ser executados pelo menos uma vez por pelo menos um caso de teste. No critério Cobertura de Decisão, todas as decisões baseadas nos valores lógicos *verdadeiro* e *falso* devem ser executadas; ou seja, cada declaração de decisão (if-else, do-while, etc.), tanto para resultado lógico *verdadeiro* quanto para *falso*, deve ser coberta (executada) por pelo menos um *caso de teste*. O critério *Cobertura de Múltipla-Condição*³ exige que todas as combinações de valor lógico *verdadeiro* e *falso* de uma condição devem ser executados pelo menos uma vez (Spillner et al., 2014).

Em relação ao cálculo de cobertura obtido com testes estruturais, tomando-se como exemplo o critério Cobertura de Comandos, tem-se que o requisito de teste será que todos os comandos do programa em teste devem ser executados por pelo menos um caso de teste. Se o conjunto de casos de teste gerado através deste critério executar todos os comandos, então a cobertura de teste será de 100%, e portanto afirma-se que o conjunto de casos de teste é C-adequado. Ressalta-se que para outros critérios, tanto estruturais como associados a outras técnicas de teste, a cobertura é analogamente calculada com base nos requisitos de teste cobertos pelos testes.

2.2.1.3 Teste Baseado em Defeitos

A técnica de *Teste Baseado em Defeitos* é promissora em garantir a qualidade dos testes e, conseqüentemente, do software (Offutt e Pan, 1997). Assim como os desenvolvedores podem cometer enganos gerando defeitos no software, também os testadores podem criar casos de teste ineficientes. Para essa técnica, são dois os principais critérios utilizados que visam verificar a qualidade dos casos de teste criados: *Semeadura de Defeitos*, e *Análise de Mutantes*.

No critério *Semeadura de Defeitos*, defeitos propositais são “semeados” no software em teste; após a semeadura, os casos de teste são executados e, posteriormente, sua eficácia na detecção de defeitos (tanto semeados quanto reais) é analisada (Budd, 1980). Este critério também pode ser utilizado para auxiliar a estimativa do número total de defeitos que devem ser revelados como critério de término dos testes (Myers et al., 2011).

³Por exemplo: a decisão *if (x > 1) AND (y < 5)* possui duas condições: $x > 1$, e $y < 5$

A Análise de Mutantes, por sua vez, baseia-se em criar versões do programa original realizando pequenas mudanças sintáticas no código; essas novas versões são chamadas *mutantes* (DeMillo et al., 1978); assim como no critério anterior, a Análise de Mutantes também verifica a capacidade dos casos de teste em “matar” os mutantes (ou seja, revelar as modificações sintáticas no programa). Embora este critério seja de grande importância, ainda não é largamente empregado na indústria por algumas desvantagens como um número elevado de mutantes criados e, talvez a principal, a criação de mutantes equivalentes que impactam na efetividade e eficiência dos testes (Rojas e Fraser, 2016a).

As técnicas e critérios de teste apresentados podem (e devem) ser utilizados, na medida do possível, de forma complementar uma à outra. Autores como Myers et al. (2011) e Spillner et al. (2014) aconselham este uso com o propósito de revelar defeitos que uma ou outra técnica não seja capaz de detectar.

2.2.2 Níveis de Teste e Estratégias de Teste

O teste de software é um processo desafiador, e um dos desafios que os gerentes de teste enfrentam é trabalhar com um número reduzido de recursos para a realização das atividades. Mas precisam fazê-las. Porém, somente com um processo bem definido utilizando técnicas e critérios apropriados em diferentes níveis de teste será possível alcançar resultados desejáveis (Spillner et al., 2014).

Postergar o início dos testes para quando o sistema já estiver todo pronto não é uma prática aconselhável. O ideal é que o processo de teste verifique o software em diferentes *níveis*⁴, desde suas partes menores até o seu todo. Tais níveis de teste são conhecidos como *unidade*, *integração*, e *sistema*.

No teste em nível de unidade testa-se o software em suas menores partes (unidades) podendo ser, dependendo da linguagem de programação utilizada, um método, função, procedimento, classe, ou mesmo um componente (conjunto de classes). O teste de unidade visa verificar o fluxo de controle das unidades de forma independente, revelando defeitos inseridos na lógica de programação, estruturas de dados ou declaração de tipos de dados incorretos. Os testes realizados neste nível podem requerer, além da especificação relacionada à unidade em teste, o conhecimento interno da estrutura do software. Assim, estes testes são, normalmente, realizados pelos próprios desenvolvedores (Spillner et al., 2014), e o critério funcional de análise de valor limite é (geralmente) o mais importante para ser utilizado (Pressman, 2010). Destaca-se também a importância de se verificar os caminhos de tratamento de exceções. Uma

⁴Também chamados de *fases*. Neste trabalho será utilizado o termo *nível* de teste a fim de evitar conflito com o termo *fase*, que é utilizado na conceituação de processo de teste, apresentada na próxima seção.

das vantagens do teste de unidade é proporcionar a depuração de forma mais rápida e nos estágios iniciais do software. Além disso, os casos de teste são normalmente automatizados utilizando *frameworks* como o JUnit⁵.

No teste em nível de integração busca-se revelar defeitos associados à comunicação entre as unidades. Sugere-se iniciar a integração pelas unidades mais críticas a fim de revelar seus defeitos o quanto antes (Graham et al., 2008). Todas as técnicas de teste podem ser empregadas. No entanto, dependendo do tamanho do software, recomenda-se o uso da técnica de teste funcional, pois basear-se em sua estrutura interna pode ser inviável (Myers et al., 2011).

Graham et al. (2008), Myers et al. (2011) e Spillner et al. (2014) apresentam duas estratégias de teste: *não-incremental* (“*big-bang*”) e *incremental*. No teste não-incremental, todas as unidades são integradas e testadas como um todo; no teste incremental, por sua vez, integra-se às unidades já testadas a próxima a ser testada. Os autores também apresentam duas ordens para se aplicar o teste de integração: *integração descendente* (*top-down*) e *integração ascendente* (*bottom-up*). No entanto, para o desenvolvimento de software orientado a objetos, outras estratégias de integração são utilizadas, como o *teste baseado em uso*, e o *teste baseado em sequência de execução* (Pressman, 2010). Considerando os prós e os contras das estratégias não-incremental e incremental, Myers et al. (2011) concluem que a abordagem incremental é superior em vantagens à não-incremental.

Por fim, no teste em nível de sistema, busca-se testar o comportamento do software como um todo, verificando se todas as unidades foram integradas corretamente; são considerados tanto requisitos funcionais quanto não funcionais (Graham et al., 2008). Neste nível, busca-se validar se o software cumpre os objetivos⁶ esperados (Myers et al., 2011), ou seja, se o sistema, já todo integrado, está de acordo com suas especificações. De acordo com Graham et al. (2008), o teste de sistema se baseiam em especificações de alto nível do comportamento do sistema (por exemplo, especificações de requisitos, casos de uso, manual do usuário, dentre outros). Além disso, neste nível, os testes devem ser realizados por um profissional capacitado e que seja independente da equipe de desenvolvimento; a execução deve ocorrer em um ambiente de teste controlado e similar ao de produção a fim de evitar que falhas específicas de ambiente deixem de ser observadas. Por fim, técnicas funcional e estrutural podem ser utilizadas nos testes em *nível de sistema* (Graham et al., 2008).

⁵Disponível em <<https://junit.org/>> - Acessado em 28 de novembro de 2019.

⁶Objetivos: “*o quê*” o programa deve fazer, e *quão bem* isto deve ser feito (tradução própria).

2.2.3 Fases do Processo de Teste

Um mesmo processo de teste pode ser seguido em cada um dos três níveis de teste apresentados, empenhando-se menor ou maior esforço de acordo com a criticidade do sistema. Neste trabalho, seis fases do processo de teste serão abordadas, sendo elas: *planejamento de teste, configuração de dados e ambiente, projeto de casos de teste, execução e avaliação, monitoramento e controle, manutenção.*

A fase de planejamento de testes envolve, desde o início do projeto, a criação de um plano de teste, e este deve se manter “vivo” (isto é, verificável e atualizável) durante o todo o ciclo de desenvolvimento do software. De acordo com Spillner et al. (2014), a principal questão durante o planejamento é definir quais e como alcançar os objetivos de cada fase dos testes. O plano de teste elaborado deve conter informações como os objetivos do teste, o escopo, os recursos necessários (incluindo tempo estimado para execução dos testes, ferramentas, servidores, recursos humanos, etc.), papéis e suas responsabilidades, quais técnicas, critérios, níveis, e fases do processo de teste serão seguidas, como eles acontecerão, e quais serão os critérios de término do teste (Graham et al., 2008; Myers et al., 2011).

Além da criação do plano de teste, é necessário identificar e priorizar as funcionalidades com maiores riscos de causar danos mais críticos caso alguma falha no software ocorra (Graham et al., 2008). Esta priorização é importante devido tanto à impossibilidade da realização do teste exaustivo, quanto à limitação dos recursos voltados às atividades de teste.

Observa-se também que, como os recursos do projeto são limitados, o planejamento deve considerar que o custo do teste e da correção de um defeito não deve ser maior do que o prejuízo que a ocorrência de uma falha causaria (isso também justifica porque nem todos os defeitos revelados são corrigidos) (Spillner et al., 2014).

Na fase de configuração de dados e ambiente, deve-se implementar e configurar o os dados e ambiente no qual a execução dos casos de teste ocorrerá (por exemplo, banco de dados, servidores, *links* dos ambientes de teste e de produção, dentre outros). Graham et al. (2008) consideram esta fase do processo de teste como sendo uma tarefa da fase de execução dos testes.

A fase projeto de casos de teste envolve a criação dos casos de teste utilizando as técnicas e critérios de teste especificadas no plano de teste. Porém, anteriormente à sua criação, deve-se considerar a revisão das especificações que servirão como base, pois defeitos são incorporados no código também por compreensão errônea da especificação de um requisito. Assim, deve-se verificar se não há ambiguidade, se a especificação é completa, e se é claramente compreensível. Outro ponto relevante a ser considerado é a importância de se manter a rastreabilidade entre os casos de teste e os requisitos verificados por eles. Quanto maior for a criticidade de um sistema, maior

deve ser o esforço empreendido nas atividades de teste, e maior deve ser a eficácia dos casos de teste em revelarem os defeitos na execução do software (Spillner et al., 2014). Isto significa que essa fase é crucial para a obtenção de resultados satisfatórios com os testes realizados. Se defeitos não forem revelados com a execução dos casos de teste criados, então ou estes não foram eficazes, ou o software foi muito bem implementado; geralmente deve-se investigar a primeira opção.

A fase execução e avaliação envolve executar os casos de teste criados e avaliar os resultados das execuções. A execução dos testes pode ser realizada tanto de forma manual, quanto automatizada (por meio da utilização de ferramentas de teste automatizado) (Graham et al., 2008). É nesta fase que o oráculo de teste desempenha seu papel avaliando se os resultados obtidos estão de acordo com o esperado. Os resultados das execuções devem ser registrados, especialmente as falhas obtidas com a execução dos casos de teste, para posterior análise, localização e correção do defeito. No caso do teste em nível de unidade, as falhas não são necessariamente registradas, pois os testes são realizados pelo próprio desenvolvedor, sendo que os defeitos são corrigidos assim que identificados (Graham et al., 2008). Após os defeitos identificados serem corrigidos, nova execução e avaliação deve ser realizada, até que o conjunto de casos de teste utilizado não revele novos defeitos. Graham et al. (2008) observam que, embora haja diferentes níveis de independência entre desenvolvedor e testador, o mais importante será o quão bem a pessoa saberá testar o software. No entanto, os autores também apresentam motivos para considerar a distinção destes papéis.

A fase de monitoramento e controle envolve um conjunto de atividades cuja finalidade é monitorar e controlar o progresso das atividades de teste com base na comparação do que foi definido no plano de testes. Estas atividades envolvem a utilização de métricas como forma de medir os resultados dos testes realizados (por exemplo, medir a densidade de defeitos encontrados por caso de teste) e do desempenho dos testadores, monitorar o progresso dos testes provendo *feedback* aos envolvidos no projeto (porcentagem da cobertura de teste, número de defeitos revelados e corrigidos, etc.), controlar o progresso definindo ações corretivas, e tomar decisões quando riscos ou mudanças colocarem em risco o sucesso do projeto (Graham et al., 2008). Monitorando-se o progresso das atividades, qualquer identificação de necessidade de alteração no plano de teste pode (e deve) ser realizada.

A última fase do processo de teste é a manutenção do conjunto de casos de teste. Esta fase envolve a alteração ou criação de novos casos de teste com o objetivo de cobrir as mudanças realizadas no software. Estas mudanças podem ser provenientes de correções dos defeitos revelados, alterações nos requisitos, e desenvolvimento de novas versões do sistema.

Segundo Carrington (1996), espera-se que novos defeitos não sejam incorporados durante a fase de manutenção do software. No entanto, isso não é o que acontece

sempre. Novos defeitos podem ser inseridos no software como um “efeito colateral” das mudanças realizadas. O *teste de regressão*, especialmente realizado na fase de manutenção do software, tem a finalidade de revelar estes novos defeitos. No teste de regressão, um conjunto de casos de teste é re-executado; os casos de teste que *passaram* em execuções anteriores devem novamente *passar* na re-execução. Se algum caso de teste falhar, então afirma-se que um defeito foi inserido. Como este teste possui características de reutilizar os casos de teste, ser repetitivo, e (às vezes) longo, é aconselhável que seja executada a sua automatização. Além disso, ele pode ser realizado tanto nos níveis de *teste de unidade*, *integração*, e *sistema* (Spillner et al., 2014).

O processo de teste nos três *níveis* apresentados (unidade, integração, sistema) pode ser realizado pelos próprios desenvolvedores ou por equipes independentes do desenvolvimento (equipes conhecidas como *Garantia da Qualidade*, ou simplesmente *QA*, do inglês, *Quality Assurance*). O teste indicado para ser realizado pelos próprios desenvolvedores é o teste de unidade, pois dependem fortemente do conhecimento da estrutura interna do sistema; o teste de integração, por sua vez, pode ser realizado tanto pelos desenvolvedores quanto pelos QAs; por fim, o teste de sistema deve ser realizado por uma equipe experiente e independente do desenvolvimento. Embora um programador seja também um testador (mesmo que não siga técnicas, critérios, fases do processo de teste), é aconselhável que indivíduos distintos testem o que foi implementado. Segundo Carrington (1996), testar seu próprio código gera um conflito de interesse. Dessa forma, um teste realizado pelo próprio programador poderia visar provar que o programa está correto, o que é contrário ao real objetivo das atividades de teste. Como afirmaram Myers et al. (2011), teste é uma atividade destrutiva. Assim, dificilmente o criador de uma funcionalidade tentaria destruí-la (Spillner et al., 2014).

Até o momento foram apresentados neste capítulo conceitos sobre teste de software, incluindo técnicas, níveis, e fases do processo de teste. Considerando a importância destas atividades, serão discutidos a seguir o cenário atual do ensino neste contexto.

2.2.4 Ensino de Teste de Software

O crescimento significativo da indústria de software e o papel fundamental das atividades de garantia da qualidade têm demandado profissionais mais capacitados. A engenharia de software (ES), por exemplo, é uma das áreas que têm evoluído e colaborado com o crescimento tecnológico da indústria. Esta evolução, por sua vez, reflete em conhecimentos abrangentes e abstratos que são complexos de aprender exigindo-se, assim, mais tempo dedicado ao ensino desses assuntos e mais atividades práticas para melhor fixação do conteúdo.

Apesar da importância da qualidade do software, diversos autores como Benitti (2017); Cheiran et al. (2017); Pinto e Silva (2017); Smith et al. (2012); Valle (2016) discutiram em seus trabalhos sobre o *deficit* de profissionais capacitados na área. Uma possível causa para esse deficit pode ser o atraso da educação no contexto de teste: a maior parte de um currículo típico de graduação em Ciência da Computação é dedicada às atividades de desenvolvimento (The Joint Task Force on Computing Curricula, 2015), enquanto disciplinas que abordam teste de software frequentemente são negligenciadas em favor das atividades de *design* e implementação. Ou seja, não há ênfase suficiente no ensino das habilidades de teste que a indústria de software demanda (Clarke et al., 2014; Gresse von Wangenheim e Silva, 2009). Agravando a situação, há estudantes e profissionais que consideram as atividades de teste de software desinteressantes em comparação a outras de caráter construtivo (como a programação).

Além do tempo dedicado ao ensino de teste não ser suficiente, resultando na falta de profissionais capacitados, há outros problemas na abordagem pedagógica atual (convencional). Dentre eles, destacam-se o desalinhamento entre o que se ensina e o que a indústria aplica, a ineficiência de aulas teóricas, aulas pouco atrativas, e a falta de atividades práticas. Tais problemas podem causar o desinteresse, desatenção e, conseqüentemente, a desmotivação e baixo desempenho dos alunos.

Nesse sentido, autores como Soska et al. (2016) e Barbosa e Maldonado (2011) observam que a indústria, reconhecendo este problema, tem demandado uma mudança no modelo de educação que consiga abranger as mudanças sociais, culturais e industriais a fim de capacitar (melhor) os alunos dentro das exigências e necessidades que o mercado tem apresentado. Em resposta a essas demandas, diversas iniciativas no contexto de teste de software têm sido propostas como o uso de jogos sérios (Rojas e Fraser, 2016b), *Flipped Classroom* (Paschoal et al., 2017), e gamificação (Yujian Fu e Clarke, 2016). Apesar dos esforços, estas abordagens emergentes ainda têm sido pouco discutidas e investigadas (Smith et al., 2012).

No trabalho de Rojas e Fraser, os autores apresentaram uma abordagem que provê experiência prática no ensino de teste mudante através do jogo CODE DEFENDERS⁷. Dentro os objetivos buscados, Rojas e Fraser visavam melhorar o desempenho dos alunos, e aumentar seu engajamento, motivação e diversão enquanto aprendiam. Além de elementos básicos de jogos (sendo, *Pontos, Selos, Quadro de classificação* - apresentados na próxima seção), foram utilizados os elementos *Duelo e Times*. Embora os autores tenham apresentado as hipóteses da pesquisa, nenhum resultado foi discutido, tampouco qualquer avaliação foi realizada.

Paschoal et al. (2017), por sua vez, exploraram o modelo educacional *Flipped Classroom*, cuja ideia é transferir o foco e prioridade de aprendizagem do professor

⁷Disponível em <<http://code-defenders.org/>> - Acessado em 28 de novembro de 2019.

para o aluno. Ou seja, as atividades passivas de aprendizagem de um conteúdo antes realizadas na sala de aula, passam a ser aprendidas pelo aluno fora da sala de aula. Dessa forma, o professor age como supervisor no horário ordinário de aula, que passa a ter outra finalidade como discussões de dúvidas levantadas pelos alunos, atividades práticas e resolução de problemas. Em sua pesquisa, Paschoal et al. concluíram que o modelo investigado é viável para o ensino de teste de software. Comparando os alunos que aprenderam com o modelo *Flipped Classroom* e com o modelo convencional, os autores descrevem os seguintes resultados: (i) o esforço empreendido no estudo fora da sala de aula foi maior do que no modelo convencional; (ii) não houve diferença significativa entre os grupos em relação ao desempenho e motivação dos alunos; e (iii) os alunos que aprenderam com o modelo alternativo foram mais eficazes na identificação de requisitos de teste, sugerindo que o modelo promove uma melhoria no aprendizado dos alunos.

Por fim, Yujian Fu e Clarke (2016) propuseram um ambiente de aprendizado chamado WReSTT-CyLE⁸. Este é um ambiente gamificado também com o uso de elementos básicos visando engajar e motivar os alunos a aprenderem teste de software. O sistema funciona como um repositório web que contém tutoriais de ferramentas e outros conteúdos sobre as técnicas de teste, porém nenhum outro detalhe como os critérios, níveis, ou fases do processo de teste abordados foram descritos. Um dos pontos avaliados pelos autores foi se há algum relacionamento entre as notas reais dos alunos com os pontos recebidos no ambiente gamificado. Como resultado, os autores concluíram que tal relacionamento existe (mesmo com viés), e que o WReSTT-CyLE é um sistema eficiente para engajar e motivar os alunos a aprenderem teste de software.

Diante do exposto até o momento, entende-se que teste de software não é um processo simples de se realizar, tampouco de se ensinar e aprender. Ao contrário, as atividades de teste são consideradas complexas, desafiadoras, longas, repetitivas, e exigem grande esforço intelectual, seja na prática da indústria ou no ambiente acadêmico (Bell et al., 2011; Fraser, 2017; Yujian Fu e Clarke, 2016). Embora haja ferramentas de teste automatizados para facilitar o trabalho, o esforço humano não pode ser substituído, especialmente em sistemas críticos (Parizi, 2016). Além disso, a experiência, intuição, inteligência, e criatividade humana são essenciais para a criação de casos de teste eficientes (Fraser, 2017).

Aspectos como a repetição e longa duração caracterizam as atividades de teste como sendo, algumas vezes, desmotivadoras, tornando-as fortes candidatas à gamificação. Os principais conceitos desta estratégia emergente que vem sendo utilizada são apresentados a seguir.

⁸Disponível em <<http://wrestt.cis.fiu.edu/>> - Acessado em 28 de novembro de 2019.

2.3 Gamificação

A fim de contextualizar o que é “*gamificação*”, apresenta-se o seguinte cenário⁹: Há um sistema no qual os funcionários de uma empresa devem registrar suas atividades de trabalho seguindo um conjunto pré-estabelecido de regras. Todos os dias. Porém, o departamento de recursos humanos enfrenta enormes desafios, inclusive legais, porque os funcionários não fazem tais registros, pelo menos não diariamente, nem como deveriam. Então o diretor da empresa decide incorporar ao seu sistema alguns recursos com o objetivo de motivar seus colaboradores a fazerem o que tem que ser feito: o registro. A equipe responsável por desenvolver os novos recursos implementa um sistema de pontos para recompensar aqueles que registrarem as suas atividades diariamente, além de somarem pontos extras quando fizerem corretamente. A quem registrar suas atividades por cinco dias consecutivos, será atribuído um selo publicamente visível em seu perfil; se isto se repetir por 3 vezes, então outro selo será dado, e assim por diante. Além disso, um quadro de líderes é implantado classificando os funcionários de acordo com os seus pontos ganhos. Os três primeiros que liderarem o ranking por mais tempo, em rodadas de dois meses, serão reconhecidos e poderão trocar seus pontos por prêmios reais. Como resultados, após seis meses de uso do sistema com estes novos recursos, os problemas do departamento relacionados com o registro de atividades diminuíram em mais de 90%; além disso, os funcionários passaram a perceber, com seus próprios *feedbacks*, o progresso de suas atividades, monitorando quais tarefas necessitam de mais atenção.

No cenário apresentado, o diretor *gamificou* o seu sistema, ou seja, motivou nos funcionários uma reação por meio do uso de elementos de jogos (pontos, selos, quadro de líderes, bens virtuais) fora do contexto de jogo (na empresa). Uma sala de aula é outro exemplo, porém em um contexto diferente: o educacional. Embora seja possível (e proposital) adicionar diversão gamificando-se um ambiente, isto deverá ser feito com finalidades como aumento da motivação, engajamento e mudança comportamental, mas não para torná-lo um jogo.

Assim, *gamificação* é definida como o uso de elementos de jogos em um contexto não-jogo (Deterding et al., 2011). Mas o que é um *jogo*? Ou o que são *elementos de jogos*? E “contexto não-jogo”? Inicialmente, é importante diferenciar *jogos*, *jogos sérios*, e *gamificação*, pois há uma importante diferença entre estes conceitos que, como Werbach e Hunter (2012) destacam, são facilmente confundidos.

Jogos têm como finalidade o entretenimento. *Jogos sérios* são jogos por possuírem as peculiaridades do jogos, porém diferem do conceito de jogo por servirem a um propósito (sério) que não tem por objetivo apenas o entretenimento. A *gamificação*,

⁹Este cenário não é real e foi adaptado do trabalho de Werbach e Hunter (2012).

diferentemente dos outros dois, **não** é um jogo, **não** se aplica ao contexto de jogo, tampouco cria um jogo, mas utiliza partes dos jogos, conhecidas como elementos, em atividades do mundo real que não buscam o entretenimento (contexto não-jogo), tais como o ensino ou os negócios.

A gamificação inclui três categorias destes elementos: *dinâmicas de jogo*, *mecânicas de jogo*, e *componentes de jogo*. Embora existam na literatura diversos elementos e classificações, neste trabalho serão considerados os principais definidos por Werbach e Hunter (2012). Além disso, o termo *elemento* será utilizado ao se referir aos componentes de jogo.

Nas seções seguintes serão apresentados mais detalhes sobre jogos e gamificação.

2.3.1 Conceitos sobre Jogos

Como o termo gamificação teve origem na indústria de mídia digital (Deterding et al., 2011), é importante compreender alguns dos conceitos que abrangem os jogos.

Werbach e Hunter (2012) resumem que “*jogo é o que acontece no círculo mágico*”. O *círculo mágico*, por sua vez, é um limite temporário de um espaço externo ao mundo real, contendo atividades de orientação própria e regras definidas que são aceitas e seguidas (Huizinga, 1980).

De acordo com Koster (2013), há diversas outras definições sobre jogos. No entanto, segundo o autor, poucas definições incluem a diversão (do inglês, *fun*) como requisito principal de um jogo. Koster afirma que a diversão é uma fonte de prazer no cérebro gerada pela liberação de endorfina; além disso, é causada quando o jogo desafia o limite das habilidades dos jogadores em um nível balanceado. Para Huizinga (1980), a diversão é a essência do jogo.

Katie Salen (2004), por sua vez, afirma que não há uma definição própria do termo, porém sumariza as definições de diversos autores sobre o tema e conclui: “*um jogo é um sistema no qual jogadores engajam em um conflito artificial, definido por regras, e com resultados quantificáveis*” (tradução própria).

Por fim, Kapp (2012) adapta a definição de jogos de Katie Salen (2004) adicionando o conceito de Koster (2013) sobre diversão. Assim, o autor define que jogo “*é um sistema no qual os jogadores se engajam em um desafio abstrato, definido por regras, interatividade, e feedback, com resultados quantificáveis provocando reação emocional*” (tradução própria).

Nestas definições apresentadas, percebe-se a presença de alguns elementos como jogador, engajamento, universo externo ao mundo real, desafio, regra, decisão, consequência (resultados), interação, *feedback*, e emoção. Dessa forma, é possível unir estes elementos, definindo-se jogos como um universo externo ao mundo real, onde as novas regras prevalecem, e os jogadores interagem se engajando a fim de

resolverem os desafios tomando decisões que, por sua vez, geram consequências, *feedback*, e emoções.

Diferentemente de *brincadeiras*, os jogos são estruturados por regras e possuem objetivos próprios a serem cumpridos (Deterding et al., 2011). Na verdade, segundo Malone (1981), um jogo não existe se não houver um objetivo claro. Quando os jogadores voluntariamente ultrapassam a fronteira para o círculo mágico, evadem o mundo real com um intervalo em sua vida cotidiana aceitando e passando a seguir as regras do jogo (Werbach e Hunter, 2012). No entanto, há alguns destes jogadores que trapaceiam estas regras para proveito próprio, configurando um comportamento conhecido como *gaming the system*. Para Koster (2013), quando este comportamento ocorre, duas interpretações são possíveis: falta de ética, ou astúcia para sobreviver e vencer. No entanto, para McGonigal (2011) e Huizinga (1980), trapaças não são bem-vindas, tampouco aceitas pelos jogadores.

Os jogos podem ser distintos entre si por possuírem variadas características; por outro lado, eles também podem ser similares por compartilharem os mesmos elementos como sistemas de recompensa e competitividade. No entanto, de acordo com McGonigal (2011), estes recursos não definem os jogos, mas reforçam as quatro peculiaridades que os identificam (definem): *participação voluntária*, *sistema de feedback*, *regras*, e *objetivo*. A autora explica que a participação voluntária indica que, conhecendo as regras, os objetivos, e os *feedback*, os jogadores livremente aceitam jogar podendo entrar ou sair do jogo a qualquer momento. O sistema de *feedback* mantém o jogador informado sobre o quão perto ou quão longe está de alguma conquista; além disso, possui diversas formas e dão a sensação de progresso no jogo. As regras, por sua vez, definem o que pode ou não ser feito para conquistar um objetivo, assim limitando as ações dos jogadores e os desafiando a buscarem novas soluções. Por fim, o objetivo dá ao jogador um senso de propósito, mantendo seu foco em conquistar o resultado específico do jogo.

De acordo com Bartle (1996), os jogadores (normalmente) escolhem os jogos que refletem suas habilidades. O autor distingue quatro tipos de jogadores, sendo eles: conquistadores (*achievers*), exploradores (*explorers*), socializadores (*socializers*), e competidores (*killers*). Segundo o autor, os conquistadores consideram como objetivos próprios conquistar o máximo de recompensas oferecidas, como pontos, níveis, e selos. Os exploradores tentam conhecer o máximo de informações e detalhes presentes no jogo, por exemplo os caminhos secretos, comandos para ataques mais potentes, e pontos extras. Os socializadores desempenham o papel de jogador como contexto para aproveitar os meios de comunicação dos jogos para interagir e colaborar com outros jogadores. Por fim, os competidores (ou matadores) buscam atingir os outros jogadores (no mundo virtual) utilizando as ferramentas disponíveis como armas, ataques, ou carros. Além disso, um jogador com um perfil de conquistador, por exemplo, pode

jogar explorando os detalhes do jogo, porém (normalmente) será com a finalidade de aumentar suas conquistas (Bartle, 1996).

Como considerações interessantes, Koster (2013) afirma que o jogo que leva os jogadores a explorarem as possibilidades para alcançarem um objetivo os ensina a considerarem um problema por diversos ângulos. Para Huizinga (1980), o jogador com perfil de competidor busca vencer a fim de manifestar sua superioridade, estima, e honra diante dos outros jogadores. Na Figura 2.1, Bartle (1996) apresenta o interesse dos jogadores em agir ou interagir com os outros jogadores, e em agir ou interagir no mundo do jogo em si.



Figura 2.1: Gráfico de interesse dos jogadores

Fonte: Bartle (1996)

Outro conceito similar aos jogos e à gamificação é o jogo sério, (ou jogo com propósito). Dicheva et al. (2015) e Deterding et al. (2011) definem jogos sérios como jogos completos cujos propósitos não são apenas o entretenimento. Ou seja, aplicados em contextos como a saúde, aviação, treinamento militar, ou educação, os jogos sérios utilizam da diversão proporcionada pelos jogos com a finalidade de alcançar um objetivo do mundo real servindo, assim, a um propósito (sério).

O mundo artificial dos jogos provê aos jogadores um ambiente seguro (Katie Salen, 2004), no qual os permitem realizar tentativas mais arriscadas que, possivelmente, não fariam no mundo real. Um dos benefícios resultantes dessas tentativas é a inovação.

2.3.2 Conceitos sobre Gamificação

A gamificação, conceituada como a inserção de elementos de jogos em um contexto que não é um jogo, teve origem na indústria de mídia digital e passou a ser amplamente adotada somente no segundo semestre de 2010 (Deterding et al., 2011) como uma estratégia de motivar e engajar as pessoas (Werbach e Hunter, 2012). Para Kapp (2012), o fator que colaborou para esse crescimento foi o aumento do uso de vídeo games.

De acordo com Werbach e Hunter (2012), a definição de gamificação não é universalmente aceita. Os autores adicionaram à definição de Deterding et al. (2011) o uso de *técnicas de design de jogos* em contextos não-jogo. Koster (2013), por sua vez, define gamificação como uma prática de design que tem por objetivo engajar os usuários com o uso de elementos de jogos em sistemas que não são divertidos de se usar.

Segundo Deterding et al. (2011), dependendo do ponto de vista, um sistema gamificado pode ou não ser compreendido como um jogo; ou seja, na perspectiva do *designer* (ou da pessoa responsável pela gamificação), o uso dos elementos de jogos não cria um jogo, enquanto na perspectiva do usuário, um sistema gamificado (que “simplesmente” contém elementos de jogos) pode ser utilizado como se fosse um jogo. No entanto, destaca-se que gamificar não é criar um jogo (Werbach e Hunter, 2012), pois o contexto onde os elementos de jogos são inseridos não visa unicamente o entretenimento. A gamificação apenas utiliza estes elementos no mundo real, fora da fronteira que adentra ao círculo mágico (jogo).

E quanto aos jogos sérios? Por não visarem o entretenimento, seriam uma gamificação? Não. Eles são jogos por possuírem as características dos jogos e utilizarem de sua essência que é a diversão, porém se diferenciam dos jogos tradicionais por não possuírem como propósito único o entretenimento. Além disso, os jogos sérios fazem parte do círculo mágico, universo além de onde a gamificação trabalha.

Na Figura 2.2, Deterding et al. (2011) separam os conceitos de gamificação com outros relacionados em um gráfico cujo eixo vertical os dividem entre *jogo* e *brincadeira*, e o eixo horizontal entre *todo* e *parte*. No gráfico, destaca-se a diferenciação de jogos e gamificação entre *todo* e *parte*. O exemplo a seguir visa representar essa divisão.

As geladeiras têm como finalidade resfriar e manter a temperatura do conteúdo que estiver em seu interior, tais como bebidas e alimentos. Muitos modelos mais modernos possuem painéis digitais que permitem, por exemplo, programá-la para resfriar os alimentos de um compartimento de forma mais rápida. No entanto, o uso desse elemento computacional (painel digital) não cria um computador capaz de resfriar os alimentos, mas adiciona à geladeira recursos com novas formas de funcionamento. Analogamente, o uso de elementos de jogos em ambientes (digitais ou não) não os

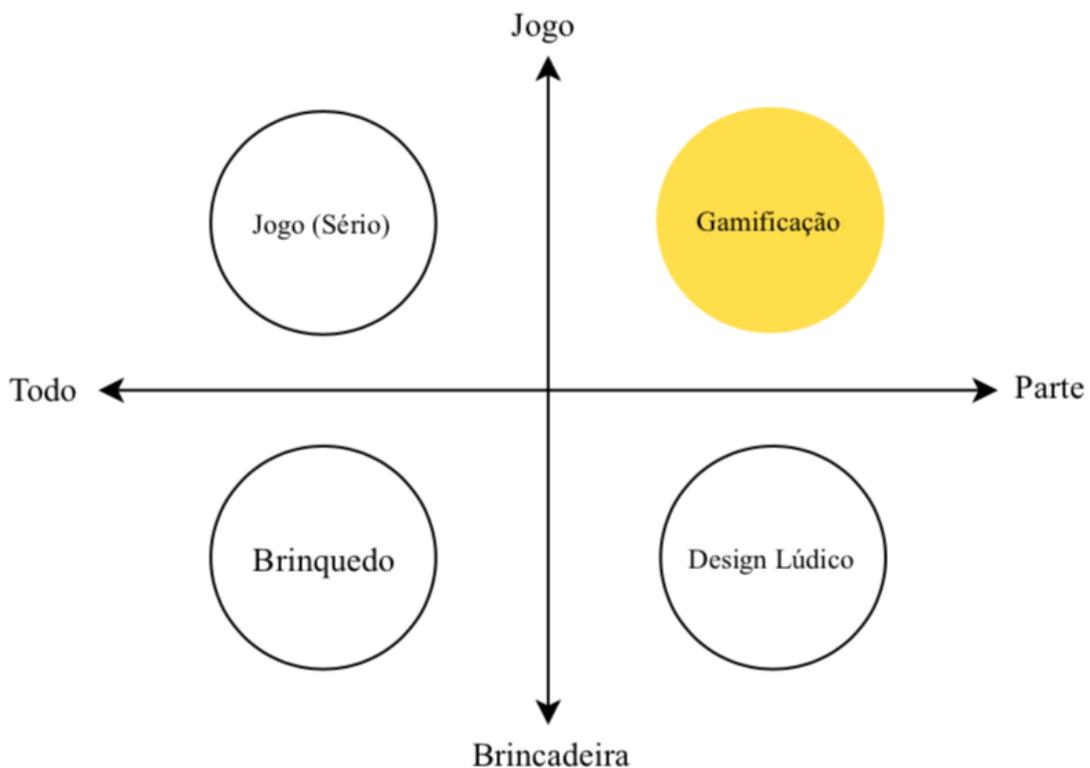


Figura 2.2: Separação de gamificação e outros conceitos.

Fonte: Deterding et al. (2011)

tornam um jogo; eles continuam tendo as mesmas finalidades como o ensino ou o treinamento, porém incrementados com recursos que motivam os seus usuários.

Werbach e Hunter (2012) apresentam três contextos nos quais a gamificação é promissora: gamificação interna, externa, e voltada à mudança de comportamento. Na gamificação interna, busca-se aumentar a motivação e produtividade das pessoas pertencentes ao ambiente onde ela é aplicada; além disso, é essencial o uso de elementos motivacionais que proporcionem aos envolvidos *status* ou aprendizagem. A gamificação externa é voltada a engajar consumidores de um produto sendo aplicada, por exemplo, em *websites*. A gamificação para mudança de comportamento, por sua vez, visa provocar novos hábitos saudáveis na população. Destaca-se que a gamificação interna aplicada no contexto educacional é interessante para este trabalho.

2.3.3 Elementos de Jogos

Werbach e Hunter (2012) comparam os *elementos de jogos* com pequenas peças de Lego. Utilizando vários Legos, é possível construir brinquedos como um carro, uma casa, ou quantos objetos a imaginação e habilidade permitirem. De forma similar, unindo os elementos (mas não apenas), desenvolve-se um *jogo*, um *jogo sério*, ou um

sistema *gamificado*. Deterding et al. (2011), por sua vez, tratam estes elementos como *building blocks*, ou recursos oriundos dos jogos.

Werbach e Hunter (2012) dividem os elementos mais importantes em três categorias relevantes, sendo elas (em nível decrescente de abstração): *dinâmicas, mecânicas, e componentes*. As dinâmicas são aspectos abstratos do que se pretende no jogo (ou no ambiente gamificado); não são diretamente instanciáveis dentro do jogo, e nem sempre são facilmente notados pelos usuários. As mecânicas, também abstratas, são o que conduzem as ações dos usuários de forma a engajá-los. Por fim, os componentes são elementos concretos e instanciáveis no jogo de forma visível aos usuários.

Por exemplo, recompensar (mecânica) um aluno atribuindo pontos (componente) provê a ele um senso de sua progressão (dinâmica); porém, pontos são visíveis aos alunos, enquanto a progressão e o ato de recompensar são apenas perceptíveis.

Os três níveis de elementos se relacionam no sentido de que cada mecânica conduz os usuários, por meio dos componentes, a tornarem uma ou mais dinâmicas perceptíveis.

Na Figura 2.3 representam-se os elementos de jogos divididos nos três níveis definidos por Werbach e Hunter (2012).



Figura 2.3: Hierarquia dos elementos de jogos

Fonte: Werbach e Hunter (2012), adaptado de Mastrodicasa (2014)

O uso destes elementos visa motivar os usuários. Segundo Malone (1981), uma atividade é intrinsecamente motivadora se as pessoas não dependem de, ou não buscam recompensas externas para realizarem, ou seja, a realização desta atividade é satisfatória por si só. Porém, nem toda atividade é intrinsecamente motivadora.

Como já visto neste trabalho, atividades longas, repetitivas, complexas, e maçantes, possuem aspectos desmotivadores. A gamificação tem sido aplicada nestes cenários utilizando elementos de jogos como pontos, selos, e quadro de classificação. Estes três elementos são conhecidos como a tríade PBL (do inglês *Points, Badges, Leaderboards*) e são a forma mais comum de prover *feedback* (Werbach e Hunter, 2012). Além disso, o PBL tem mostrado benefícios funcionando como motivadores extrínsecos, estando entre os mais utilizados (Mastrodicasa, 2014). Werbach e Hunter (2012) sugerem que a tríade pode ser utilizada no estágio inicial do processo de gamificação como um motivador extrínseco, mas também advertem que a superficialidade destes elementos torna necessário o uso de outros em conjunto a fim de alcançar o máximo benefício que a gamificação pode oferecer.

Os *designers* de jogos buscam motivar os jogadores a jogarem. De forma similar, os *designers* da gamificação também buscam motivar os usuários a realizarem suas atividades (Werbach e Hunter, 2012). No entanto, a gamificação pode trazer riscos de desmotivação se não for bem planejada. Um exemplo é o quadro de classificação que deve ser utilizado com cautela, pois usuários que se encontram nas últimas colocações podem se sentir desmotivados a alcançar os objetivos. Outro cuidado é com o uso de elementos que não sejam adequados aos perfis dos usuários. Por exemplo, se os usuários presentes em um contexto educacional não possuírem o perfil de competidor, pouco (ou nenhum) benefício seria gerado pelo elemento *combate*. Ademais, se o elemento *equipe* não é inserido em um contexto onde há socializadores, os benefícios da gamificação poderiam não ser obtidos. Por fim, talvez o maior risco de fracasso seja relacionado ao termo conhecido como “*pointisification*”, que se refere ao uso apenas de um sistema de pontos como elementos na gamificação, concentrando a atenção nas recompensas e não em proporcionar uma melhor experiência. Segundo Werbach e Hunter (2012), esta atitude é a forma mais fácil de ter problemas e não atingir os objetivos e potenciais da gamificação.

2.3.4 Exemplo de Aplicação da Gamificação

Conforme visto, a gamificação é uma estratégia recente que tem sido utilizada para finalidades como motivação e engajamento. Pesquisadores têm avaliado os efeitos da gamificação e alguns vieses foram identificados em alguns experimentos como o foco dos usuários se voltarem mais para as recompensas do que para as atividades que estão sendo realizadas em si. No entanto, os resultados em geral têm sido positivos e os benefícios da gamificação têm atraído pesquisadores e profissionais da indústria.

Um exemplo prático é o uso da gamificação no aplicativo Duolingo¹⁰. Este aplicativo é voltado ao aprendizado de novos idiomas e possui diversos elementos de jogos como

¹⁰Disponível em <<https://www.duolingo.com/>> - Acessado em 28 de novembro de 2019.

conquistas (1), selos (2), avatar (3), bens virtuais (4), níveis (5), pontos (6), quadro de classificação (7), e gráfico social (8), conforme destacadas com seus respectivos números na Figura 2.4.

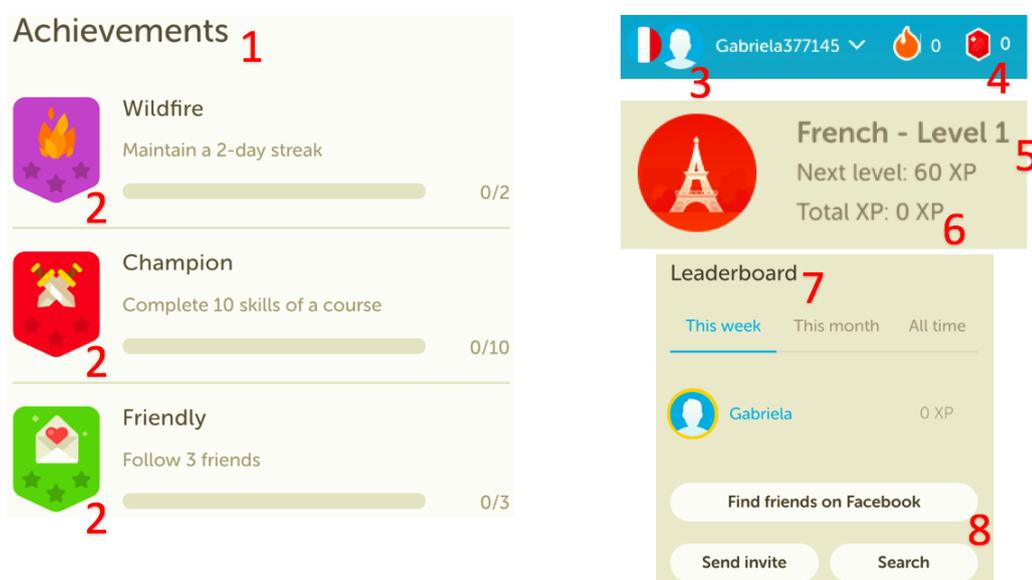


Figura 2.4: Elementos de jogos utilizados no aplicativo Duolingo

Fonte: Duolingo

A seguir são apresentados dois exemplos de estudos cujos autores propuseram ambientes gamificados aplicados em um contexto educacional.

Anderson et al. (2015) apresentaram um ambiente de aprendizado chamado Learn2Mine¹¹. O objetivo dos autores foi prover aos professores um ambiente no qual os professores pudessem ensinar *data science* através de exercícios de programação, e prover aos alunos (tanto em ambiente acadêmico ou industrial) um sistema útil e agradável que os guiassem durante o processo de aprendizagem.

O Learn2Mine contém lições que devem ser resolvidas de forma incremental por meio de subproblemas, até que se alcance o objetivo final de cada lição. Os alunos são recompensados após completarem cada lição e recebem *feedback* imediato após submeterem suas soluções. O ambiente é gamificado com o uso de pontos, selos, quadro de classificação, e combates como elementos de jogos a fim de aumentar a diversão, engajamento, e motivação.

Uma pesquisa foi aplicada pelos autores para avaliar a usabilidade pedagógica e utilidade das características do Learn2Mine; todas as questões receberam respostas positivas, e o sistema de re-submissões recebeu as notas mais altas. Outro experimento foi realizado para comparar resultados de dois grupos de alunos: um com acesso ao ambiente de aprendizado gamificado, e outro grupo sem a gamificação (mas ambos com acesso às re-submissões, correções e notas automáticas, e feedback imediatos).

¹¹Disponível em <<http://learn2mine.appspot.com/>> e <https://github.com/Anderson-Lab/Learn2Mine-Main>> - Acessado em 28 de novembro de 2019.

Os resultados revelaram que o desempenho do primeiro grupo foi significativamente melhor na resolução das tarefas em geral, apoiando as hipóteses dos autores de que um ambiente de ensino guiado e gamificado aumenta a taxa de realizações das tarefas. Alguns alunos deram feedback tanto positivos quanto negativos sobre o uso do sistema, mas nenhum dos que usaram a versão gamificada deram opiniões apenas negativas. Portanto, a conclusão do autores foi que o Learn2Mine é um ambiente benéfico no contexto educacional.

Yujian Fu e Clarke (2016) propuseram, avaliaram, e analisaram a efetividade do sistema WReSTT-CyLE¹² (Web-Based Repository of Software Testing Tutorials - a Cyberlearning Environment). O WReSTT-CyLE é um ambiente de aprendizado cibernético que contém tutoriais de ferramentas baseadas em plataformas IDE (*Integrated Development Environment*), e conteúdos para ensinar técnicas de testes funcional e estrutural.

O ambiente é gamificado com os elementos PBL, níveis, e equipes como estratégia para engajar e motivar alunos a aprenderem metodologias e ferramentas de teste de software, além de poderem melhorar seus conhecimentos.

Yujian Fu e Clarke avaliaram se há um relacionamento entre: os pontos virtuais dos alunos e suas notas reais (a); pontos ganhos e número de postagens (b); e pontos virtuais e frequência de acesso ao curso (c). Os resultados mostraram que há um relacionamento em (a), mas dois vieses foram identificados. Em (b), pouca ou nenhuma correlação foi encontrada. Finalmente, o resultado de (c) foi positivo, mas também com dois vieses. Dessa forma, os autores concluíram que WReSTT-CyLE é um ambiente de aprendizado eficiente, e que há um relacionamento entre gamificação e um ambiente online para engajar e motivar alunos para aprenderem teste de software. Uma observação interessante que os autores mencionaram foi que os alunos mais motivados foram os mais eficientes.

Além destes dois estudos, há outros que também utilizaram a gamificação (especificamente em teste de software) e são apresentados no Apêndice A.

2.3.5 A Importância da Motivação

Autores como García et al. (2017) e Koscianski e dos Santos Soares (2007) afirmam que os recursos humanos são o ativo mais importante de uma organização. Dessa forma, mantê-los motivados e engajados é considerado a chave de sucesso de um projeto. A gamificação é uma estratégia promissora neste cenário porque tem como principal objetivo o uso dos elementos apresentados para motivar as pessoas, o que também envolve diversos aspectos psicológicos (Mastrodicasa, 2014).

¹²Disponível em <<http://wrestt.cis.fiu.edu/>> - Acessado em 28 de novembro de 2019.

De acordo com (Werbach e Hunter, 2012), o termo *motivação*, do Latin *motivus*, traduz-se em dar um sentido para mover-se a fazer algo. Os autores ainda discutem que há dois tipos de motivação, sendo eles: motivação intrínseca e motivação extrínseca, e ambas se resumem no *querer* fazer algo, e no *sentir* que se deve fazer algo, respectivamente. Dentre as diversas teorias que envolvem a motivação, há a *Teoria da Auto-Determinação* (SDT - *Selfdetermination Theory*), na qual sugere-se que a *competência*, *autonomia*, e *relacionamento* são necessidades dos seres humanos. A competência envolve as habilidades e sua eficiência para se realizar uma atividade. A autonomia, por sua vez, é caracterizada pela sensação do indivíduo em se ter domínio sobre as situações, enquanto o relacionamento envolve a sensação de estar conectado e interagindo com outras pessoas (Ryan e Deci, 2000).

Dessa forma, novamente chama-se a atenção quanto à importância de um planejamento adequado para a aplicação da gamificação. Como seus objetivos principais têm sido aumentar a motivação e o engajamento dos usuários (Kapp, 2012), é essencial que a escolha dos elementos e a definição de como eles irão interagir entre si de fato corrobore para que os objetivos sejam alcançados. Por exemplo, quantos pontos serão atribuídos para determinada atividade? A cada quantos pontos um usuário poderá mudar de nível? Qual a intermitência entre as recompensas? Como serão definidos os desafios para as conquistas dos selos? Os usuários poderão trocar seus pontos por privilégios? Gamificação, como visto, não é “pointsification”, tampouco o uso aleatório de elementos em um ambiente. Na verdade, esta é uma promissora estratégia para se alcançar objetivos importantes por tornar as atividades da vida real mais divertidas.

2.4 Considerações Finais

Neste capítulo foram apresentados os principais fundamentos de teste de software e de gamificação, a fim de proporcionar uma melhor compreensão deste trabalho. A respeito de teste de software, foram apresentados os conceitos básicos, técnicas, critérios, níveis, e fases do processo teste. Sobre gamificação, foram apresentados os seus conceitos básicos, assim como os conceitos de jogos, elementos de jogos, e motivação.

Conforme visto, o objetivo das atividades de teste é revelar defeitos no software, sendo assim um processo crítico para o sucesso da qualidade do produto. Por serem atividades longas, repetitivas, e complexas, são caracterizadas como desmotivadoras, o que as tornam fortes candidatas a serem gamificadas.

Gamificação, por sua vez, é uma nova tendência promissora que tem sido utilizada principalmente para aumentar a motivação, engajamento, e performance através do uso de elementos de jogos como motivadores extrínsecos em contextos do mundo real.

No próximo capítulo serão apresentados o planejamento, o processo, a condução, e a análise do Mapeamento Sistemático da literatura que foi realizado para caracterizar como a gamificação tem sido aplicada para apoiar as atividades de teste de software.

Capítulo 3

REVISÃO DA LITERATURA

3.1 Considerações Iniciais

No capítulo anterior foram abordados os conceitos sobre teste de software e gamificação necessários para a compreensão deste trabalho. Dentre estes conceitos, definiu-se que teste é um conjunto de atividades cujo objetivo é revelar defeitos no software, enquanto gamificação é a aplicação de elementos de jogos em contextos do mundo real com propósitos como motivação e engajamento.

Dado o contexto apresentado no Capítulo 1, um Mapeamento Sistemático da literatura foi conduzido visando caracterizar o uso da gamificação como uma estratégia para apoiar (e reforçar a qualidade) das atividades de teste de software. O MS realizado proveu uma visão mais concreta do contexto pesquisado e serviu de subsídio para o desenvolvimento deste projeto de mestrado. Além disso, os resultados do MS que serão apresentados neste capítulo também se encontram em um artigo publicado no III Simpósio Brasileiro de Teste de Software Sistemático e Automatizado (SAST'18)¹ (Jesus et al., 2018).

Ao longo deste capítulo são apresentados o processo e planejamento do MS realizado (Seções 3.2 e 3.2.1), a condução da pesquisa, (Seções 3.3 e 3.4), e a classificação e apresentação dos resultados (Seção 3.5). Em seguida, discussões adicionais são apresentadas (Seção 3.6), seguidas da resposta à questão de pesquisa principal (Seção 3.7). Além disso, limitações identificadas na condução do MS e comparações com outros trabalhos relacionados também são apresentadas (Seção 3.8). Por fim, as observações finais encerram este capítulo.

¹Disponível em <<http://cbsoft2018.icmc.usp.br/#/sast/>> - Acessado em 28 de novembro de 2019.

3.2 Processo e Planejamento da Pesquisa

Mapeamento Sistemático (MS) é um tipo de estudo secundário guiado por um protocolo de pesquisa pré-definido e evolutivo (Fabbri et al., 2013; Petersen et al., 2008); além disso, um MS provê uma visão geral da área a ser estudada e permite a identificação de novas oportunidades de pesquisa.

Um processo de MS é similar ao processo de Revisão Sistemática da Literatura (Fabbri et al., 2013; Kitchenham, 2004), contendo três principais fases, sendo elas: planejamento, condução, e síntese. Essas fases abrangem etapas que foram seguidas para a realização do MS e são apresentadas na Figura 3.1

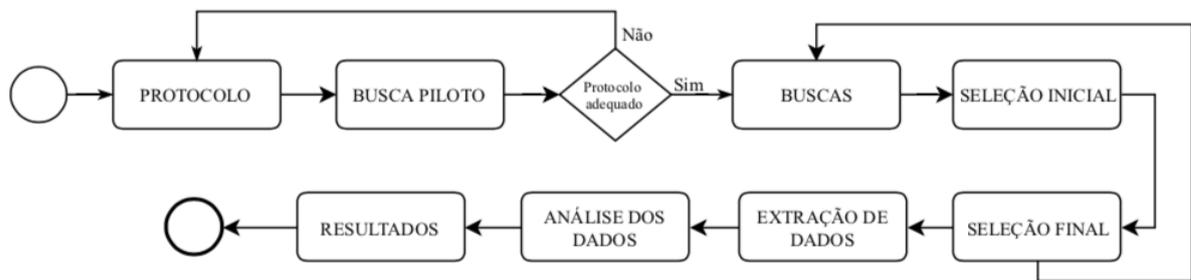


Figura 3.1: Etapas seguidas durante a realização do Mapeamento Sistemático

Fonte: Elaborado pela autora

O planejamento é a primeira etapa do processo de MS, e consiste na definição do protocolo de pesquisa. Durante a sua criação, buscas-pilotos são realizadas a fim de refiná-lo. Após definido o protocolo, as etapas de seleção inicial e final são primeiramente realizadas nos estudos encontrados através da busca automática. As buscas podem ser complementadas pelo processo de *snowballing* (Wohlin, 2014). Tendo definido o conjunto de estudos selecionados após as etapas de seleções, as extrações dos dados e suas análises são realizadas. Por fim, os resultados das análises são sintetizados em formato publicável. Detalhes do protocolo definido e dos resultados obtidos para o MS descrito neste capítulo serão apresentados nas próximas seções.

3.2.1 Protocolo do Mapeamento Sistemático

O protocolo de MS especifica o objetivo do estudo, as questões de pesquisa, a *string* de busca, os critérios de seleção, e os procedimentos para a seleção dos trabalhos encontrados. Estas e outras informações são apresentadas a seguir²:

²Optou-se por manter o tempo verbal do texto do protocolo no futuro. Desta forma, o protocolo é apresentado conforme originalmente definido.

- **Objetivo do estudo:** Dada a recente e crescente atenção que a gamificação tem atraído, este MS tem por objetivo caracterizar o estado da arte de gamificação aplicada em teste de software.
- **Questão de pesquisa:** A questão de pesquisa principal do estudo é: *Como a gamificação tem sido investigada para apoiar o teste de software?*

A resposta para a questão de pesquisa será traçada baseada em seis perspectivas que são abordadas pelas seguintes subquestões (todas relacionadas à gamificação como uma forma de apoiar o teste de software):

1. Em que contexto a gamificação foi aplicada?
2. Quais elementos de gamificação foram utilizados?
3. Quais objetivos do uso da gamificação foram buscados?
4. Quais técnicas de teste foram abordadas?
5. Quais níveis de teste foram abordados?
6. Quais fases do processo de teste foram cobertas?

- **Palavras chaves:** As palavras-chaves para formar as *strings* de busca são “gamificação” e “teste de software”.
- **String de busca e base de dados pesquisada:** A *string* de busca definida é “(gamif*) AND (test*)”.

A ideia é que esta *string* seja genérica retornando uma ampla gama de resultados, de forma a possibilitar sua análise cuidadosa chegando, assim, ao conjunto final de estudos. O caractere “*” é utilizado a fim de considerar todas as possíveis variações de palavras formadas a partir dele. Por exemplo, considerando o termo “test*”, espera-se que sejam retornadas as suas variações como *test*, *tests*, *tested*, *testing*, dentre outros.

A base de dados escolhida é a Scopus, e os campos dos trabalhos a serem considerados são: *title*, *abstract*, e *keywords*.

- **Método de pesquisa:** As pesquisas serão realizadas através de execução automática da *string* de busca, e via *backward snowballing* seguindo apenas uma rodada.
- **Crerios de seleção:** Os critérios de seleção compreendem os critérios de inclusão (i) e exclusão (e) dos estudos encontrados através dos métodos de pesquisa. Eles serão utilizados durante as etapas de seleção inicial e final do MS a fim de selecionar quais dos estudos retornados irão compor a base de discussão para a pesquisa. Os critérios definidos são listados a seguir:

- (i1): Propõe ou aplica uma tecnologia (abordagem, ferramenta, *framework*, método, etc.) para gamificar a educação de teste de software (ensino / aprendizagem).
- (i2): Propõe ou aplica uma tecnologia (abordagem, ferramenta, *framework*, método etc.) para gamificar a prática de teste de software.
- (e1): Não aborda direta ou indiretamente a gamificação de teste de software.
- (e2): Não está escrito em inglês.
- (e3): É um estudo secundário.

Serão selecionados os estudos que passarem em i1 **OU** i2, **E** que **não** passarem por qualquer um dos critérios de exclusão (e1, e2, e3).

- **Definição de tipos de estudos:** Apenas os estudos primários relacionados a gamificação em teste de software serão selecionados para compor a base de discussão. Os estudos secundários serão analisados e discutidos como trabalhos relacionados (Seção 3.8.2), mas não irão compor o conjunto de estudos selecionados.
- **Seleção inicial:** Na etapa de seleção inicial, os critérios de seleção serão aplicados durante a leitura do título, resumo, e palavras-chave dos estudos encontrados através dos métodos de pesquisa a fim de identificar os candidatos à seleção final.
- **Seleção final:** Na etapa de seleção final, os critérios de seleção serão reaplicados durante a leitura completa dos estudos identificados como candidatos na etapa de seleção inicial. Os estudos selecionados serão os que cumprirem os critérios de seleção definidos neste protocolo.
- **Campos para extração:** Durante a leitura completa dos estudos selecionados, um resumo será criado considerando as seis perspectivas abordadas nas sub-questões de pesquisa, sendo elas: contexto de aplicação, elementos de gamificação utilizados, objetivos da gamificação, técnicas de teste, níveis de teste, e fases do processo de teste gamificados. Os dados serão extraídos destes resumos e organizados em uma tabela com base nas seis perspectivas.

Como nota adicional, ressalta-se que inicialmente a *string* de busca definida incluía o termo “*serious games*”. A busca utilizando este termo ocorreu em fevereiro de 2018 e retornou 1400 estudos. Após estudos e discussões com outros pesquisadores, concluiu-se que jogos sérios (do inglês *serious games*) e gamificação se tratam de conceitos distintos, como visto no final da Seção 2.3.1. Dessa forma, a *string* de busca foi redefinida e o protocolo foi alterado a fim de conter tal mudança.

Após a alteração do protocolo, a *string* foi novamente executada na máquina de busca da Scopus. A condução da pesquisa e os resultados das buscas são apresentados na próxima seção.

3.3 Condução da Pesquisa e Resultados das Buscas

A fase de condução do MS abrange as etapas de realização das buscas, seleções inicial e final dos trabalhos encontrados, extração e análise dos dados (Kitchenham, 2004). Um resumo de como esta fase foi realizada após cada método de pesquisa (busca por execução automática e via *backward snowballing*) (Wohlin, 2014) é apresentada na Figura 3.1.

As etapas de seleções ocorreram tanto para as buscas via execução automática quanto via *backward snowballing* (Wohlin, 2014). As seleções iniciais ocorreram por meio da leitura do título, resumo e palavras-chave dos trabalhos encontrados nos dois métodos de pesquisa, enquanto as seleções finais ocorreram através da leitura completa dos trabalhos que foram selecionados durante as seleções iniciais. Em ambas as etapas foram aplicados os critérios de inclusão e exclusão para a seleção dos trabalhos.

Na Tabela 3.1 encontram-se as quantidades de estudos encontrados pela máquina de busca da Scopus (utilizada na execução automática da *string*), no *backward snowballing*, e os valores totais de cada etapa.

Tabela 3.1: Número de estudos em relação aos métodos de pesquisa

Métodos de Pesquisa	Início	Não duplicados	Seleção inicial	Seleção final
Busca automática	540 (57,1%)	540 (66,1%)	47 (63,5%)	10 (52,6%)
Backward Snowballing	405 (42,9%)	277 (33,9%)	27 (36,5%)	9 (47,4%)
Total	945	817	74	19

Fonte: Dados da pesquisa

A *string* foi executada em abril de 2018 no mecanismo de busca da Scopus³, onde foram recuperados 540 estudos (na busca automática não foram verificados estudos duplicados). Na etapa de seleção inicial, 47 estudos dentre os 540 foram aceitos como candidatos. Já na etapa de seleção final, 10 estudos dentre os 47 da seleção inicial foram selecionados. Observa-se que estes 10 trabalhos selecionados são estudos

³Disponível em <<http://www.scopus.com>> - Acessado em 28 de novembro de 2019.

primários. Além destes, outros 3 estudos foram reservados para discussão de trabalhos relacionados, porém não fazem parte dos 10 selecionados por serem (estes 3) estudos secundários.

Em relação ao *backward snowballing*, foram analisadas tanto as referências dos 10 estudos primários quanto dos 3 estudos secundários encontrados através da busca automática. Observa-se que o *backward snowballing* foi realizado em uma única rodada onde, dos 13 trabalhos analisados, foram encontradas 405 estudos, sendo 277 deles não duplicados. Na seleção inicial, 27 estudos dentre os 277 foram aceitos como candidatos. Já na etapa de seleção final, foram selecionados 9 estudos dentre os 27 da seleção inicial.

Do total de 19 trabalhos selecionados, destacam-se 4 estudos (Anderson et al., 2014; Clarke et al., 2014; Mastrodicasa, 2014; Parizi et al., 2015) que foram incluídos (atualizados ou estendidos) por estudos mais recentes. Destes 4 estudos, 1 foi encontrado via busca automática e 3 via *backward snowballing*. Os estudos incluídos e os inclusores são apresentados na Tabela 3.2.

Tabela 3.2: Estudos Incluídos e Inclusores.

Estudos Inclusores		Estudos Incluídos	
Autores, Ano	Título do Estudo	Autores, Ano	Título do Estudo
(Anderson et al., 2015)	Facilitating Programming Success in Data Science Courses Through Gamified Scaffolding and Learn2Mine	(Anderson et al., 2014)	An Extensible Online Environment for Teaching Data Science Concepts Through Gamification
(Yujian Fu e Clarke, 2016)	Gamification-Based Cyber-Enabled Learning Environment of Software Testing	(Clarke et al., 2014)	Integrating Testing into Software Engineering Courses Supported by a Collaborative Learning Environment
(Dal Sasso et al., 2017)	How to Gamify Software Engineering	(Mastrodicasa, 2014)	Ludus Opus Proficit - A Gamification Framework for Software Engineering
(Parizi, 2016)	On the Gamification of Human-centric Traceability Tasks in Software Testing and Coding	(Parizi et al., 2015)	Towards Gamification in Software Traceability: Between Test and Code Artifacts

Fonte: Dados da pesquisa

Por fim, os 15 estudos selecionados restantes passaram a compor o conjunto da base de discussão e são apresentados na Tabela 3.3.

Chama-se a atenção para a diferença entre a quantidade de estudos encontrados e o número de estudos selecionados. Tal diferença é justificada pelo termo “test*” (utilizado na *string* de busca) que levou à combinação de estudos de outras áreas além da Computação (por exemplo, estudos de ciências médicas que abrangeram um software gamificado que, de alguma forma, foi “testado”).

Tabela 3.3: Estudos selecionados através do Mapeamento Sistemático.

Autores	Ano	Título do Estudo
García et al.	2017	A Framework for Gamification in Software Engineering
Rojas e Fraser	2016a	CODE DEFENDERS: A Mutation Testing Game
Rojas et al.	2017	Code Defenders: Crowdsourcing Effective Tests and Subtle Mutants with a Mutation Testing Game
Anderson et al.	2015	Facilitating Programming Success in Data Science Courses through Gamified Scaffolding and Learn2Mine
Yujian Fu e Clarke	2016	Gamification Based Cyber-Enabled Learning Environment of Software Testing
Dal Sasso et al.	2017	How to Gamify Software Engineering
Parizi	2016	On the Gamification of Human-Centric Traceability Tasks in Software Testing and Coding
Liechti et al.	2017	Supporting Agile Teams with a Test Analytics Platform: A Case Study
Laurent et al.	2017	Towards a Gamified Equivalent Mutants Detection Platform
Sheth et al.	2012	Increasing Student Engagement with Gamification
Bell et al.	2011	Secret Ninja Testing with HALO Software Engineering
Rojas e Fraser	2016b	Teaching Mutation Testing using Gamification
Clegg et al.	2017	Teaching Software Testing Concepts using a mutation testing game
Passos et al.	2011	Turning Real-World Software Development into a Game
Dubois e Tamburrelli	2013	Understanding Gamification Mechanisms for Software Development

Fonte: Dados da pesquisa

3.4 Extração dos Dados e Visão Geral dos Estudos Selecionados

A extração de dados basicamente consistiu na elaboração de um resumo de cada estudo selecionado, com especial atenção para o apoio provido pela gamificação à atividade (tanto para educação quanto para a prática) de teste. Durante a elaboração do resumo, seis classificações associadas a cada subquestão de pesquisa foram aplicadas. Cada categoria classificada é apresentada com uma breve descrição na Tabela 3.4 (elementos de jogos), Tabela 3.5 (objetivos da gamificação), Tabela 3.6 (contexto de aplicação), Tabela 3.7 (técnicas de teste), Tabela 3.8 (níveis de teste), e Tabela 3.9 (fases do processo de teste).

Tabela 3.4: Resumo dos conceitos principais dos elementos de jogos.

Categoria	Descrição
ELEMENTOS DE JOGOS	Partes concretas de jogos introduzidos em um ambiente gamificado.
Conquista	Visa definir objetivos a serem alcançados. Podem ser usados para estimular trabalhos árduos.
Avatar	Visa prover uma representação visual dos personagens. Pode ser usado para estimular engajamento, motivação, e causar um apelo emotivo.
Selo	Visa prover uma representação pessoal das conquistas. Pode ser usado para estimular engajamento, motivação, e prover <i>feedback</i> .
Duelo	Visa prover batalhas. Pode ser usado para estimular competição e engajamento.
Quadro de classificação	Visa prover uma visualização pública da performance. Pode ser usado para reforçar a competição e estimular a motivação.
Nível	Visa definir passos a serem alcançados. Pode ser utilizado para estimular engajamento, competição, e prover um senso de progresso.
Ponto	Visa quantificar o progresso. Pode ser usado para encorajar trabalho árduo, e prover <i>feedback</i> pessoal.
Desafio	Visa definir desafios para se alcançar um objetivo maior. Pode ser usado para guiar ações e estimular engajamento.
Gráfico social	Visa representar uma rede social dentro de um ambiente. Pode ser usado para estimular colaboração, engajamento, e motivação.
Equipe	Visa definir grupos para trabalharem em conjunto. Pode ser usado para encorajar colaboração, motivação, e competição.
Bem virtual	Visa prover benefícios de valor virtual ou real. Pode ser usado para estimular engajamento e motivação.

Fonte: Elaborado pela autora

Tabela 3.5: Resumo dos conceitos principais sobre os objetivos da gamificação.

Categoria	Descrição
OBJETIVOS DA GAMIFICAÇÃO	Resultados esperados do uso da gamificação.
Conscientização	Visa aumentar a conscientização das pessoas em relação aos seus progressos e resultados.
Aumentar adoção	Visa aumentar a adoção da prática de teste de software.
Desenvolver criatividade	Visa motivar o desenvolvimento da criatividade na realização das tarefas
Facilitar a correção	Visa motivar as pessoas a realizarem suas tarefas de forma que minimize o esforço no processo de correção de defeitos.
Encorajar hábitos de teste	Visa encorajar desenvolvedores a realizarem testes até que se torne um hábito.
Aumentar o engajamento	Visa engajar as pessoas nas atividades de teste.
Melhorar habilidades	Visa estimular a melhoria do conhecimento dos estudantes, eficiência, performance, dentre outras habilidades.
Aumentar o prazer	Visa aumentar o prazer durante o aprendizado ou a realização de teste de software.
Aumentar a motivação	Visa aumentar a motivação para aprender ou realizar teste de software.
Reforçar o monitoramento	Visa reforçar o monitoramento de todas as pessoas envolvidas no desenvolvimento de artefatos de software complexos.
Aumentar a persuasão	Visa persuadir as pessoas a terem os comportamentos desejados.
Estimular a colaboração	Visa estimular a colaboração entre pessoas envolvidas em uma atividade.
Melhorar o treinamento	Visa melhorar o treinamento para realizar os teste ou outras atividades relacionadas ao desenvolvimento de software.

Fonte: Elaborado pela autora

Tabela 3.6: Resumo dos conceitos principais dos contextos de aplicação.

Categoria	Description
CONTEXTO DE APLICAÇÃO	Contextos nos quais abordagens gamificadas para teste de software podem ser aplicadas.
Educacional	Visa utilizar a gamificação para ensinar teste de software tanto em um contexto acadêmico quanto industrial.
Industrial	Visa utilizar a gamificação em teste de software em um contexto industrial para propósitos de negócios.
Qualquer	

Fonte: Elaborado pela autora

Tabela 3.7: Resumo dos conceitos principais sobre as técnicas de teste.

Categoria	Description
TÉCNICAS DE TESTE	Abordagens baseadas em artefatos de software variados para derivar os requisitos de teste.
Teste funcional	Baseia-se em documentos de especificações de software para derivar os requisitos de teste (também é conhecido como teste caixa-preta).
Teste estrutural	Baseia-se nos detalhes da implementação do software para derivar os requisitos de teste (também é conhecido como teste caixa-branca).
Teste baseado em defeitos	Baseia-se nos defeitos de software recorrentes e documentados (modelos de defeitos e/ou taxonomias) para derivar os requisitos de teste.
Qualquer	

Fonte: Elaborado pela autora

Tabela 3.8: Resumo dos conceitos principais sobre os níveis de teste.

Categoria	Description
NÍVEIS DE TESTE	Considera a granularidade da porção do software que será testado.
Teste de unidade	Visa testar cada unidade do software isoladamente, com a intenção de revelar os defeitos relacionados à lógica de implementação.
Teste de integração	Visa identificar problemas relacionados à interface entre unidades combinadas (unidades integradas) de um dado software.
Teste de sistema	Visa verificar as funcionalidades e performance do software em geral, quando executado em sua infraestrutura final.
Qualquer	

Fonte: Elaborado pela autora

É importante destacar que a lista de categorias das quais são específicas para gamificação (ou seja, elementos de gamificação utilizados, e objetivos de gamificação) evoluíram durante os passos de análise e extração dos dados. Em outras palavras, uma lista completa e abrangente de categorias não foi criada de antemão. Além disso, as listas não são completas e definitivas, uma vez que outros estudos podem ser adicionados a este conjunto final no futuro.

Destaca-se também que a inclusão da opção “Qualquer” em algumas das classificações (sendo: contexto de aplicação, técnicas de teste, níveis de teste, e fases do processo de teste) significa que um dado estudo poderia ser classificado em qualquer categoria específica dentro daquela classificação. Assim, embora seja mostrado na

Tabela 3.9: Resumo dos conceitos principais sobre as fases do processo de teste.

Categoria	Description
FASES DO PROCESSO DE TESTE	Fases presentes em modelos de processo de testes típicos e abrangentes.
Planejamento	Compreende a definição de como o teste será realizado e o que será testado.
Config. de dados e ambiente	Aborda a priorização e a implementação dos requisitos do ambiente de teste estabelecidos no plano de teste.
Projeto de casos de teste	Definição das classes de teste e condições, requerendo acesso ao planejamento e aos artefatos da configuração.
Execução e avaliação	Execução dos testes e eventual registro de falhas percebidas, assim como a garantia de que os objetivos de teste foram alcançados.
Monitoramento e controle	Visa organizar, consolidar, e prover acesso rápido às informações produzidas durante o processo de execução.
Manutenção	Visa manter as suítes de teste (especialmente as automatizadas) durante a evolução do software.
Qualquer	

Fonte: Elaborado pela autora

Tabela 3.13 que nenhum estudo foi explicitamente classificado em “Teste Funcional”, isto não significa que nenhum deles tenha considerado essa técnica. Na verdade, isto significa que todos os estudos classificados como “Qualquer” contemplam (ou são aplicáveis à) a técnica de teste funcional (e todas as outras) como candidata a ser gamificada. Algumas observações adicionais relacionadas às classificações e as categorias vêm a seguir.

Elementos de gamificação utilizados: Dentre os três elementos de jogos (dinâmicas, mecânicas, e componentes) apresentados na Seção 2.3.3, somente as pequenas partes que os usuários conseguem realmente visualizar - componentes - serão considerados na classificação deste MS. Além disso, lembra-se que o termo *elemento* tem sido utilizado ao se referir aos componentes de jogos.

Objetivos da gamificação: Destaca-se, por fim, que os objetivos educacionais/comerciais e os objetivos de gamificação podem não ser os mesmos. Por exemplo, o jogo CODE DEFENDERS (Clegg et al., 2017; Rojas e Fraser, 2016a,b; Rojas et al., 2017) tem como objetivo educacional o ensino de teste de mutação. Por outro lado, seus objetivos de gamificação são, principalmente, aumentar a motivação e o engajamento dos alunos durante o processo de aprendizagem

Uma visão geral de cada estudo é apresentada no Apêndice A. A seguir, discutem-se os resultados, com base nas seis perspectivas definidas nas subquestões de pesquisa.

3.5 Análise e Discussão das Classificações

Os dados extraídos dos estudos selecionados foram divididos em uma planilha de acordo com as seis classificações apresentadas nas Tabelas 3.4, 3.5, 3.6, 3.7, 3.8 e 3.9. As análises destas classificações abordadas nas seções seguintes são relacionadas aos resultados apresentados na Tabela 3.10 (elementos de jogos), Tabela 3.11 (objetivos da gamificação), Tabela 3.12 (contextos de aplicação), Tabela 3.13 (técnicas de teste), Tabela 3.14 (níveis de teste), e Tabela 3.15 (fases do processo de teste).

Tabela 3.10: Resultado da classificação dos estudos na categoria *Elementos de Jogos*.

Perspectiva	#	Estudos
ELEMENTOS DE JOGOS UTILIZADOS		
Conquista	3	(Bell et al., 2011) (Passos et al., 2011) (Dubois e Tamburrelli, 2013)
Avatar	3	(Dal Sasso et al., 2017) (Parizi, 2016) (Passos et al., 2011)
Selo	6	(García et al., 2017) (Anderson et al., 2015) (Yujian Fu e Clarke, 2016) (Dal Sasso et al., 2017) (Liechti et al., 2017) (Passos et al., 2011)
Combate	6	(Rojas e Fraser, 2016a) (Rojas et al., 2017) (Anderson et al., 2015) (Rojas e Fraser, 2016b) (Clegg et al., 2017) (Dubois e Tamburrelli, 2013)
Quadro de classificação	10	(García et al., 2017) (Anderson et al., 2015) (Yujian Fu e Clarke, 2016) (Dal Sasso et al., 2017) (Parizi, 2016) (Liechti et al., 2017) (Laurent et al., 2017) (Bell et al., 2011) (Rojas e Fraser, 2016b) (Dubois e Tamburrelli, 2013)
Nível	10	(García et al., 2017) (Rojas e Fraser, 2016a) (Rojas et al., 2017) (Yujian Fu e Clarke, 2016) (Dal Sasso et al., 2017) (Sheth et al., 2012) (Bell et al., 2011) (Rojas e Fraser, 2016b) (Clegg et al., 2017) (Passos et al., 2011)
Ponto	14	(García et al., 2017) (Rojas e Fraser, 2016a) (Rojas et al., 2017) (Anderson et al., 2015) (Yujian Fu e Clarke, 2016) (Dal Sasso et al., 2017) (Parizi, 2016) (Liechti et al., 2017) (Laurent et al., 2017) (Sheth et al., 2012) (Bell et al., 2011) (Rojas e Fraser, 2016b) (Clegg et al., 2017) (Passos et al., 2011)
Desafio	5	(García et al., 2017) (Parizi, 2016) (Sheth et al., 2012) (Bell et al., 2011) (Passos et al., 2011)
Gráfico social	1	(García et al., 2017)
Time	5	(Rojas et al., 2017) (Yujian Fu e Clarke, 2016) (Sheth et al., 2012) (Bell et al., 2011) (Rojas e Fraser, 2016b)
Bem virtual	1	(Dal Sasso et al., 2017)

Fonte: Dados da pesquisa

3.5.1 Classificação: *Contexto de aplicação*

Foram considerados dois contextos nos quais a gamificação foi aplicada nos estudos selecionados: educacional (tanto acadêmico quanto industrial), e industrial. O contexto educacional foi explicitamente considerado por autores em 10 estudos, enquanto o contexto industrial foi considerado em 8 estudos.

Tabela 3.11: Resultado da classificação dos estudos na categoria *Objetivos do uso da Gamificação*.

Perspectiva	#	Estudos
OBJETIVOS DA GAMIFICAÇÃO		
Aumentar conscientização	1	(Passos et al., 2011)
Encorajar adoção	1	(Rojas e Fraser, 2016a)
Desenvolver a criatividade	1	(Dal Sasso et al., 2017)
Facilitar o processo de correção	1	(Dal Sasso et al., 2017)
Encorajar hábitos de teste	2	(Sheth et al., 2012) (Bell et al., 2011)
Aumentar o engajamento	13	(García et al., 2017) (Rojas e Fraser, 2016a) (Rojas et al., 2017) (Anderson et al., 2015) (Yujian Fu e Clarke, 2016) (Parizi, 2016) (Laurent et al., 2017) (Sheth et al., 2012) (Bell et al., 2011) (Rojas e Fraser, 2016b) (Clegg et al., 2017) (Passos et al., 2011) (Dubois e Tamburrelli, 2013)
Melhorar habilidades	11	(García et al., 2017) (Rojas e Fraser, 2016a) (Rojas et al., 2017) (Yujian Fu e Clarke, 2016) (Dal Sasso et al., 2017) (Parizi, 2016) (Sheth et al., 2012) (Bell et al., 2011) (Rojas e Fraser, 2016b) (Clegg et al., 2017) (Dubois e Tamburrelli, 2013)
Aumentar a diversão	8	(Rojas e Fraser, 2016a) (Rojas et al., 2017) (Anderson et al., 2015) (Liechti et al., 2017) (Sheth et al., 2012) (Bell et al., 2011) (Rojas e Fraser, 2016b) (Clegg et al., 2017)
Aumentar a motivação	9	(Rojas et al., 2017) (Anderson et al., 2015) (Yujian Fu e Clarke, 2016) (Dal Sasso et al., 2017) (Parizi, 2016) (Liechti et al., 2017) (Sheth et al., 2012) (Rojas e Fraser, 2016b) (Dubois e Tamburrelli, 2013)
Reforçar o monitoramento	1	(Dubois e Tamburrelli, 2013)
Aumentar persuasão	1	(Liechti et al., 2017)
Estimular a colaboração	3	(Dal Sasso et al., 2017) (Sheth et al., 2012) (Bell et al., 2011)
Melhorar treinamento	1	(Dubois e Tamburrelli, 2013)

Fonte: Dados da pesquisa

Tabela 3.12: Resultado da classificação dos estudos na categoria *Contextos de Aplicação de Teste*.

Perspectiva	#	Estudos
CONTEXTO DE APLICAÇÃO		
Educacional	2	(Anderson et al., 2015) (Yujian Fu e Clarke, 2016)
Industrial	1	(García et al., 2017)
Qualquer	12	(Rojas e Fraser, 2016a) (Rojas et al., 2017) (Dal Sasso et al., 2017) (Parizi, 2016) (Liechti et al., 2017) (Laurent et al., 2017) (Sheth et al., 2012) (Bell et al., 2011) (Rojas e Fraser, 2016b) (Clegg et al., 2017) (Passos et al., 2011) (Dubois e Tamburrelli, 2013)

Fonte: Dados da pesquisa

No entanto, com poucas exceções, todas as abordagens propostas podem ser aplicadas tanto em contextos educacionais quanto industriais. As exceções foram a abordagem de García et al. (2017) que, como compreendido, não é aplicável a propósitos educacionais devido ao fato de seu *framework* ter sido criado com a finalidade de apoiar os *designers* a gamificarem o ambiente de trabalho, não para ensinarem teste de software; e as abordagens propostas por Anderson et al. (2015) e Yujian Fu e Clarke (2016) que são voltadas à educação por serem sistemas utilizados para o ensino de teste de software, e não para outros propósitos industriais tais como melhorar

Tabela 3.13: Resultado da classificação dos estudos na categoria *Técnicas de Teste*.

Perspectiva	#	Estudos
TÉCNICAS DE TESTE		
Teste Funcional	1	(Parizi, 2016)
Teste Estrutural	5	(Rojas e Fraser, 2016a) (Rojas et al., 2017) (Laurent et al., 2017) (Rojas e Fraser, 2016b) (Clegg et al., 2017)
Teste Baseado em Defeitos	9	(García et al., 2017) (Anderson et al., 2015) (Yujian Fu e Clarke, 2016) (Dal Sasso et al., 2017) (Sheth et al., 2012) (Bell et al., 2011) (Passos et al., 2011) (Dubois e Tamburrelli, 2013)
Qualquer		

Fonte: Dados da pesquisa

Tabela 3.14: Resultado da classificação dos estudos na categoria *Níveis de Teste*.

Perspectiva	#	Estudos
NÍVEIS DE TESTE		
Teste de Unidade	6	(Rojas e Fraser, 2016a) (Rojas et al., 2017) (Parizi, 2016) (Laurent et al., 2017) (Rojas e Fraser, 2016b) (Clegg et al., 2017)
Teste de Integração	9	(García et al., 2017) (Anderson et al., 2015) (Yujian Fu e Clarke, 2016) (Dal Sasso et al., 2017) (Liechti et al., 2017) (Sheth et al., 2012) (Bell et al., 2011) (Passos et al., 2011) (Dubois e Tamburrelli, 2013)
Teste de Sistema		
Qualquer		

Fonte: Dados da pesquisa

Tabela 3.15: Resultado da classificação dos estudos na categoria *Fases de Teste*.

Perspectiva	#	Estudos
FASES DO PROCESSO DE TESTE		
Planejamento de teste	7	(Rojas e Fraser, 2016a) (Rojas et al., 2017) (Parizi, 2016) (Liechti et al., 2017) (Laurent et al., 2017) (Rojas e Fraser, 2016b) (Clegg et al., 2017)
Configuração de dados e ambiente	6	(Rojas e Fraser, 2016a) (Rojas et al., 2017) (Liechti et al., 2017) (Laurent et al., 2017) (Rojas e Fraser, 2016b) (Clegg et al., 2017)
Projeto de casos de teste	5	(Rojas e Fraser, 2016a) (Rojas et al., 2017) (Liechti et al., 2017) (Rojas e Fraser, 2016b) (Clegg et al., 2017)
Execução e Avaliação	6	(Rojas e Fraser, 2016a) (Rojas et al., 2017) (Parizi, 2016) (Liechti et al., 2017) (Rojas e Fraser, 2016b) (Clegg et al., 2017)
Monitoramento e controle	8	(García et al., 2017) (Anderson et al., 2015) (Yujian Fu e Clarke, 2016) (Dal Sasso et al., 2017) (Sheth et al., 2012) (Bell et al., 2011) (Passos et al., 2011) (Dubois e Tamburrelli, 2013)
Manutenção		
Qualquer		

Fonte: Dados da pesquisa

o desempenho dos funcionários ou engajá-los a testarem seus códigos, aumentando assim a qualidade do produto desenvolvido.

Apesar do fato de alguns autores terem mencionado um contexto específico no qual aplicaram suas propostas (educação acadêmica, por exemplo), identificou-se que a maioria delas pode também ser utilizada em outros contextos (educação industrial, ou

mesmo na indústria com propósitos comerciais). Por exemplo, Passos et al. (2011) afirmaram que sua proposta não era intencional ao uso em sala de aula, mas foi percebida uma possibilidade real de utilizá-la para ensinar teste de software. Os autores propuseram uma abordagem para gamificar o processo de desenvolvimento de software, destacando que gamificaram as atividades cotidianas dos desenvolvedores em seu mundo real. Passos et al. então definiram conquistas que o desenvolvedor (ou a equipe) precisaria alcançar para conquistar medalhas (consideradas neste MS como selos). Foi atribuído um selo como recompensa extra ao desenvolvedor que atingisse uma cobertura de código substancial em seu teste automatizado. Convertendo este cenário a um contexto educacional, poderia-se considerar encorajar os estudantes a aprenderem teste de software gamificando-se a sala de aula. Por exemplo, seriam criadas missões que os alunos deveriam cumprir, assim como um determinado número de vezes em que teriam que atingir pelo menos 80% da cobertura de código para ganhar uma medalha. Dessa forma, mesmo não sendo a intenção dos autores, este estudo foi compreendido e classificado como também aplicável ao contexto educacional.

Portanto, para correta leitura da Tabela 3.10 (elementos de jogos), Tabela 3.11 (objetivos da gamificação), Tabela 3.12 (contextos de aplicação), Tabela 3.13 (técnicas de teste), Tabela 3.14 (níveis de teste), e Tabela 3.15 deve-se considerar que as perspectivas utilizadas nas classificações foram embasadas na compreensão das leituras dos trabalhos, e não necessariamente nas intenções apresentadas pelos autores (como neste último exemplo dado).

3.5.2 Classificação: *Elementos de jogos utilizados*

Pontos, quadro de classificação, e níveis foram os elementos de jogos mais utilizados. Dos 15 estudos selecionados, somente Dubois e Tamburrelli (2013) não utilizaram pontos em sua abordagem. Tanto níveis quanto quadro de classificação foram utilizados em 10 estudos cada um. Selo, o qual compõe a tríade PBL, foi mencionado em 6 estudos.

Werbach e Hunter (2012) discutem sobre a tríade PBL e afirmam que ela tem seus benefícios, mas destacam que são elementos superficiais. Os autores também sugerem que PBL podem ser usados como um motivador intrínseco inicial, como feito por Liechti et al. (2017), mas advertem que é necessário ir além, com outros elementos, a fim de alcançar o máximo benefício que a gamificação pode oferecer. Por exemplo, para atingir os objetivos da gamificação pretendidos com o CODE DEFENDERS, Rojas e Fraser (2016a) incorporaram elementos como pontos e combates. CODE DEFENDERS é uma plataforma online utilizada para ensinar teste de mutação: os jogadores podem desempenhar papéis de atacantes (criando mutantes), ou de defensores (criando testes unitários para matar os mutantes). Os pontos são dados a um defensor baseado

em quantos mutantes ele/ela matar; ou eles são dados a um atacante baseado em quantos mutantes sobreviverem. Já os combates ocorrem quando um defensor rotula um mutante como sendo equivalente; assim, o atacante tem que provar que o mutante não é equivalente, criando para tal um caso de teste que mate este mutante; ou tem que concordar, caso contrário.

O único estudo no qual os autores não definiram elementos de jogos concretos foi o apresentado por Dubois e Tamburrelli (2013). Os autores deram um exemplo de como usar a gamificação em um contexto educacional, mas somente mencionaram a mecânica de jogo recompensa, não definindo quais elementos concretos (componentes) poderiam ser utilizados para recompensar um estudante.

3.5.3 Classificação: *Objetivos da gamificação*

Os objetivos do uso da gamificação mais buscados foram aumentar o engajamento, melhorar as habilidades, aumentar a motivação, e aumentar o prazer na realização das tarefas. No entanto, uma variedade de outros objetivos foi identificada nos estudos. Alguns exemplos são impulsionar a adoção, estimular a colaboração, e desenvolver criatividade. Observa-se que as melhorias *eficiência, desempenho, criação de testes mais eficazes, qualidade de registros de defeitos, qualidade de resultados de rastreamento, educação de engenharia de software, resultados de aprendizados, e habilidades de teste* foram unificadas em um único objetivo: *Melhorar habilidades*. A lista completa e a distribuição dos trabalhos podem ser vistos na Tabela 3.11.

3.5.4 Classificação: *Técnicas de teste de software*

Conforme apresentado na Tabela 3.13, os estudos selecionados foram classificados dentro de três técnicas de teste, sendo elas: funcional, estrutural, e baseada em defeitos. Além disso, a opção “qualquer” foi incluída, como já mencionado.

Teste baseado em defeitos foi a técnica mais gamificada. Além disso, 9 dos 15 estudos propuseram abordagens gamificadas que, como compreendido durante as leituras, são aplicáveis a quaisquer das três técnicas.

Por exemplo, Yujian Fu e Clarke (2016) mencionaram que a ferramenta WReSTT-CyLE contém tutoriais e conteúdos para ensinar teste funcional e estrutural; no entanto, seu estudo foi classificado considerando que o sistema pode ser usado para ensinar qualquer técnica, uma vez que outros conteúdos podem ser adicionados no sistema. Da mesma forma, teste funcional e estrutural foram citados por Liechti et al. (2017), mas também foram classificados considerando que qualquer técnica de teste pode ser usada em sua abordagem.

Em relação aos 9 estudos classificados como aplicáveis a qualquer técnica, Yujian Fu e Clarke (2016), Sheth et al. (2012), e Bell et al. (2011) propuseram sistemas gamificados para apoiar o ensino/aprendizagem de teste de software.

A proposta de Anderson et al. (2015) não tem foco em ensinar conceitos de teste de software, mas os envolve nas lições que devem ser resolvidas. Já García et al. (2017), Dal Sasso et al. (2017), Passos et al. (2011), e Dubois e Tamburrelli (2013) propuseram abordagens para gamificar qualquer atividade de engenharia de software. Estas propostas, por possuírem características flexíveis e/ou genéricas, foram consideradas como possíveis de serem aplicadas a qualquer técnica de teste de software.

3.5.5 Classificação: Níveis de teste de software

Conforme apresentado na Tabela 3.13, os estudos foram classificados de acordo com os seguintes níveis de teste: unidade, integração, e sistema. A opção “qualquer” foi aplicada aos estudos que (potencialmente) abordam gamificação em qualquer nível de teste.

Os trabalhos que mencionaram explicitamente um nível de teste foram: Anderson et al. (2015), García et al. (2017), Laurent et al. (2017), Liechti et al. (2017), Parizi (2016), Rojas e Fraser (2016a), Rojas e Fraser (2016b), Rojas et al. (2017), e Clegg et al. (2017). Em todos estes o teste em nível de unidade foi mencionado.

Liechti et al. (2017) foram os únicos autores que explicitamente levaram em consideração os outros dois níveis (teste de integração e teste de sistema).

Bell et al. (2011), Dal Sasso et al. (2017), Sheth et al. (2012), Yujian Fu e Clarke (2016), e Dubois e Tamburrelli (2013) não especificaram um nível de teste em seus trabalhos. No entanto, eles foram classificados como aplicáveis a qualquer nível de teste (considerando a leitura e compreensão destes trabalhos). Os trabalhos de Anderson et al. (2015), García et al. (2017), e Liechti et al. (2017) que abordam teste de unidade também foram propostas aplicáveis a qualquer outro nível.

3.5.6 Classificação: Fases do processo de teste de software

Conforme apresentado na Tabela 3.13, as fases do processo de teste de software que foram consideradas na classificação foram: planejamento de teste, configuração de dados e ambiente, projeto de casos de teste, execução e avaliação, monitoramento e controle, manutenção, e “qualquer” (quando aplicável a qualquer fase).

Nota-se que, inicialmente, manutenção não era considerada uma opção na lista de fases do processo de teste. Esta fase foi adicionada quando se observou que um estudo (Parizi, 2016) explicitamente definiu e explicou as fases que sua abordagem iria abranger (projeto de casos de teste, e manutenção). Além disso, outros autores

(Liechti et al., 2017) mencionaram que a manutenção de uma suíte de teste era uma das razões para terem feito uso do *Fogg Behaviour Model* (FBM).

Com relação à classificação obtida, é importante notar que, embora alguns estudos possuam a mesma classificação nas fases do processo (Clegg et al., 2017; Liechti et al., 2017; Rojas e Fraser, 2016a,b; Rojas et al., 2017), não se pode classificá-los como aplicáveis a qualquer fase. Isto é devido ao fato de que estes trabalhos não são aplicáveis às fases de planejamento de teste, e configuração de dados e ambiente.

Os estudos de García et al. (2017), Anderson et al. (2015), Yujian Fu e Clarke (2016), Passos et al. (2011), Sheth et al. (2012), Dal Sasso et al. (2017), Bell et al. (2011), e Dubois e Tamburrelli (2013) foram classificados como aplicáveis a qualquer fase do processo de teste pela mesma razão explicada na classificação da técnica de teste (ou seja, as abordagens são genéricas e/ou flexíveis).

Como exemplo de um estudo que foi classificado com múltiplas fases de testes, considera-se o trabalho de Parizi (2016). O autor apresentou uma nova abordagem para criar links de rastreabilidade entre artefatos de teste e código. O seu trabalho aborda as fases de projeto de caso de teste, e manutenção. Nesta proposta, os *links* de rastreabilidade são incorporados ao código de teste no momento em que os desenvolvedores projetam os seus testes. Na fase de manutenção, as informações de rastreabilidade podem ser extraídas e uma ferramenta automática verifica e especifica a validade do *status* dos *links*. Com sua proposta, o esforço para verificar estes *links* seria reduzido, e somente na ocorrência de um *status* inválido as informações precisariam ser analisadas e os *links*/testes corrigidos.

3.6 Discussões Adicionais

Diversas abordagens gamificadas propostas nos estudos mencionaram atividades de teste de software automatizado, sendo a maioria deles em nível de teste de unidade. Por exemplo, Rojas et al. (2017) usaram três ferramentas automatizadas com o CODE DEFENDERS: EvoSuite, Randoop, e Major. Os dois primeiros geram automaticamente testes unitários para classes Java em formato JUnit. Major, por outro lado, é uma ferramenta de teste de mutação. Essas ferramentas foram utilizadas para apoiar o modo single-player do jogo, e para comparar os testes e os mutantes criados.

Parizi (2016) também listou algumas abordagens de rastreabilidade automática. O autor discutiu que experimentos revelaram alguns resultados indesejáveis nas métricas analisadas, apesar da alta aplicabilidade destas abordagens. Outra observação importante foi que resultados de técnicas automatizadas (especialmente se aplicadas em sistemas críticos) devem passar por uma inspeção humana. A abordagem gamificada

proposta pelo autor revelou resultados melhores em acurácia que os resultados de um sistema não gamificado.

Liechti et al. (2017) afirmaram que teste automatizado aumenta a qualidade de software significativamente. Em seu trabalho, os autores apresentam um conceito de *test analytics*, o qual é composto por três práticas de metodologias ágeis: melhoria contínua, mecanismos de *feedback*, e teste automatizado. Liechti et al. usaram o *Fogg Behaviour Model* para direcionar o comportamento dos funcionários visando torná-los rigorosos na manutenção de conjuntos de testes automatizados. A gamificação foi usada para adicionar diversão ao processo, motivando, dessa forma, os funcionários a escreverem os testes.

Passos et al. (2011), por sua vez, propuseram gamificação para engajar desenvolvedores em suas atividades diárias. Embora diversos elementos de jogos tenham sido usados para motivá-los, o estudo focou no elemento conquistas. Os autores criaram uma conquista específica para recompensar equipes por sua excelência em teste automatizado, considerando a cobertura de código como medida. Dessa forma, eles destacam a importância de realizar os testes automatizados de forma apropriada.

3.7 Revisitando a Questão de Pesquisa

Os estudos foram analisados e classificados de acordo com seis perspectivas, sendo elas: contexto de aplicação, elementos de jogos utilizados, objetivos da gamificação, técnicas de teste, níveis de teste, e fases do processo de teste gamificados.

A classificação baseada em contexto de aplicação revelou que quase todas as propostas abrangeram tanto contextos educacionais quanto industriais, intencionalmente ou não. Além disso, os elementos de jogos mais utilizados foram pontos, níveis, e quadro de classificação, enquanto os principais objetivos do uso da gamificação nas atividades de teste foram para aumentar o engajamento, melhorar habilidades, e aumentar a motivação.

No gráfico de bolhas da Figura 3.2 apresentam-se resultados combinados baseados nas duas últimas classificações mencionadas: elementos de jogos utilizados, e objetivos da gamificação. No gráfico, considerando-se a leitura vertical, os dados rotulados (e os tamanhos das bolhas) representam o número de vezes que um dado elemento de jogo foi abordado em um estudo que buscou um dado objetivo. Por exemplo, pontos foram usados em 12 estudos que investigaram a gamificação para aumentar o engajamento.

Werbach e Hunter (2012) argumentam que “gamificação é sobre engajamento” (tradução própria). Os resultados apresentados corroboram para esta afirmação. Os

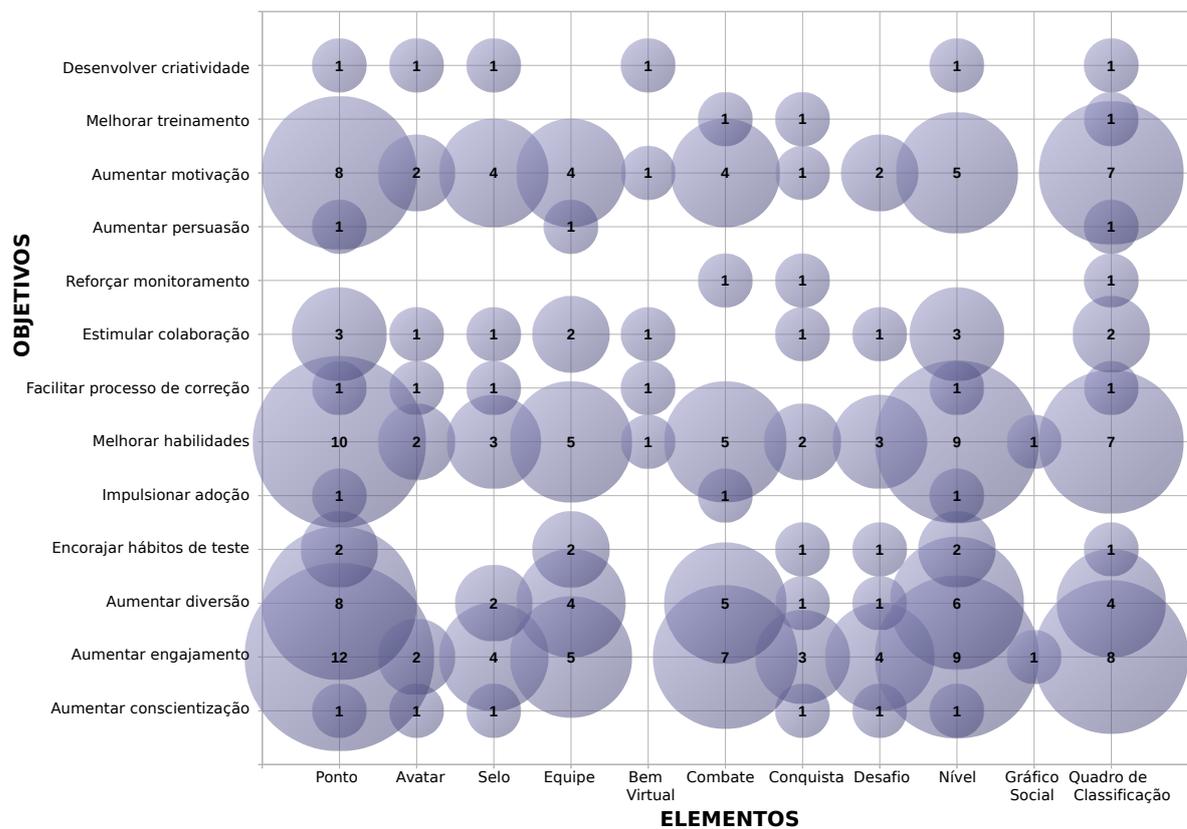


Figura 3.2: Número de estudos que relacionam elementos de gamificação a objetivos da gamificação.

Fonte: Elaborado pela autora

autores também observam que a tríade PBL são os elementos de jogos mais básicos. Uma vez mais os resultados deste MS reafirmam as suas observações.

Levando em consideração as outras classificações que foram aplicadas aos estudos selecionados, a seguinte resposta à questão de pesquisa foi traçada:

Gamificação tem sido investigada para apoiar o teste de software principalmente com a aplicação de elementos de jogos básicos (tais como pontos, níveis, e quadro de classificação), e com o objetivo de aumentar o engajamento, a motivação, e melhorar habilidades, sem qualquer foco claro em uma particular técnica, nível, ou fase do processo de teste.

3.8 Limitações e Trabalhos Relacionados

Nesta seção são apresentadas as limitações identificadas durante o processo de MS, assim como similaridades e diferenças com os trabalhos relacionados.

3.8.1 Limitações

O escopo da busca por estudos primários é a principal limitação do estudo realizado, uma vez que se restringe a: (i) um único mecanismo de busca (Scopus); e (ii) uma única rodada de *backward snowballing*. Além disso, para a busca, foi utilizada uma lista restrita de termos relacionados a gamificação e teste.

Dito isso, argumenta-se que o motor de busca da Scopus recuperou uma ampla gama de resultados, uma vez que foi pesquisada uma abundância de bancos de dados indexados (Kitchenham et al., 2015). No entanto, acredita-se que a inclusão de outros repositórios e termos de pesquisa (como “qualidade de software” e “processo de desenvolvimento”) pode resultar na seleção de outros estudos relevantes que possam aprimorar e ampliar as classificações. Além disso, mais *rodadas* de *backward snowballing*, assim como rodadas de *forward snowballing* também podem contribuir para aprimorar e ampliar as classificações.

3.8.2 Trabalhos Relacionados

Em relação a trabalhos relacionados, alguns estudos secundários também realizaram um MS a fim de obter o estado da arte da gamificação em teste de software. Esses estudos são brevemente descritos a seguir.

Gamification in Software Engineering - A Systematic Mapping - (Pedreira et al., 2015)

Pedreira et al. (2015) realizaram um mapeamento sistemático com o objetivo de caracterizar a gamificação aplicada ao processo de desenvolvimento de software. Baseado na análise de 29 estudos publicados até 2014, os autores responderam questões sobre processos de engenharia de software que têm sido gamificados, elementos de jogos utilizados, e métodos de pesquisa que foram usados na avaliação da qualidade de gamificação de software. Nota-se que os objetivos do MS apresentado neste trabalho de mestrado até então diferem dos objetivos de Pedreira et al., dado que o foco daqueles autores foi em todas as atividades de desenvolvimento de software, enquanto neste trabalho se concentra apenas na gamificação aplicada em atividades de teste de software.

Gamification of Software Testing: An MLR - (Mäntylä e Smolander, 2016)

Mäntylä e Smolander (2016) apresentaram um relatório muito breve sobre uma revisão da literatura conduzida sobre a gamificação de teste de software. Os autores utilizaram o estudo de Pedreira et al. (2015) para realizarem o *backward* e *forward snowballing*. Depois disso, pesquisaram por mais estudos através do *Google Search Engine*. No total, selecionaram 20 estudos e quantificaram os resultados em relação

aos tipos de teste, sistemas em teste, papéis dos indivíduos, elementos de jogos utilizados, estudos que apresentaram evidências empíricas, estudos que apresentaram “*support constructs*” à gamificação, e desafios para gamificar o teste de software.

Em contrapartida, o MS aqui apresentado também analisou os estudos recuperados com base nos tipos de teste (mais especificamente, técnicas de teste) e nos elementos de jogos utilizados. Porém, diferentemente da pesquisa realizada por Mäntylä e Smolander, as informações apresentadas neste MS são mais profundas, além das classificações dos estudos selecionados terem considerado três perspectivas sobre teste de software, sendo elas: técnicas, níveis e fases do processo. Além disso, o MS apresentado neste capítulo forneceu detalhes mais refinados sobre as classificações aplicadas e os resultados da pesquisa.

Gamification of Software Testing - (Fraser, 2017)

Fraser (2017) discutiu problemas que teste de software tem enfrentado, e introduziu a gamificação como solução para resolvê-los. O autor apresentou aplicações de teste de software em três domínios: educação, prática, e crowdsourcing. Diferentemente deste trabalho e dos de Pedreira et al. (2015) e Mäntylä e Smolander (2016), Fraser (2017) não mapeia, revisa, ou caracteriza os estudos encontrados na literatura; nem mesmo tenta responder a uma questão de pesquisa. Ao contrário, ele cita e apresenta o potencial uso de algumas ferramentas gamificadas existentes baseadas nos seus domínios de aplicação.

3.9 Considerações Finais

Como gamificação é uma nova tendência em Engenharia de Software, é importante e conveniente monitorar seu crescimento. Estudos secundários são ferramentas relevantes para esse monitoramento, pois mapeiam, revisam, e caracterizam o estado da arte de determinado assunto a fim de revelar lacunas e oportunidades de pesquisa.

A contribuição principal do MS apresentado neste capítulo foi a caracterização da gamificação como uma estratégia de apoio para o teste de software. Espera-se, com essa contribuição, que antigas e novas abordagens de gamificação sejam utilizadas. Acredita-se também que os resultados apresentados podem (e devem) ser utilizados por pesquisadores e profissionais que visam explorar perspectivas combinadas tais como “quais elementos de jogos já foram utilizados para aumentar o engajamento de estudantes no processo de aprendizagem?”, “quais elementos de jogos já foram utilizados para aumentar o engajamento de profissionais em suas tarefas diárias?”, ou ainda “quais elementos de jogos já foram utilizados para motivar estudantes e/ou

profissionais a aprenderem e aplicarem a técnica de teste funcional?” Dessa forma, tanto contextos educacionais quanto industriais podem tomar vantagem dos resultados que foram aqui apresentados.

Além disso, os resultados do MS discutidos neste capítulo proveram uma visão mais concreta do contexto pesquisado, servindo de subsídio para o desenvolvimento deste projeto de mestrado, cujos procedimentos e métodos utilizados serão apresentados no capítulo seguinte.

Capítulo 4

PROCEDIMENTOS E MÉTODOS

4.1 Considerações Iniciais

No capítulo anterior foi apresentado o mapeamento sistemático da literatura conduzido com o objetivo de caracterizar e compreender como a gamificação tem sido explorada no contexto de teste de software. Concluiu-se que as propostas apresentadas pelos autores dos estudos selecionados são aplicáveis tanto em contextos educacionais quanto industriais, com objetivos como o aumento da motivação e melhoria das habilidades pessoais. Além disso, os elementos de jogos básicos têm sido os mais utilizados.

Percebeu-se como oportunidade de pesquisa a exploração de perspectivas combinadas como, por exemplo, definir um conjunto de elementos de jogos que os diversos autores dos estudos analisados utilizaram quando o objetivo buscado era o aumento da motivação e desempenho.

Ao longo deste capítulo são apresentados o referencial metodológico utilizado nesta pesquisa (Seção 4.2), e os procedimentos realizados (Seção 4.3) para atingir os objetivos definidos e apresentados no Capítulo 1. Em seguida, considerações finais (Seção 4.4) encerram este capítulo.

4.2 Referencial Metodológico

Esta pesquisa de mestrado é de natureza exploratória e experimental. Considerando-se inicialmente sua natureza exploratória, foi realizado um mapeamento sistemático da literatura visando compreender como a gamificação tem sido explorada no contexto de teste de software. Em seguida, alguns dos problemas que o ensino de teste tem enfrentado foram identificados por meio de uma busca *ad hoc*. Um estudo piloto foi

conduzido com o objetivo de identificar possíveis pontos a serem refinados e, por fim, apoiar a construção das hipóteses.

Finalizada a fase exploratória, deu-se início à fase experimental, cujo objetivo foi investigar os efeitos da gamificação no ensino de teste de software. Por meio de sessões experimentais, os dados foram extraídos e analisados. Em seguida, cálculos estatísticos foram realizados para validação das hipóteses construídas (Lakatos e Marconi, 2010).

4.3 Procedimentos

Os procedimentos seguidos para a realização deste trabalho compreendem cinco passos, sendo eles:

- (i) revisão da literatura para: (a) caracterizar o uso da gamificação no contexto de teste de software; e (b) identificar os problemas enfrentados no ensino de teste de software;
- (ii) desenvolvimento da abordagem gamificada (definição dos objetivos e dos elementos de jogos a serem utilizados, criação de um sistema de pontos, recompensas, conquistas);
- (iii) implementação do ambiente gamificado contendo a abordagem desenvolvida;
- (iv) criação dos materiais experimentais para serem utilizados nas sessões experimentais e servirem de base para extração e análise dos dados; e
- (v) condução de execuções experimentais para validação das hipóteses e conclusão com respeito às questões de pesquisas.

Cada passo adotado é apresentado nas subseções seguintes.

4.3.1 Revisão da Literatura

O primeiro passo desta pesquisa foi a realização do Mapeamento Sistemático (MS) da literatura, cujo objetivo foi caracterizar e compreender como a gamificação tem sido explorada no contexto de teste de software. Conforme apresentado no Capítulo 3, os trabalhos selecionados no MS apresentam diversos objetivos buscados com o uso de elementos de jogos em contextos do mundo real, porém nenhum deles buscou resolver ou minimizar os problemas enfrentados pelo ensino de teste de software, tampouco

responder as questões de pesquisa definidas neste trabalho, ambas apresentadas no próximo capítulo deste documento (Seção 5.3.1).

Após o mapeamento sistemático, buscou-se identificar alguns dos desafios que o ensino de teste tem enfrentado. Para isso, foi realizada uma busca *ad hoc* na máquina *Google Scholar*¹ utilizando-se termos como "*problems in software testing education*". Como os resultados foram muito abrangentes, iniciou-se a análise com a dissertação de mestrado de Valle (2016) para extração dos problemas identificados, seguido de outros estudos (Shepard et al., 2001; Smith et al., 2012; Valle et al., 2015) que foram mencionados por Valle. Em seguida, analisou-se outros artigos (Benitti, 2017; Cheiran et al., 2017; Clarke et al., 2014; Gresse von Wangenheim e Silva, 2009; Pinto e Silva, 2017; Soska et al., 2016), que também foram citados pelos estudos anteriores.

Na Tabela 4.1 são apresentados a sumarização dos problemas identificados, assim como a definição das possíveis soluções, comportamentos esperados, e atividades gamificadas para abranger cada problema. Conforme discutido na subseção 2.2.4, um dos problemas identificados é o pouco tempo dedicado ao ensino de teste de software, permitindo assim apenas uma apresentação teórica de um conteúdo que, assim como a programação, demanda tempo e prática para aprendizagem. Além disso, a indústria tem demandado modelos de ensino alternativos aos convencionais diante das diversas mudanças que vêm ocorrendo.

Nota-se que algumas das possíveis soluções apresentadas na Tabela 4.1 não envolvem a gamificação diretamente. No entanto, conforme apresentado na coluna *Atividades Gamificadas*, pode-se utilizar esta abordagem alternativa, inserindo-se elementos de jogos nas atividades a fim de se alcançar os comportamentos esperados dos alunos enquanto se busca resolver (ou amenizar) os problemas identificados. Dessa forma, embora a gamificação tenha como foco abordar questões pontuais (por exemplo, tornar as aulas mais divertidas e agradáveis, atrair a atenção dos alunos, ou ainda reforçar a adoção de determinada técnica), também é possível utilizá-la para apoiar outras possíveis soluções.

Após a identificação dos problemas enfrentados pelo ensino de teste de software, e visando minimizá-los, analisou-se novamente os resultados do mapeamento sistemático realizado na primeira fase desta pesquisa. Conforme mostrado na Figura 3.2, sumarizou-se em um gráfico de bolhas o número de estudos que relacionam os elementos de jogos utilizados com os objetivos buscados com o uso da gamificação. Com base nesse gráfico e nas informações apresentadas na Tabela 4.1, foram definidos quais elementos de jogos deveriam ser utilizados nas atividades aplicadas visando resolver (ou minimizar) alguns dos problemas no ensino de teste, sendo eles a contradição entre o que se ensina na Academia e o que a indústria exige na prática, a diferença entre o nível do conteúdo ensinado e das questões cobradas nas avaliações, ineficiência das

¹Disponível em <<https://scholar.google.com.br/>> - Acessado em 03 de dezembro de 2019

Tabela 4.1: Problemas no Ensino de Teste de Software

Problemas	Possíveis Soluções	Comportamentos Esperados	Atividades Gamificadas
Tempo insuficiente ou inexistente para o ensino de teste de software (Clarke et al., 2014; Gresse von Wannenheim e Silva, 2009; Valle et al., 2015)	Ensinar mais profundamente uma técnica de teste com atividades práticas para fixar o conteúdo	Alunos atentos e participativos durante as aulas	Quizzes competitivos após os conceitos ensinados
Contradição entre o que a Academia ensina e o que a indústria exige (Benitti, 2017; Soska et al., 2016; Valle, 2016)	Ensinar a técnica e os critérios de teste de software mais utilizados na indústria	Utilizar os critérios da técnica de teste ensinada para a criação de casos de teste	Atividade prática em grupo
Diferença do nível em que se ensina e se cobra nas atividades avaliativas (Valle, 2016)	Cobrar nas atividades avaliativas apenas o que foi ensinado e no nível apropriado	Desempenho de pelo menos 70% nos quizzes	Quizzes e atividade prática
Ineficiência das aulas teóricas do modelo convencional de ensino (Barbosa e Maldonado, 2011; Smith et al., 2012; Soska et al., 2016)	Utilizar uma abordagem de ensino alternativa à tradicional	Alunos atentos e participativos durante a aula	Uso de uma abordagem gamificada
Aulas pouco atrativas (Pinto e Silva, 2017; Valle, 2016)	Usar a gamificação para tornar as aulas mais agradáveis e divertidas	Alunos mais motivados, engajados e participativos.	Quizzes competitivos, atividade prática em grupo, uso de um ambiente gamificado
Falta de atividades práticas (Cheiran et al., 2017)	Aplicar atividades práticas após o ensino de um conceito	Performance superior a 70%	Quizzes e atividade prática

Fonte: Elaborado pela autora

aulas apenas teóricas do modelo convencional de ensino, aulas pouco atrativas, e falta de atividades práticas durante as aulas.

4.3.2 Desenvolvimento da Abordagem Gamificada

O desenvolvimento da abordagem gamificada se baseou principalmente nos resultados da revisão da literatura apresentada anteriormente. Uma vez definidos quais problemas seriam tratados com o uso da gamificação, iniciou-se a criação da abordagem respondendo os seis passos para gamificação (conhecidos como *6 D's*) propostos por Werbach e Hunter (2012) (os 6 *D's* serão descritos a seguir em inglês a fim de manter sua originalidade):

1. **DEFINE business objectives:** Atrair a atenção dos alunos para aumentar sua motivação e desempenho no contexto de ensino de teste de software.

2. **DELINEATE** *target behaviors*: Atrair a atenção e fomentar a participação dos alunos.
3. **DESCRIBE** *your players*: Os perfis considerados são os quatro tipos definidos por Bartle (1996), sendo eles: socializadores, exploradores, matadores, e conquistadores. Como não é possível identificar previamente o perfil dos alunos participantes antes das sessões experimentais, todos os quatro tipos são considerados no *design* do ambiente gamificado.
4. **DEVISE** *activity cycles*: Apresentar os conceitos de teste de software → aplicar o *quiz* 1 → ensinar a técnica de teste funcional e seus critérios particionamento de equivalência e análise de valor limite → aplicar o *quiz* 2 → aplicar a atividade prática em grupo.
5. **DON'T** *forget the fun!*: Utilizar a ferramenta gamificada Kahoot!² para inserir os aspectos divertidos e competitivos durante os *quizzes*.
6. **DEPLOY** *the appropriate tools*: Implementar a plataforma gamificada inserindo o conjunto de elementos de jogos definidos a partir da revisão da literatura.

Entendendo que o objetivo buscado nesta pesquisa é atrair a atenção dos alunos de forma a aumentar a sua motivação e desempenho, definiu-se um conjunto de elementos de jogos que deveriam ser utilizados para alcançá-lo. Nota-se que os elementos de jogos a serem inseridos no ambiente gamificado desta pesquisa foram selecionados de acordo com o gráfico de bolhas gerado nos resultados do mapeamento sistemático apresentado na Figura 3.2. No gráfico, combinam-se o número de estudos que relacionam os elementos de jogos utilizados com os objetivos buscados com o uso da gamificação. Dessa forma, o conjunto definido para a gamificação nesta pesquisa de mestrado contém os seguintes 10 elementos de jogos: *avatar, bens virtuais, competição (combate), conquista, desafio, níveis, pontos, quadro de classificação, selo, time (equipe)*. Nota-se que o elemento *gráfico social* presente no gráfico bolha não foi adicionado ao conjunto devido a nenhum estudo selecionado o ter utilizado buscando o aumento da motivação e do desempenho dos participantes.

Além da seleção dos elementos de jogos, os seguintes comportamentos desejáveis foram definidos para se obter com a abordagem gamificada:

- **Atenção**: os alunos devem prestar atenção às explicações do conteúdo ao qual estão sendo expostos durante a aula. Os *quizzes* aplicados, por exemplo, foram uma estratégia para atrair sua atenção, pois a distração os impediria de responder corretamente cada questão. Além disso, eles serviram tanto para instigar a competição entre os alunos quanto para desafiá-los, visando motivá-los a aprenderem mais.

²Disponível em <<https://kahoot.com/>> - Acessado em 28 de novembro de 2019.

Ao final de cada *quiz*, os três melhores colocados são recompensados com pontos, selos, bens virtuais, e a possibilidade de melhorar o avatar caso sua pontuação atingisse o valor necessário para a mudança de nível.

- Participação ativa: os alunos devem participar ativamente fazendo e/ou respondendo perguntas. Para encorajá-los a participar, foram definidos pontos e selo como recompensas.
- Colaboração: os alunos devem colaborar uns com os outros no desafio final; além disso, a plataforma contém uma aba chamada *Fórum*, na qual devem ser postadas dúvidas e respostas. Para motivar a colaboração entre os alunos, foram atribuídos selos, pontos e bens virtuais. Embora colaboração e competição possam ser contraditórias, a primeira foi inserida motivada por expectativas do ponto de vista da indústria, pois saber trabalhar em equipe tem sido uma das *soft skills* mais valorizadas e exigidas em processos seletivos. Por outro lado, a competição foi inserida para aumentar a diversão durante a atividade. A dinâmica competitiva foi introduzida durante os *quizzes* mas, na atividade final, a expectativa foi a colaboração em grupo para vencer o desafio. Sem isso, menos recompensas seriam alcançadas, afetando diretamente tanto os alunos quanto seus times.
- Aplicação dos conceitos aprendidos: os alunos devem utilizar os critérios do teste funcional ensinados (particionamento de equivalência, e análise de valor limite) para a criação da suíte de testes durante a atividade prática. Trabalhando em times, os alunos foram divididos em grupos e tiveram a missão de revelar os defeitos no sistema sendo testado.

Na Tabela 4.2 são apresentados as atividades e os elementos de jogos utilizados para motivar os comportamentos desejáveis dos alunos.

Tabela 4.2: Atividades gamificadas para motivação dos comportamentos desejáveis

Comportamentos Desejáveis	Elementos de Jogos	Atividade
Atenção	Competição, pontos, selo	<i>Quizzes</i> e desafio final
Participação ativa	Pontos, selo	Fazer e/ou responder às perguntas
Colaboração	Moedas, pontos, selo	Fórum e desafio final
Aprender e aplicar os critérios Funcionais	Pontos, selo, desafio, time	Desafio final

Fonte: Elaborado pela autora

Conforme mencionado na subseção 2.3.5, deve-se levar em consideração, no momento da definição das atividades e recompensas, o nível de habilidade exigido para cada desafio, de forma que os participantes não se sintam entediados ou frustrados

(Csikszentmihalyi, 1990). Em relação à coerência entre os níveis de habilidade e dificuldade, tomou-se o cuidado de não extrapolar o escopo do conteúdo ensinado durante as atividades avaliativas. Por outro lado, uma série de dúvidas desafiadoras foram enfrentadas no momento de definir o funcionamento e interação entre os elementos de jogos utilizados, tais como “*quais tipos de recompensas deve-se aplicar para cada conquista?*”, “*qual sistema de pontuação deverá ser usado nas conquistas?*”, e “*qual intervalo de pontuação deverá ser usado entre os níveis?*”.

Embora haja alguns estudos que propuseram *frameworks* para gamificar um ambiente (Dal Sasso et al., 2017; Dubois e Tamburrelli, 2013; García et al., 2017), ou ainda outros estudos que propuseram ferramentas gamificadas para a educação, (Anderson et al., 2015; Yujian Fu e Clarke, 2016), nenhum deles se adequou completamente às necessidades deste projeto de mestrado. Em alguns casos, as propostas não contemplam o conjunto de elementos de jogos definidos para esta pesquisa, outros propõem ferramentas de apoio para uso dos alunos fora da sala de aula, outros são voltados ao ensino de teste de software dentro de outras disciplinas como, por exemplo, de programação ou ciência de dados, e outros ainda são voltados para experimentos com análises de longo prazo (experimentos realizados no decorrer de todo o semestre letivo).

Dessa forma, dentre os estudos selecionados, a proposta de Dal Sasso et al. (2017) foi utilizada como inspiração para apoiar a criação de uma abordagem gamificada própria para esta pesquisa. A proposta de Dal Sasso et al. auxiliou na definição de regras para atribuição de recompensas aos alunos por seus comportamentos desejáveis, a definição de conquistas, as melhorias do avatar, e a definição do sistema de pontuação, dos bens virtuais, dos selos, e do intervalo de pontuações para cada nível a ser alcançado.

A proposta de Dal Sasso et al. foi preliminarmente adaptada, conforme mostrado na Figura 4.1. Entretanto, percebeu-se que tal adaptação tornaria complexo o desenvolvimento do ambiente gamificado, pois cada atividade e suas recompensas seriam descritas em tabelas diferentes, o que seria um dificultador para a concentração e visualização rápida das informações.

Visando concentrar a visualização das informações, uma planilha³ foi desenvolvida com o intuito de incluir o sistema de recompensas, o sistema de pontos e níveis, os avatares, e os bens virtuais. No processo de criação do sistema de pontos, primeiramente definiu-se o intervalo de pontuação de 0 a 100 pontos, sendo 0 a pontuação inicial do nível 1, e 100 a pontuação máxima do último nível. No entanto, cada recompensa aplicaria uma pontuação baixa para um comportamento desejável, o que poderia causar uma sensação de “grande esforço para pouco ganho”. Neste sentido, o sistema de pontos e níveis foi sendo refinado (assim como a lista de conquistas e

³Disponível em <<https://bit.ly/2XaFFez>> - Acessado em 28 de novembro de 2019.

ATIVIDADE	
Descrição	Descrição da atividade realizada pelo aluno.
Recompensas	Definição das recompensas que abrangem a atividade.
IMPLEMENTAÇÃO	
Regras	Definição das regras relacionadas à atividade.
Riscos	Descrição de riscos como trapaças e desmotivação.
Prevenção	Descrição de como prevenir cada risco identificado.

Figura 4.1: Tabela de definição de atividades e implementação

Fonte: Adaptado de (Dal Sasso et al., 2017)

suas recompensas) até que a planilha de definição da abordagem gamificada estivesse pronta com os níveis, atividades e recompensas definidas, conforme mostrado na Figura 4.2. O foco principal desta planilha é definir a faixa de pontuação para cada nível (colunas *H* a *J*) e dividir as atividades e recompensas dentro dos níveis existentes (colunas *B* a *F*). Dessa forma, cada nível contém um conjunto de atividades a serem realizadas (coluna *C*), os objetivos a serem conquistados em cada atividade (coluna *D*), e as recompensas (coluna *E*) e seus valores (coluna *F*) para cada conquista.

Idealmente, é esperado que os alunos estejam, em cada atividade, em seu respectivo nível. Por exemplo, no momento da atividade *Conceitos Básicos*, que consiste no ensino dos conceitos básicos de teste de software, é esperado que os alunos tenham conquistado a pontuação necessária para estarem no *nível 2*. As conquistas que os participantes podem (e devem) alcançar durante esta atividade são manter a *atenção*, e *participação ativa*. Os alunos que se mantiverem atentos e/ou participativos serão recompensados com 100 e/ou 30 *XP*s, respectivamente. A mesma ideia segue para a atividade do *Quiz* presente no nível 2, ou seja, o aluno que conquistar o primeiro lugar receberá 100 *XP*s, o selo *Ouro* e 4 moedas.

Na Tabela 4.3 é apresentada, de forma mais clara, a lista final com as possíveis conquistas e suas respectivas recompensas. A primeira coluna desta tabela contém as conquistas a serem alcançadas, e as quatro últimas colunas contém as recompensas aplicadas para cada conquista.

SISTEMA DE RECOMPENSAS					SISTEMA DE PONTOS E NÍVEIS		
Nível	ATIVIDADE	OBJETIVO	RECOMPENSA	VALOR	NÍVEL	XP MÍNIMO	XP MÁXIMO
PASSOS INICIAIS (PERFIL + TUTORIAL)					NÍVEL 1	0	99
NÍVEL 1	Início	Perfil (PE)	XP	50	NÍVEL 2	100	279
			Selo	Newbie	NÍVEL 3	280	464
			Avatar	Newbie	NÍVEL 4	465	624
		Moeda	1	NÍVEL 5	625	735	
	Tutorial (TU)	XP	50	MELHORIAS AVATAR			
			Selo	Aprendiz	NÍVEL	ITEM	REQUISITO
CONCEITOS BÁSICOS + QUIZ					AVATAR BÁSICO (AB)		
NÍVEL 2	Mudança para Nível 2 (N2)		Avatar	Caçador	NÍVEL 1	Newbie	Criar perfil
			Moeda	2	NÍVEL 2	Caçador	XP >= 100
			Selo	Caçador	NÍVEL 3	Experiente	XP >= 280
	Conceitos Básicos (CB)	Atenção (AB)	XP	100	NÍVEL 4	Mestre	XP >= 465
		Participação (PB)	XP	30	NÍVEL 5	Tutor	XP >= 625
	Quiz (QCB)	1 lugar (QB1)	XP	100	VENCEDOR	Vencedor	Vencer a competição
			Selo	Ouro	AVATAR EXTRA (AE)		
			Moeda	4	AE1	Estagiário	1
		2 lugar (QB2)	XP	90			
			Selo	Prata			
Moeda			3				
3 lugar (QB3)	XP	80					
	Selo	Bronze					

Figura 4.2: Planilha de definição do ambiente gamificado

Fonte: Elaborado pela autora

Tabela 4.3: Conquistas e suas recompensas

Conquistas	XP	\$	Recompensas	
			Selo	Avatar
Acessar a plataforma	50	1	Newbie	Newbie
Acompanhar o tutorial	50	-	Aprendiz	-
Atingir o nível 2	-	2	Caçador	Caçador
Prestar atenção - Conceitos Básicos	100	-	-	-
Participar ativamente - Conceitos Básicos	30	-	-	-
Quiz 1 - Conquistar o 1º lugar	100	4	Ouro	-
Quiz 1 - Conquistar o 2º lugar	90	3	Prata	-
Quiz 1 - Conquistar o 3º lugar	80	-	Bronze	-
Quiz 1 - Outras colocações	70	-	-	-
Atingir o nível 3	-	-	Experiente	Experiente
Presença na sessão 2	-	2	-	-
Prestar atenção - Teste Funcional	115	-	Sentinela	-
Participar ativamente - Teste Funcional	40	-	Participativo	-
Colaborar (Fórum)	50	2	Amigo	-
Quiz 2 - Conquistar o 1º lugar	100	4	Ouro	-
Quiz 2 - Conquistar o 2º lugar	90	3	Prata	-
Quiz 2 - Conquistar o 3º lugar	80	-	Bronze	-
Quiz 2 - Outras colocações	70	-	-	-
Atingir o nível 4	-	-	Mestre	Mestre
Desafio - Conquistar o 1º lugar	100	4	Ouro	-
Desafio - Conquistar o 2º lugar	90	3	Prata	-
Desafio - Conquistar o 3º lugar	80	-	Bronze	-
Desafio - Outras colocações	70	-	-	-
Atingir o nível 5	-	-	Ascensão	Ascensão
Ranking final - Conquistar o 1º lugar	-	-	Ouro	-
Ranking final - Conquistar o 2º lugar	-	-	Prata	-
Ranking final - Conquistar o 3º lugar	-	-	Bronze	-
Ranking final - Outras colocações	70	-	-	-

Fonte: Elaborado pela autora

4.3.3 Desenvolvimento da Plataforma Gamificada

O passo seguinte à criação da abordagem gamificada foi o desenvolvimento do ambiente gamificado chamado *Bug Hunter*⁴, no qual a abordagem criada foi inserida visando alcançar os objetivos de aumentar a motivação e desempenho dos alunos. O *Bug Hunter* contém o conjunto dos 10 elementos de jogos (definidos através do mapeamento sistemático) e foi desenvolvido para ser um ambiente no qual os alunos pudessem receber feedback constante de seu progresso após cada atividade realizada, além de serem recompensados e reconhecidos por suas conquistas. Além disso, o uso da plataforma visou fomentar a competitividade e diversão tornando as aulas mais agradáveis.

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
2	ADM					NÍVEL 1					NÍVEL 2 - Conceitos Básicos								
3	Nome dos Alunos	Total				Perfil			Tutorial		Atenção	Participação	Mudança Nível 2		Quiz Conc. Básico			Mudança Nível 3	
4		XP 0 - 735	Nível	\$	Bens - \$	XP 50	\$1	Selo Newbie	XP 50	Selo Aprendiz	XP 100	XP 30	\$2	Selo Caçador	XP 70 - 100	1º: \$4	2º: \$3	Selo: Ouro, Prata, Bronze	Selo Experiente
5		Aluno 1	615	4	12	0	50	1	Newbie	50	Aprendiz	100		2	Caçador	90	3	Prata	Experiente
6	Aluno 2	625	5	9	0	50	1	Newbie	50	Aprendiz	100	30	2	Caçador	70	0	-	Experiente	
7	Aluno 3	655	5	12	0	50	1	Newbie	50	Aprendiz	100	30	2	Caçador	100	4	Ouro	Experiente	
8	Aluno 4	495	4	9	0	50	1	Newbie	50	Aprendiz			2	Caçador	70	0	-	Experiente	
9	Aluno 5	605	4	9	0	50	1	Newbie	50	Aprendiz	100		2	Caçador	80	0	Bronze	Experiente	
10	Aluno 6	350	3	7	0	50	1	Newbie	50	Aprendiz			2	Caçador	70	0	-	Experiente	
11	Aluno 7	595	4	9	0	50	1	Newbie	50	Aprendiz	100		2	Caçador	70	0	-	Experiente	
12	Aluno 8	635	5	1	-8	50	1	Newbie	50	Aprendiz	100	30	2	Caçador	70	0	-	Experiente	
13	Aluno 9	585	4	9	0	50	1	Newbie	50	Aprendiz	100	30	2	Caçador	70	0	-	Experiente	
14	Aluno 10	595	4	9	0	50	1	Newbie	50	Aprendiz	100		2	Caçador	70	0	-	Experiente	
16	Notificações															Classificação			
17	04/10/2019	08:00	Seja bem-vindo! Vamos iniciar a nossa jornada! Coragem, caçador de bug!!!																
18	04/10/2019	09:10	Sua primeira missão nesta jornada é responder a um breve questionário para podermos te conhecer melhor. Por favor, verifique seu e-mail.																
19	04/10/2019	9:32	Aluno 3 e Aluno 8 estão liderando o ranking, mas ainda há chance de virar este cenário																
20	04/10/2019	11:00	Aluno 3 e Aluno 8 permanecem liderando o ranking! O desafio final será a última chance para reverter os resultados...																
	<div style="display: flex; justify-content: space-between;"> Admin Regras Selos Avatar Loja Virtual Fórum A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 Explo </div>																		

Figura 4.3: Ambiente gamificado - Aba Admin.

Fonte: Elaborado pela autora

O protótipo desenvolvido contém 7 abas nomeadas *Admin*, *Conquistas*, *Selos*, *Avatar*, *Loja*, *Fórum* e *AlunoX*. Conforme mostrado na Figura 4.3, a aba *Admin* é o local onde o instrutor aplica as recompensas aos alunos. Nesta aba, as únicas inserções manuais feitas pelo instrutor durante as sessões experimentais são os *posts* na área de Notificações (célula B16) e a atribuição de XPs⁵ aos estudantes que alcançarem um objetivo. Por outro lado, as células C5:F14 apresentam o cálculo automático, para cada aluno, do total de XPs acumulados (C5:C14), o nível alcançado (D5:D14), total de moedas conquistadas (E5:E14), e total de débito das moedas trocadas por bens virtuais (F5:F14). Dessa forma, as demais recompensas (selos, moedas, mudanças de níveis, e melhorias no avatar) são automaticamente calculadas e aplicadas pela plataforma

⁴Disponível em <<https://bit.ly/2rMhPdr>> - Acessado em 28 de novembro de 2019.

⁵XP significa *Experience Points*, ou Pontos de Experiência.

quando o total de XP e/ou nível conquistado pelo aluno contém o valor definido nas regras de atribuição de recompensas. Nota-se que as células de atribuição manual contém a fonte na cor preta (G5:G14; J5:J14; L5:L14; M5:M14; P5:P14; etc.), enquanto as células com atribuições automáticas contém a fonte na cor vermelha (H5:H14, I5:I14, K5:K14, N5:N14, O5:O14, Q5:Q14, R5:R14, S5:S14).

Como um exemplo de aplicação, quando um aluno acessa a plataforma pela primeira vez, é atribuído a ele 50 XPs, 1 moeda (\$), e um selo *Newbie*. Em seguida, é esperado que este aluno siga o tutorial de apresentação da plataforma. Os alunos que participarem deste tutorial recebem mais 50 XPs e o selo de *Aprendiz*. Com a soma de 100 XPs, o aluno atinge o Nível 2, recebe mais 2 moedas (\$), o selo de *Caçador*, e seu avatar pessoal recebe uma melhoria. Além disso, é liberada a compra do avatar *Estagiário* por 3 moedas.

Cada aluno possui uma aba pessoal nomeada *AlunoX* (substitui-se o “AlunoX” pelo nome do Aluno), conforme ilustrado na Figura 4.4. Nesta aba o aluno recebe feedback a respeito de seu progresso através dos elementos de jogos inseridos na plataforma, visualiza suas recompensas, compra bens virtuais, e acompanha o quadro de classificação e as notificações postadas pelo instrutor. Além disso, todas as informações em cada aba de cada aluno estão vinculadas à aba administrativa e são automaticamente preenchidas.

The image displays a user interface for a gamified environment, divided into several sections:

- Aluno 3 Profile:** Shows a character icon, level 5, 655 XP, and 12 coins.
- Loja (Store):** A table listing items for purchase.

Item:	Preço	Disponível	Comprar
Chapeu de estagiário	1	Sim	<input type="checkbox"/>
Excluir questão quiz	7	Sim	<input type="checkbox"/>
Auxílio no desafio	8	Sim	<input type="checkbox"/>
- Conquistas (Achievements):** A grid of achievement icons including Newbie, Aprendiz, Caçador, Quiz Básico 1º, Experiente, Sentinela, Participativo, Quiz Funcional 2º, Mestre, Desafio 3º, and Ascensão.
- Classificação (Ranking):** A list of top performers.

Classificação	
1º	Aluno 3
2º	Aluno 8
3º	Aluno 2
4º	Aluno 1
- Notificações (Notifications):** A list of messages with timestamps and content.

Notificações		
04/10/2019	8:00	Seja bem-vindo! Vamos iniciar a nossa jornada! Coragem, caçador de bug!!!
04/10/2019	9:10	Sua primeira missão nesta jornada é responder a um breve questionário para podermos te conhecer melhor. Por favor, verifique seu e-mail.
04/10/2019	9:32	Aluno 3 e Aluno 8 estão liderando o ranking, mas ainda há chance de virar este cenário
04/10/2019	11:00	Aluno 3 e Aluno 8 permanecem liderando o ranking! O desafio final será a última chance para reverter os resultados...
04/10/2019	11:10	Este é o grande momento! Uma missão será dada a você. Trabalhe em equipe e coloque em prática tudo o que aprendeu. Quanto mais critérios e fases do processo de teste você utilizar, mais experiência conquistará!

Figura 4.4: Ambiente gamificado - Aba *AlunoX*.

Fonte: Elaborado pela autora

Além das abas *Admin* e individuais de cada aluno, a plataforma possui a aba *Conquistas*, que contém os possíveis objetivos a serem conquistados, a descrição de

como alcançá-los, e as recompensas atribuídas a cada um, conforme apresentado na Tabela 4.3. A aba *Selos*, por sua vez, contém todos os selos que podem ser conquistados e a descrição de como conquistá-los (ver Figura 4.4, área de *Conquistas*). Na aba *Avatar*, conforme Figura 4.5, são exibidos os avatares pertencentes a cada Nível, incluindo o avatar *Estagiário* que pode ser comprado como bem virtual, e o avatar *Vencedor* atribuído somente ao vencedor da competição. Cada aluno possui seu próprio avatar, sendo este um elemento de jogo que funciona como uma dinâmica emocional.

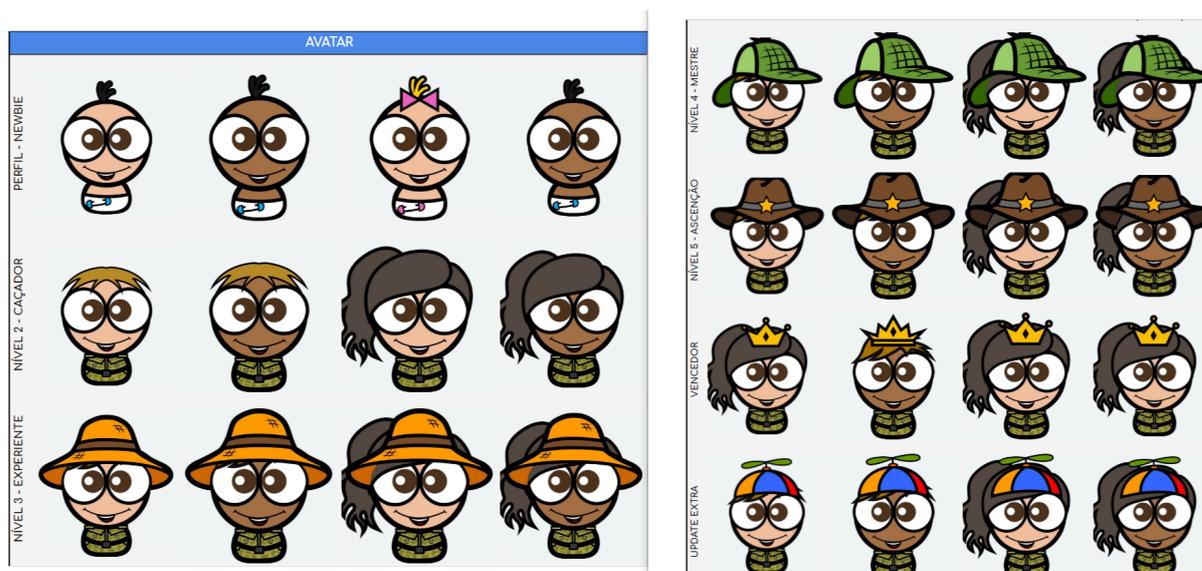


Figura 4.5: Ambiente gamificado - Aba Avatar.

Fonte: Elaborado pela autora

Por fim, conforme Figura 4.6, os bens virtuais que podem ser “comprados” pelos alunos estão presentes na aba *Loja*, enquanto a aba *Fórum*, apresentada na Figura 4.7, representa-se um espaço virtual no qual os alunos podem socializar e postar suas dúvidas, comentários ou respostas às perguntas que outros alunos tenham feito.

4.3.4 Material Experimental

Para responder as questões de pesquisa, analisou-se se o uso da abordagem gamificada para ensinar teste de software contribui para uma maior motivação e desempenho dos alunos quando comparado à abordagem convencional de ensino. Esta análise foi realizada com dados obtidos a partir da aplicação de duas métricas, sendo a primeira relacionada à motivação, e a segunda relacionada ao desempenho dos estudantes⁶.

⁶Por desempenho, considerou-se o aprendizado dos alunos, medido por teste de conhecimento através de pré e pós-teste, e *quizzes* abrangendo os conceitos de teste ensinados.

Loja Virtual		
Melhorias no Avatar	Descrição	Valor (\$)
Chapeu Estagiário	Torne-se um estagiário!	1
Outros Bens	Descrição	Valor (\$)
Exclusão de uma questão em um Quiz:	Compre o direito de excluir uma resposta incorreta em um Quiz.	7
Auxílio durante o desafio:	Compre o direito de ter um auxílio da instrutora durante o Desafio Final.	8

Figura 4.6: Ambiente gamificado - Aba Loja.

Fonte: Elaborado pela autora

FÓRUM			
Dúvida 1		Dúvida 2	
Aluno	Pergunta/Comentário	Aluno	Pergunta/Comentário

Figura 4.7: Ambiente gamificado - Aba Fórum.

Fonte: Elaborado pela autora

Não tendo a disponibilidade de um profissional da Psicologia para aferir a motivação dos alunos em todas as sessões experimentais, optou-se pela utilização do *Intrinsic Motivation Inventory* (IMI), um instrumento de medição multidimensional utilizado para verificar a experiência subjetiva dos participantes em relação a atividades experimentais. Este instrumento foi proposto (e aplicado) por Deci e Ryan (2011), e é composto por questões curtas com uma escala Likert de 1 a 7, sendo: (1) não é verdade, (4) um pouco verdade, e (7) verdade. Estas questões visam medir a motivação em quatro categorias, sendo: interesse/prazer, competência, escolha, e pressão/tensão percebidas. Embora o IMI tenha sido originalmente publicado em inglês (Apêndice B), neste trabalho foi utilizado a sua versão traduzida e validada para a língua portuguesa por Pedro (2016). O IMI é apresentado no Apêndice C deste documento.

Em relação à segunda métrica analisada, o desempenho dos alunos foi medido e acompanhado por meio da pontuação obtida em dois quizzes. O primeiro quiz abrange questões relacionadas aos conceitos básicos de teste de software e suas terminologias, tais como garantia da qualidade, processo, níveis e técnicas de teste. O segundo quiz, por sua vez, abrange tanto questões relacionadas aos conceitos básicos, quanto à técnica de teste funcional e seus critérios análise de valor limite, e particionamento

de equivalência. Todas as questões de ambos os *quizzes* foram retiradas dos slides de apresentação do conteúdo ensinado. Dessa forma, evitou-se cobrar dos alunos o conhecimento sobre um assunto não explicado.

Outras análises, além da motivação e do desempenho, foram realizadas neste trabalho. Primeiramente, um pré-teste⁷ contendo três perguntas foi aplicado visando verificar o conhecimento prévio dos alunos sobre teste de software. Após a intervenção (ensino dos conceitos de teste de software e aplicação de *quizzes* e atividade prática), foi aplicado o pós-teste⁸ para analisar se os alunos compreenderam qual é o objetivo do teste de software, sua importância, e o que é a técnica de teste funcional. O pré-teste e o pós-teste continham exatamente as mesmas perguntas e na mesma ordem.

Além disso, aplicou-se um exercício prático no qual os alunos deveriam criar e executar casos de teste baseando-se em três requisitos do *website e-commerce* da Casas Bahia⁹. Após o término do exercício, foram analisados quantos e quais critérios funcionais ensinados durante o experimento foram utilizados na criação dos casos de teste. Os requisitos propostos foram criados com o cuidado de estarem no mesmo nível dos que normalmente são apresentados em disciplinas de engenharia de software.

Por fim, um questionário para feedback da percepção dos alunos¹⁰ foi aplicado visando identificar se a abordagem utilizada colaborou na resolução ou minimização dos desafios enfrentados no ensino de teste de software, conforme apresentado na Tabela 4.1.

O conteúdo dos slides de apresentação utilizados durante as execuções experimentais é uma junção do conteúdo apresentado na fundamentação teórica apresentada no Capítulo 2 com uma adaptação dos slides utilizados pelo professor orientador deste trabalho nas disciplinas de engenharia de software que ministra.

Além das análises descritas anteriormente, foram observados os comentários, comportamentos e expressões dos alunos durante as sessões experimentais. Dessa forma, assim como as análises quantitativas, estas observações também serão apresentadas no Capítulo 5.

Destaca-se que todo material experimental utilizado encontra-se disponível em <<https://bit.ly/2NqKdUr>>.

⁷Pré-teste disponível em <<https://bit.ly/2Ku8ehZ>> - Acessado em 28 de novembro de 2019.

⁸Pós-teste disponível em <<https://bit.ly/32XaKn7>> - Acessado em 28 de novembro de 2019.

⁹*Website* da Casas Bahia disponível em <<https://www.casasbahia.com.br/>> - Acessado em 28 de novembro de 2019.

¹⁰Questionário de Avaliação do Minicurso disponível em <<https://bit.ly/37dCTKj>> - Acessado em 28 de novembro de 2019.

4.3.5 Experimento

Para a realização do experimento, iniciou-se uma busca por instituições de ensino superior nas quais professores de cursos voltados à computação pudessem dispor de um período de tempo para que seus alunos participassem de uma execução experimental. Em seguida, o comitê de ética da instituição de origem da pesquisadora (UFSCar) e das instituições participantes foram acionadas com a devida documentação preparada e submetida.

Participantes

Para realização desta pesquisa, foram convidados para participarem dos experimentos estudantes de cursos voltados à computação de quatro Instituições de Ensino Superior (IES) brasileiras, sendo elas: Universidade Federal do Espírito Santo - campus Alegre (IES A), Instituto Federal do Espírito Santo - campus Cachoeiro de Itapemirim (IES B), Instituto Federal do Espírito Santo - campus Alegre (IES C), e Universidade Federal de São Carlos - campus São Carlos (IES D)¹¹. Conforme mostrado na Tabela 4.4, a amostra contém 70 estudantes ao total, dos quais 13 pertencem à IES A, 18 à IES B, 22 à IES C e, 17 à IES D.

Tabela 4.4: Dados Demográficos.

Abordagem	IES	# Sujeitos	# considerados atividades	# considerados IMI	Curso
Tradicional	A	8	-	-	SI
Gamificada	A	5	-	-	SI
Gamificada	B	18	11	11	SI
Tradicional	C	22	22	22	ADS
Gamificada	D	17	16	16	CC/EC
TOTAL		70	61	60	

Fonte: Dados da pesquisa

Os participantes das IES A e B cursam Sistemas de Informação (SI); os alunos da IES C fazem parte do curso de Análise e Desenvolvimento de Sistemas (ADS), e na IES D a turma contém alunos tanto do curso de Ciência da Computação (CC) quanto de Engenharia da Computação (EC). Em todos os grupos, os participantes eram alunos do 2º ano em diante, ou seja, já tinham concluído pelo menos a metade da grade curricular. Nota-se que, como os dados extraídos das execuções do estudo piloto na IES A não foram analisados, a quantidade de participantes considerada em cada atividade experimental realizada nesta IES A não será descrita a seguir.

¹¹A fim de facilitar a leitura no decorrer da dissertação, as IES participantes do estudo serão mencionadas como A, B, C e D.

Para analisar o impacto da gamificação na motivação dos alunos, foram considerados apenas os participantes que responderam o questionário motivacional IMI. No total, foram 11 respondentes no grupo experimental da IES B, 22 no grupo de controle da IES C, e 16 alunos no grupo experimental da IES D.

Para a análise do impacto da gamificação no aprendizado, foram considerados apenas os alunos que participaram dos dois *quizzes* e do exercício final. No grupo experimental da IES B, 4 alunos não participaram da sessão completa (se ausentaram no primeiro ou segundo dia), e outros 3 não participaram do exercício final. Portanto, foram considerados 11 dos 18 alunos. Na IES C, todos os 22 alunos participaram de todas as atividades do experimento. Dessa forma, todos foram considerados para a análise dos dados. Por fim, na IES D, 1 aluno se ausentou na segunda parte da sessão, e todos os demais participaram de todas as atividades. Dessa forma, os 16 alunos restantes foram considerados.

Os participantes foram questionados se já tinham estudado teste de software durante o curso. De acordo com eles, alguns conceitos foram ensinados como um tópico na disciplina de engenharia de software na IES B, e na disciplina de garantia da qualidade na IES C. Os alunos da IES D ainda não tinham estudado sobre o assunto.

Nota-se que todos os participantes estavam cientes que sua participação era voluntária e não implicaria em qualquer punição caso deixassem o experimento. Além disso, tanto a pesquisadora que aplicou as sessões experimentais quanto os professores que convidaram os alunos destacaram os ganhos que eles poderiam obter em aprender a técnica de teste de software mais utilizada na indústria atualmente.

Metodologia

Para a realização das sessões experimentais foi seguido o processo experimental descrito por Wohlin et al. (2012). O *design* incluiu um fator (abordagem de ensino) e dois tratamentos (gamificado e convencional). Nesta perspectiva, as abordagens de ensino foram variáveis independentes que causaram algum efeito nos resultados (motivação e desempenho dos estudantes).

Na Figura 4.8 representa-se a metodologia seguida durante a execução das sessões experimentais. O estágio 1 é a fase operacional do experimento que abrange o convite para participação do estudo e preparação de todo o material experimental descrito na subseção 4.3.4. No segundo estágio, a pesquisadora apresentou a pesquisa aos participantes no início de cada sessão experimental. Em seguida, foram dadas as instruções para leitura e assinatura do Termo de Consentimento Livre e Esclarecido (TCLE). Após todos os alunos terem assinado o TCLE, o pré-teste foi aplicado e, após todos os alunos terem respondido, foi conduzido um tutorial sobre a plataforma *Bug Hunter* para o grupo experimental (este passo não foi realizado no grupo de controle dado que a

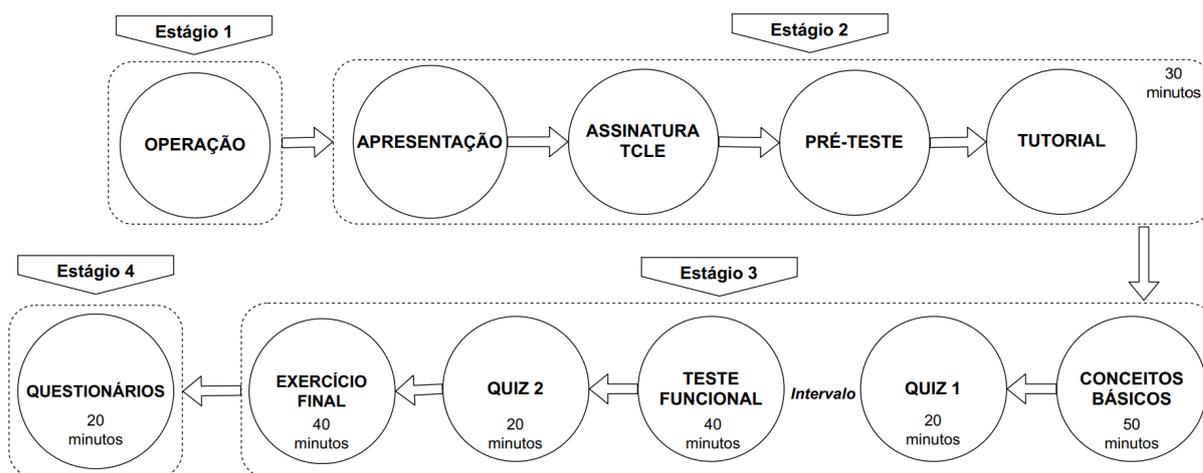


Figura 4.8: Abordagem Metodológica

Fonte: Elaborado pela autora

plataforma gamificada não foi utilizada na abordagem convencional). As atividades do segundo estágio duraram aproximadamente trinta minutos.

O terceiro estágio abrange a fase de intervenção, compreendendo o ensino de teste de software, a aplicação dos quizzes e um exercício final. O ensino dos conceitos básicos de teste teve duração de aproximadamente cinquenta minutos, onde foram ensinados a definição de termos relacionados a teste de software¹², assim como os níveis, técnicas, e fases do processo de teste. Imediatamente após o ensino dos conceitos básicos, um *quiz* contendo questões relacionadas ao que fora ensinado foi aplicado e teve duração de aproximadamente vinte minutos. Finalizado o primeiro *quiz*, os alunos tiveram um intervalo de vinte minutos (para os alunos da IES B o intervalo foi até o dia seguinte). Quando todos retornaram para a segunda parte do experimento, deu-se início ao ensino da técnica de teste funcional e seus critérios particionamento de equivalência e análise de valor limite. No decorrer das explicações, os alunos foram encorajados a participarem ativamente e, juntos com a pesquisadora, ajudarem a resolver os exemplos apresentados. Esta fase de ensino durou aproximadamente quarenta minutos. Em seguida, o segundo *quiz* foi aplicado contendo tanto questões sobre os conceitos básicos quanto sobre a técnica e os critérios do teste funcional. A aplicação do segundo *quiz* também durou aproximadamente vinte minutos.

Para a aplicação dos dois *quizzes* foi utilizada a ferramenta gamificada *Kahoot!*¹³ no grupo experimental, enquanto no grupo de controle foi utilizada a ferramenta não gamificada *Google Forms*¹⁴. Destaca-se que as mesmas questões e opções de

¹² engano, defeito, erro, falha, domínio de entrada, dado de teste, caso de teste, oráculo de teste, suíte de teste, etc.

¹³Disponível em <<https://kahoot.com/>> - Acessado em 28 de novembro de 2019.

¹⁴Disponível em <<https://google.com/forms/about/>> - Acessado em 28 de novembro de 2019.

respostas foram aplicadas em ambos os grupos, diferenciando apenas as ferramentas utilizadas para a aplicação.

Imediatamente após o segundo *quiz*, os alunos se organizaram em grupos de 3 a 4 membros para a realização da atividade final, última fase do estágio de intervenção que durou aproximadamente quarenta minutos. No início da atividade, foram apresentados três requisitos funcionais do *website e-commerce* da Casas Bahia¹⁵, sendo eles:

1. O usuário deve poder realizar login no sistema
2. O usuário deve poder adicionar itens no carrinho de compras
3. O usuário deve poder editar os itens do carrinho de compras

Em seguida, os alunos foram orientados a criar e executar seus casos de teste baseando-se nestes três requisitos funcionais apresentados. Além disso, foi solicitado que os grupos entregassem os casos de teste criados para que fosse possível analisar quantos dos critérios funcionais ensinados durante o experimento foram utilizados para a sua criação.

No quarto e último estágio, após finalizada a atividade prática, os participantes responderam quatro questionários, sendo eles: (1) pós-teste (contendo as mesmas questões do pré-teste), (2) *Intrinsic Motivation Inventory* (IMI), (3) Avaliação do Mini-curso, e (4) Avaliação da Plataforma *Bug Hunter* (este último questionário foi aplicado apenas no grupo experimental). Este estágio durou aproximadamente vinte minutos. Os dados extraídos durante as sessões experimentais realizadas nas IES *B*, *C* e *D* foram analisadas e os resultados serão apresentados no capítulo a seguir.

4.4 Considerações Finais

Embora os autores dos estudos selecionados apresentados no Capítulo 3 tenham reportado resultados positivos em suas pesquisas, o uso da gamificação nem sempre é suficiente para resolver os problemas tratados. Além disso, percebeu-se na prática que o desenvolvimento de uma abordagem gamificada não é uma tarefa fácil, especialmente pela atual carência de pesquisas e ferramentas que apoiem o processo de gamificação de um ambiente.

As análises dos dados obtidos através das sessões experimentais realizadas serão apresentadas no capítulo seguinte, assim como a discussão dos resultados.

¹⁵Disponível em <<https://casasbahia.com.br/>> - Acessado em 28 de novembro de 2019.

Capítulo 5

ANÁLISE DOS DADOS

5.1 Considerações Iniciais

Conforme apresentado no Capítulo 4, este projeto de mestrado possui natureza exploratória e experimental. Considerando a natureza exploratória, um dos passos que a compõe é a realização de um estudo piloto com propósito de refinamentos. Dentro da natureza experimental, por sua vez, um estudo experimental foi planejado baseado no processo de Wohlin et al. (2012).

No início deste capítulo são apresentados o estudo piloto e as lições aprendidas (Seção 5.2). Em seguida, apresentam-se os detalhes e resultados do experimento realizado (Seção 5.3). Na Seção 5.4 revisitam-se as questões de pesquisa e discutem-se os resultados.

Destaca-se que todas as execuções foram conduzidas pela mesma pesquisadora que, no início de cada sessão, deixou claro aos alunos sobre o aspecto voluntário da participação na pesquisa. Além disso, os resultados parciais do experimento também se encontram em um artigo publicado no XVIII Simpósio Brasileiro de Qualidade de Software (SBQS'19)¹ (Jesus et al., 2019).

5.2 Estudo Piloto

O estudo piloto foi conduzido em setembro de 2018 com dois grupos da IES A. Duas execuções foram realizadas, cada uma com aproximadamente cinco horas de duração e fora do horário ordinário de aula. Na primeira execução utilizou-se a abordagem convencional com um grupo de controle contendo 8 alunos. Na segunda execução, por sua vez, utilizou-se a abordagem gamificada com o grupo experimental contendo 5

¹Disponível em <<http://sbqs.sbc.org.br/index.php/pt/anais>> - Acessado em 28 de novembro de 2019.

alunos, inicialmente. Na segunda parte da sessão um participante precisou se ausentar, permanecendo os outros 4 alunos até o final.

Essas sessões piloto tiveram propósitos de refinamento em relação ao conteúdo ensinado, à plataforma gamificada desenvolvida para dar suporte ao grupo experimental, aos questionários, *quizzes*, à atividade prática, e ao tempo de execução do experimento. A abordagem metodológica seguida foi a mesma das execuções experimentais, conforme apresentado na subseção 4.3.5.

5.2.1 Lições Aprendidas

Em relação ao conteúdo ensinado, percebeu-se um excesso de informação apresentada em um curto período de tempo, fazendo com que os alunos se confundissem com os termos e nomes dos critérios de cada técnica de teste de software. Dessa forma, decidiu-se por apenas citar (sem explicar) as técnicas de teste estrutural e teste baseado em defeitos, focando a explicação mais detalhada na técnica de teste funcional e seus critérios particionamento equivalência, e análise de valor limite. Além disso, também foi reduzido em uma hora o tempo de execução de cada sessão experimental.

Os *quizzes* foram refinados para estarem mais alinhados e coerentes com o conteúdo ensinado. Já na atividade prática, ao invés de permitir que os próprios alunos optassem por um *website e-commerce* de sua preferência, como ocorreu nas sessões piloto, definiu-se o *website* da Casas Bahia² para todos os grupos testarem com a suíte de teste criada durante a atividade. Os questionários de avaliação do minicurso e da plataforma, por sua vez, também foram refinados para abrangerem questões relacionadas aos problemas enfrentados no ensino de teste de software, viabilizando analisar se o uso da gamificação colaborou em minimizá-los.

Por fim, em relação à plataforma gamificada, percebeu-se que é exigido uma maior atenção da pesquisadora para observar os alunos que se mantiveram atentos e participativos ao longo da aula. Além disso, notou-se uma certa dificuldade por parte da pesquisadora em manipulá-la atribuindo todas as recompensas manualmente enquanto ensinava o conteúdo. Neste sentido, o uso da plataforma seria inviável em turmas com grande quantidade de alunos. Visando reduzir este esforço, o ambiente gamificado foi automatizado de forma que os *selos*, *moedas*, *melhorias no avatar*, e *mudanças de nível* fossem calculados e atribuídos a partir de uma única entrada manual: os *XP*s.

²Disponível em <<https://www.casasbahia.com.br/>> - Acessado em 28 de novembro de 2019.

5.3 Experimento

Após a execução das duas sessões piloto e realização dos refinamentos necessários, deu-se início às execuções experimentais para extração e análise dos dados. O *design* do experimento inclui um fator (*abordagem de ensino*) e dois tratamentos (*convencional e gamificado*). Nesta perspectiva, as abordagens de ensino foram variáveis independentes que causaram algum efeito nas variáveis dependentes (*motivação e desempenho*) avaliadas. O experimento foi definido seguindo o processo experimental proposto por Wohlin et al. (2012) e é sumarizado da seguinte forma:

Analisar a abordagem gamificada **com o propósito de** verificar o seu impacto **com respeito** à motivação e ao desempenho dos alunos **do ponto de vista** da pesquisadora, **no contexto** brasileiro de ensino de teste de software a alunos de computação em nível de graduação.

A análise de causa e efeito foi baseada nas questões de pesquisa (**QP1 e QP2**) levantadas neste trabalho. Para responder cada questão, foram formuladas hipóteses nulas (**H1.0 e H2.0**), conforme apresentado na Tabela 5.1.

Tabela 5.1: Questões de Pesquisa e Hipóteses

Questão de Pesquisa	Hipótese Nula
<p>QP1. A motivação dos alunos que aprendem teste de software por meio de uma abordagem gamificada é maior do que a motivação dos alunos que aprendem por meio da abordagem convencional?</p>	<p>H1.0 A motivação dos alunos que aprendem teste de software por meio de uma abordagem gamificada não é diferente da motivação dos alunos que aprendem por meio da abordagem convencional.</p>
<p>QP2. O desempenho dos alunos que aprendem teste de software por meio de uma abordagem gamificada é maior do que o desempenho dos alunos que aprendem por meio da abordagem convencional?</p>	<p>H2.0 O desempenho dos alunos que aprendem teste de software por meio de uma abordagem gamificada não é diferente do desempenho dos alunos que aprendem por meio da abordagem convencional.</p>

Fonte: Elaborado pela autora

5.3.1 Design e Procedimento

O experimento abrange duas execuções e uma replicação. A primeira execução ocorreu com um grupo experimental na IES *B*. Um professor do curso de Sistemas de Informação desta IES convidou os alunos de sua disciplina para participarem e forneceu duas aulas em dois dias seguidos, totalizando aproximadamente quatro horas. Ao todo participaram 18 alunos, porém, conforme explicado na subseção 4.3.5, apenas 11 foram considerados para análise dos dados.

A segunda execução foi realizada utilizando a abordagem convencional de ensino com um grupo de controle da IES *C*. O convite para participação foi feito pelo coordenador do curso de Análise e Desenvolvimento de Sistemas. Esta execução também teve duração de quatro horas, porém ocorreu em uma única tarde fora do horário ordinário de aula. Ao todo participaram 22 alunos, e todos foram considerados para análise dos dados.

Por fim, a replicação ocorreu na IES *D* com um novo grupo experimental. O objetivo desta replicação foi aumentar a amostra do grupo experimental e possibilitar duas novas comparações, sendo uma entre os resultados do grupo de controle com este novo grupo experimental, e outra entre o grupo de controle com a junção dos dois grupos experimentais. A execução da replicação teve quatro horas de duração e ocorreu durante o horário de aula na IES *D* a convite de um professor da disciplina de Engenharia de Software, composta por alunos dos cursos Engenharia da Computação e Ciência da Computação. A abordagem gamificada foi utilizada e, ao todo, foram considerados 16 alunos para a análise dos dados.

Além disso, todo o material experimental descrito na subseção 4.3.4 foi devidamente preparado antes de cada execução. Em todas as três execuções foi seguida a abordagem metodológica contendo quatro estágios, conforme representados na Figura 4.8.

5.3.2 Análise Quantitativa – Observações Gerais

Nesta seção apresentam-se os detalhes das análises quantitativas dos dados e os resultados do experimento. Inicialmente foram realizadas análises de medidas por meio de estatística descritiva (por exemplo, média, mediana e variância). Além disso, o teste Shapiro-Wilk foi aplicado para verificar se os dados possuíam distribuição normal, considerando um intervalo de confiança $\alpha = 0,05$. Constatando-se a distribuição não normal dos dados, aplicou-se o teste não paramétrico de Mann-Whitney, uma vez que a configuração do experimento envolveu categorias independentes (um grupo de controle e outro experimental), um fator, e dois tratamentos.

Os testes de conhecimento (pré- e pós-teste) foram aplicados para apoiar a análise do desempenho dos alunos. Em seguida calculou-se o valor Δ , que representa o incremento do conhecimento observado imediatamente após as instruções utilizando-se determinada abordagem de ensino (nota-se que uma abordagem similar foi adotada em estudos anteriores, como o de Lyra et al. (2016), como uma forma de medir o quanto um aluno aprendeu após uma aula). O valor Δ foi calculado utilizando-se a seguinte fórmula,

$$\Delta = X_{(i)} - Y_{(i)}$$

onde:

X = Número de acertos no pós-teste;

Y = Número de acertos no pré-teste;

$i = \{\text{aluno}_1, \text{aluno}_2, \text{aluno}_3, \dots, \text{aluno}_n\}$.

Para complementar a análise de desempenho dos alunos, foram aplicados dois *quizzes*, sendo o primeiro imediatamente após o ensino dos conceitos básicos de teste, e o segundo imediatamente após o ensino da técnica de teste funcional. Os *quizzes* foram corrigidos e a eficácia dos alunos foi calculada considerando o número máximo de respostas corretas através da seguinte fórmula:

$$Eficacia_{(i)} = \frac{n_{(i)}}{TOTAL}$$

onde:

n = Número de respostas corretas do aluno;

$i = \{\text{aluno}_1, \text{aluno}_2, \text{aluno}_3, \dots, \text{aluno}_n\}$;

$TOTAL$ = Número total de questões no *quiz*.

Em relação à motivação dos alunos, aplicou-se em ambos os grupos (controle e experimental) o questionário *Intrinsic Motivation Inventory* (IMI) proposto por Deci e Ryan (2011), considerando-se quatro aspectos motivacionais, sendo eles: (a) *Interesse/diversão*, (b) *Escolha percebida*, (c) *Competência percebida*, e (d) *Pressão/Tensão*. Este questionário contém 22 questões curtas com uma escala Likert de 7 pontos, e cada aspecto é avaliado por um conjunto destas questões (mais detalhes no Apêndice B). Para medir cada aspecto, calculou-se inicialmente a média das respostas de cada questão, seguido da soma das médias das questões que compõem cada conjunto. Para exemplificar este cálculo, considerando-se que 10 alunos tenham respondido o questionário IMI e que as questões de número 1, 5, 8, 10, 14, 17 e 20 compõem o conjunto de questões relacionadas ao aspecto motivacional *Interesse/diversão*, para se obter o resultado deste primeiro aspecto, procede-se da seguinte forma: calcula-se a média das dez respostas atribuídas à questão 1, em seguida a média das dez respostas da questão 5, a média das dez respostas da questão 8, e assim por diante,

até a última questão (20) das que compõem este conjunto. Em seguida, somam-se estas sete médias e o resultado é o valor atribuído ao aspecto *Interesse/diversão*. O mesmo processo deve ser seguido para obter os resultados dos demais aspectos.

Por fim, analisou-se o questionário³ que visava coletar o feedback dos alunos com respeito às duas abordagens de ensino. Uma escala Likert com 7 opções de resposta foi utilizada para sumarizar a análise das dez questões presentes no questionário; o menor valor da escala (= 1) significa *discordo totalmente*, e o maior valor (= 7) significa *concordo totalmente*. As respostas de cada grupo (controle e experimental) foram avaliadas através da seguinte fórmula,

$$Result = \sum_{i=1}^N x_{(i)}$$

onde:

x = A resposta que o aluno atribuiu à questão, isto é $x = \{1, 2, \dots, 7\}$;

N = Número total de resposta obtidas em cada uma das opções de resposta;

i = {aluno ₁, aluno ₂, aluno ₃, ..., aluno _n};

A seguir são apresentados os resultados das análises realizadas.

5.3.3 Análise Quantitativa 1 - (GC vs GE1)

Nesta primeira análise foram comparados os resultados das execuções experimentais realizadas com o grupo de controle (GC) da IES C com os resultados do grupo experimental (GE1) da IES B.

A primeira variável dependente analisada foi a motivação dos 22 alunos do grupo de controle e 11 alunos do grupo experimental. Na Figura 5.1⁴ são representados os resultados do questionário *Intrinsic Motivation Inventory* (IMI) aplicado aos alunos de ambos os grupos. O grupo experimental obteve uma pontuação maior do que o grupo de controle nos aspectos *Interesse/diversão* (44,36) e *Escolha percebida* (30,64). Por outro lado, os alunos do grupo de controle emitiram uma pontuação maior do que o grupo experimental nos aspectos *Competência percebida* (27,41) e *Pressão/tensão* (12,45). Dessa forma, acredita-se que o grupo experimental tenha se interessado e se divertido mais ao aprender o conteúdo de teste de software por meio da abordagem gamificada do que o grupo de controle com a abordagem convencional. Além disso, embora os alunos do grupo de controle tenham se sentido mais pressionados e tensos durante a aula, também houve uma maior auto-percepção de competência do que no

³Disponível em <<https://bit.ly/35NqDyk>> - Acessado em dezembro de 2019

⁴Ressalta-se que, por questões de consistência entre documentos de trabalho que estão em elaboração pelo grupo de pesquisa no laboratório LaPES, os rótulos de dados exibidos nos gráficos deste capítulo foram mantidos em língua inglesa. A sigla EG representa um grupo experimental (isto é, GE mencionado no texto), e a sigla CG representa o grupo de control (isto é, GC mencionado no texto).

grupo experimental para a realização das atividades, o que é confirmado nos resultados da análise de desempenho.

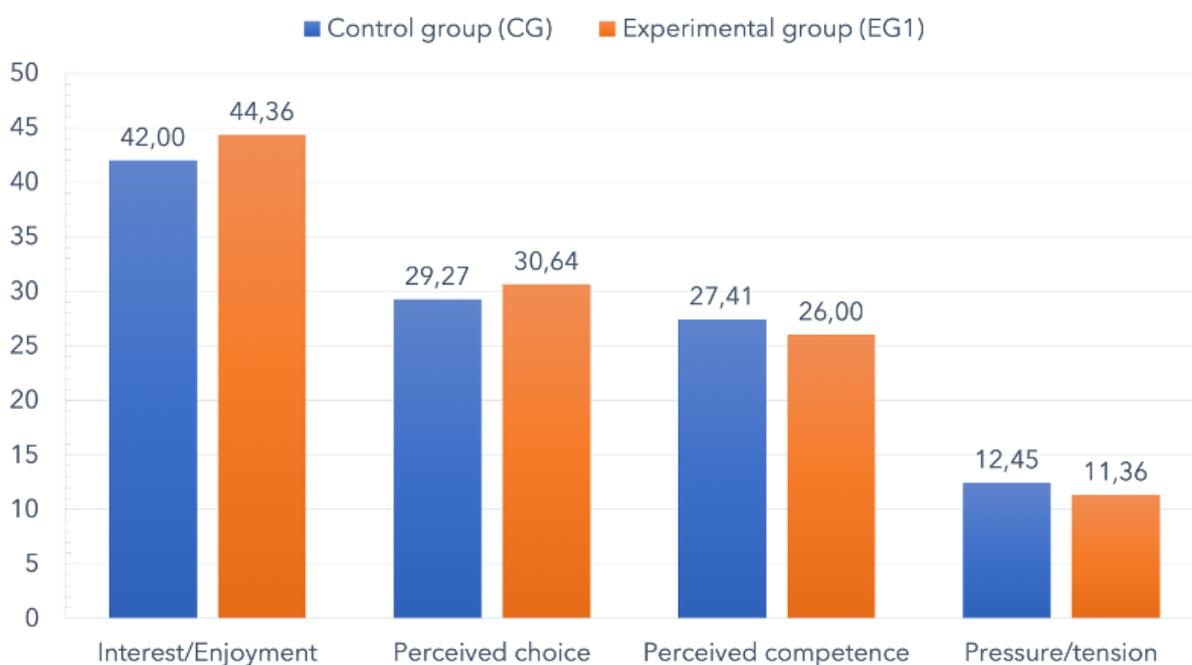


Figura 5.1: Resultados da Motivação - GC vs GE1

Fonte: Dados da pesquisa

A segunda variável dependente analisada foi o desempenho dos alunos. Com base nas correções dos testes de conhecimento (pré- e pós-teste), notou-se que o grupo de controle (GC) obteve pontuações inferiores à pontuação do grupo experimental (GE1) no pré-teste. Conforme representado na Figura 5.2, enquanto a pontuação máxima do grupo de controle foi de 2 pontos, no grupo experimental alguns alunos atingiram 3 pontos, ou seja, acertaram todas as questões. Após o ensino da técnica de teste funcional aplicou-se o pós-teste, onde os grupos obtiveram resultados similares. A menor pontuação em cada grupo foi de 1 ponto, enquanto a maior foi de 3 pontos (a pontuação máxima do teste de conhecimento). A mediana para ambos foi igual a 2. Estes resultados sugerem que alunos de ambos os grupos atingiram níveis similares de conhecimento, e que a abordagem convencional promoveu uma maior aprendizagem.

Para averiguar qual abordagem de ensino promoveu resultados melhores em relação à eficácia na aprendizagem dos alunos, calculou-se o valor de Δ . Conforme observado na Figura 5.3, nota-se uma maior variação entre os valores de Δ no grupo controle e, assim, nenhuma tendência para um valor particular. Os valores mínimo e máximo no pré-teste e pós-teste para o grupo de controle foram 0 e 2, e 1 e 3, respectivamente. A mediana demonstra que o grupo de controle obteve valores Δ superiores que o grupo experimental.

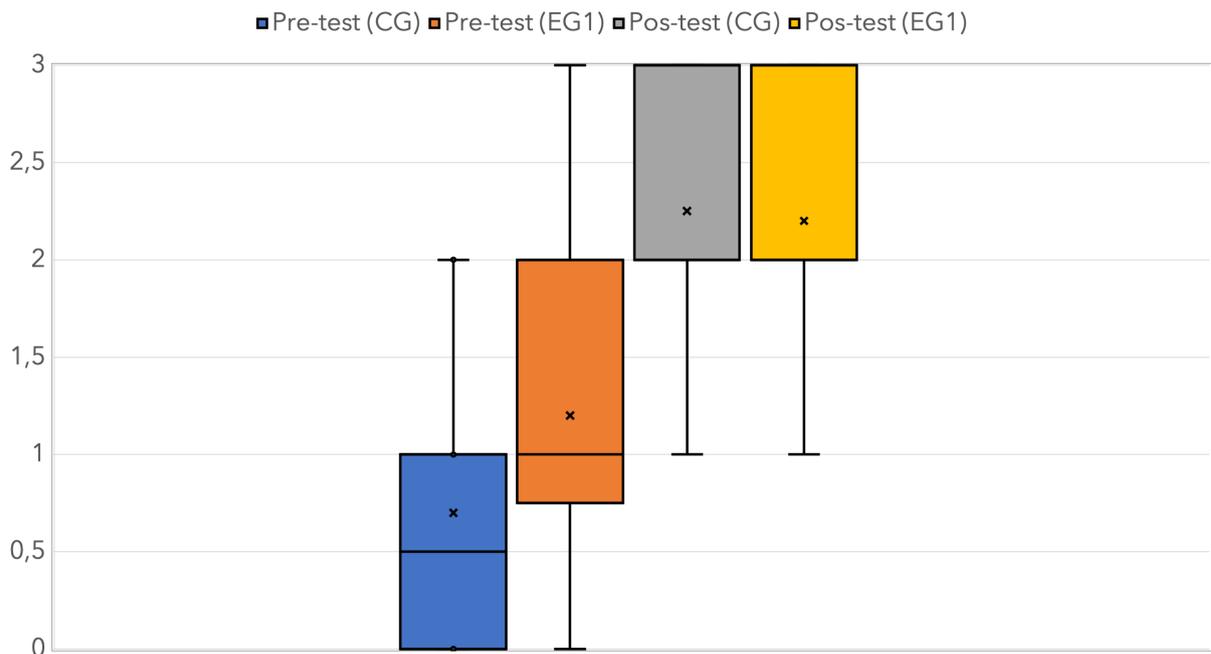


Figura 5.2: Resultado dos testes de conhecimento - GC vs GE1

Fonte: Dados da pesquisa

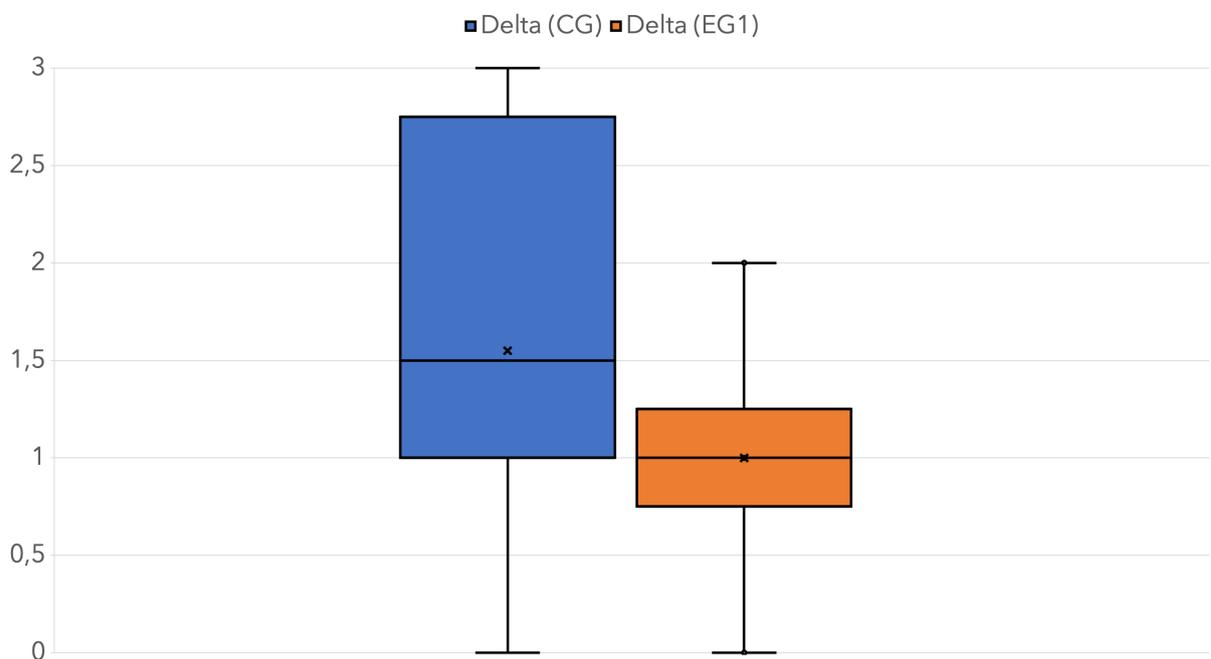


Figura 5.3: Boxplot com os valores Δ s dos testes de conhecimento - GC vs GE1

Fonte: Dados da pesquisa

Na verificação da normalidade na distribuição dos dados, constatou-se que tanto o grupo de controle (GC) quanto o grupo experimental (GE1) possuem distribuição não gaussiana, uma vez que $p - value < 0,05$ em ambos os grupos, conforme representado

na Figura 5.4. Dessa forma, embora o cálculo da mediana reforce o resultado de que o grupo experimental tenha obtido pontuações superiores, o teste de Mann-Whitney foi aplicado resultando em $p = 0,1010$, conforme representado na Figura 5.5, o que indica que (estatisticamente) não há diferença com significância estatística entre os grupos GC e GE1 em relação ao nível de aprendizagem quando a abordagem adotada é a convencional ou a gamificada.

Resultados		- 1 -
Tamanho da amostra =	20	
Média =	1.5500	
Desvio padrão =	1.0990	
W =	0.8736	
p =	0.0135	

GRUPO DE CONTROLE (GC)

Resultados		- 1 -
Tamanho da amostra =	10	
Média =	1.0000	
Desvio padrão =	0.6667	
W =	0.8154	
p =	0.0278	

GRUPO EXPERIMENTAL (GE1)

Figura 5.4: Resultado do teste de normalidade da distribuição dos dados no pré- e pós-teste - GC e GE1

Fonte: Dados da pesquisa

Resultado	Amostra 1	Amostra 2
Tamanho da amostra	20	10
Soma dos Postos (Ri)	339.0	126.0
Mediana =	1.50	1.00
U =	71.00	
Z(U) =	1.2758	
p-valor (unilateral) =	0.1010	
p-valor (bilateral) =	0.2020	

Figura 5.5: Teste de Mann-Whitney para o pré- e pós-teste - GC vs. GE1

Fonte: Dados da pesquisa

Além do pré- e pós-teste, uma segunda análise do conhecimento adquirido ao longo da sessão experimental foi realizada através de dois *quizzes*, sendo o primeiro aplicado

após o ensino dos conceitos básicos de teste, e o segundo após o ensino da técnica de teste funcional. Nesta análise observou-se a eficácia dos alunos em responder corretamente as questões. Assim, é possível comparar o desempenho dos grupos de controle e experimental em cada conteúdo ensinado (ou seja, conceitos básicos de teste de software e os critérios do teste funcional).

O *box-plot* representado da Figura 5.6 demonstra a eficácia dos alunos de todos os grupos (controle e experimentais) em responder corretamente as questões sobre os conceitos básicos de teste de software no *quiz 1*. Comparando os resultados do grupo de controle e do grupo experimental GE1, pode-se notar que o (GE1) obteve uma mediana de 70% de eficácia, enquanto a mediana do grupo controle (GC) foi de 90%. Em relação aos valores mínimos, ambos os grupos obtiveram 50% de eficácia. Por outro lado, o valor máximo obtido no grupo controle foi de 100% de eficácia, contra 90% no grupo experimental.

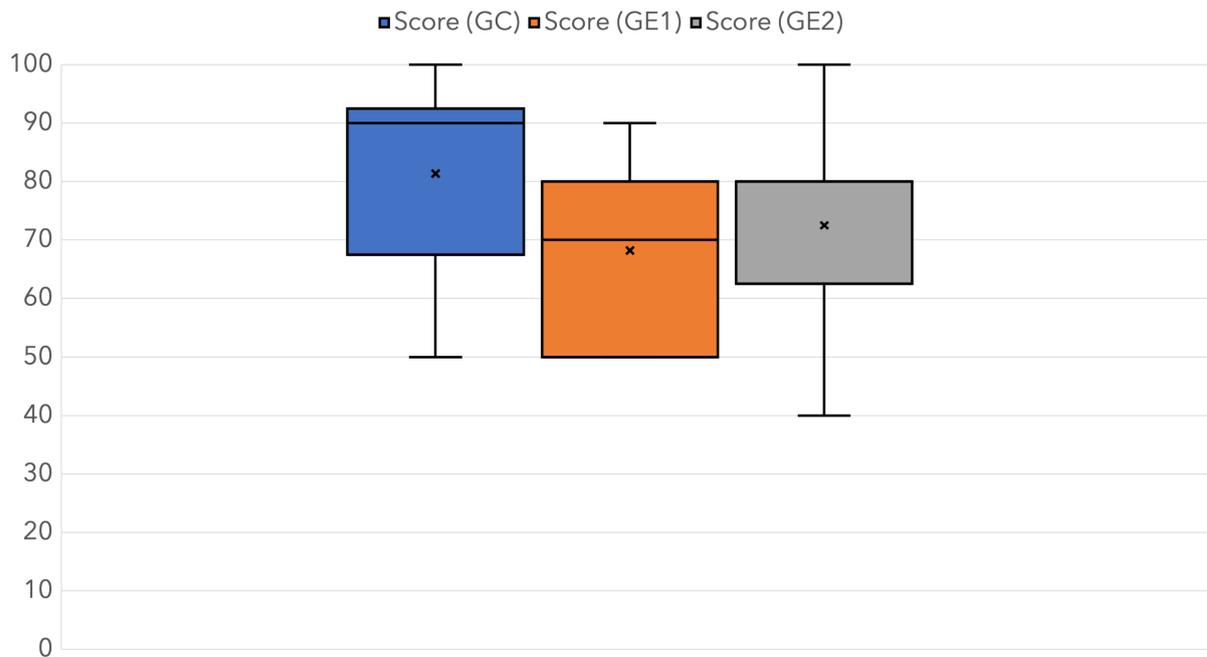


Figura 5.6: Eficácia dos alunos no *quiz 1* - GC, GE1, GE2

Fonte: Dados da pesquisa

Na verificação da normalidade na distribuição dos dados, constatou-se que o grupo de controle (GC) possui uma distribuição não gaussiana ($p - value = 0,0124$), enquanto o grupo experimental (GE1) possui distribuição gaussiana ($p - value = 0,2361$), conforme representado na Figura 5.7. Como um dos grupos apresentou distribuição não normal, aplicou-se o teste não paramétrico de Mann-Whitney que revelou uma diferença com significância estatística entre o valor da eficácia de cada grupo, uma vez que $p - value = 0,0155$, conforme representado na Figura 5.8. Considerando esta perspectiva, conclui-se que é possível inferir que os alunos que aprenderam os

conceitos básicos de teste de software por meio da abordagem convencional foram mais eficazes em responder corretamente as questões do *quiz 1* do que os alunos que aprenderam por meio da abordagem gamificada.

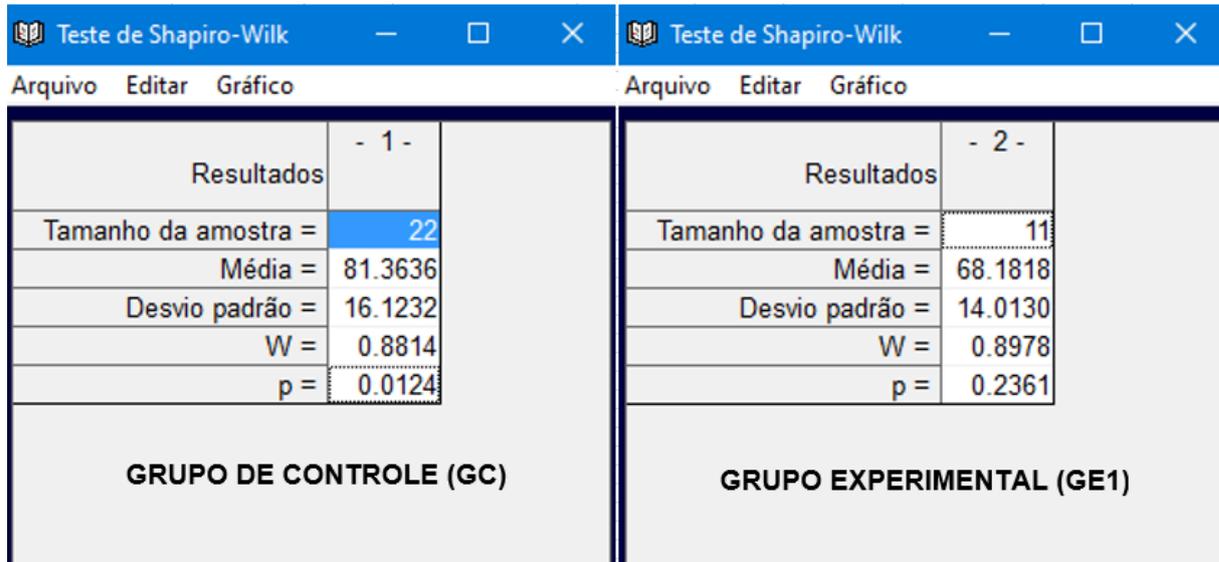


Figura 5.7: Resultado do teste de normalidade da distribuição dos dados no Quiz 1 - GC e GE1

Fonte: Dados da pesquisa

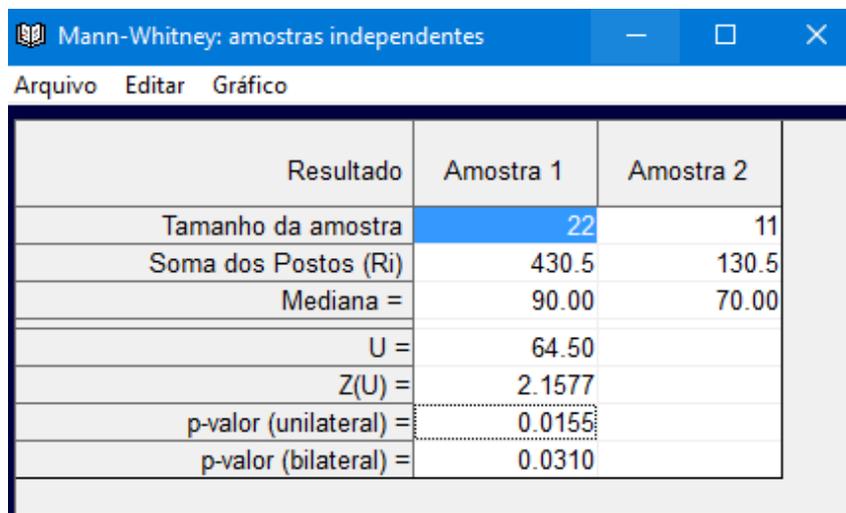


Figura 5.8: Teste de Mann-Whitney para o Quiz 1 - GC vs. GE1

Fonte: Dados da pesquisa

O segundo *quiz* foi aplicado após o ensino dos critérios do teste funcional. Similar aos resultados do primeiro *quiz*, o grupo de controle obteve resultados melhores (e maior eficácia) que o grupo experimental, conforme representado na Figura 5.9 A mediana para o grupo de controle foi 80%, contra 70% no grupo experimental. Nesta

análise, observa-se um *outlier* no grupo controle, o que pode indicar que um aluno não compreendeu os critérios do teste funcional satisfatoriamente.

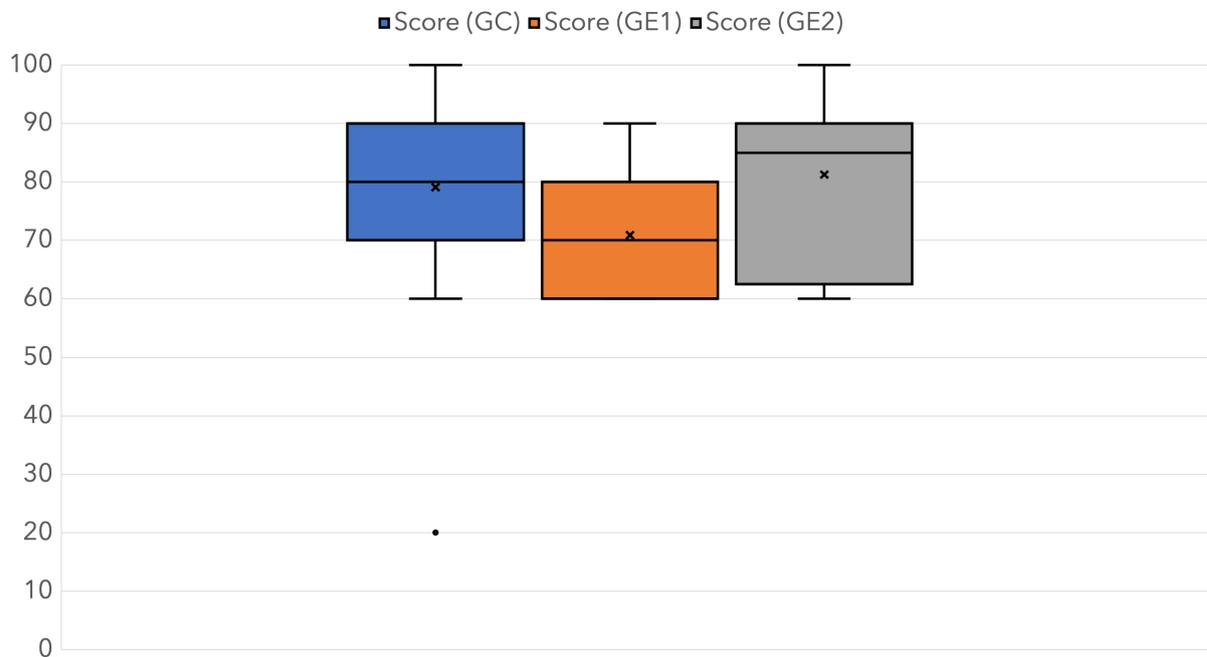


Figura 5.9: Eficácia dos alunos no *quiz 2* - GC, GE1, GE2

Fonte: Dados da pesquisa

Na verificação da normalidade na distribuição dos dados, constatou-se que o grupo de controle (GC) possui uma distribuição não gaussiana (p -value = 0,0085), enquanto o grupo experimental (GE1) possui distribuição gaussiana (p -value = 0,0889), conforme representado na Figura 5.10. Como um dos grupos apresentou distribuição não normal, aplicou-se o teste não paramétrico de Mann-Whitney que revelou uma diferença com significância estatística entre o valor da eficácia de cada grupo (p -value = 0,0140), conforme representado na Figura 5.11. Considerando esta perspectiva, assim como na análise do primeiro *quiz*, é possível inferir que os alunos que aprenderam os critérios particionamento de equivalência e análise de valor limite por meio da abordagem convencional foram mais eficazes em responder corretamente as questões do *quiz 2* do que os alunos que aprenderam por meio da abordagem gamificada.

The figure shows two side-by-side screenshots of the Shapiro-Wilk test results window. The left window is for 'GRUPO DE CONTROLE (GC)' and the right is for 'GRUPO EXPERIMENTAL (GE1)'. Both windows show a table of results including sample size, mean, standard deviation, W statistic, and p-value.

- 1 -		- 2 -	
Resultados		Resultados	
Tamanho da amostra =	22	Tamanho da amostra =	11
Média =	79.0909	Média =	70.9091
Desvio padrão =	16.3034	Desvio padrão =	10.4447
W =	0.7488	W =	0.8702
p =	0.0085	p =	0.0889
GRUPO DE CONTROLE (GC)		GRUPO EXPERIMENTAL (GE1)	

Figura 5.10: Resultado do teste de normalidade da distribuição dos dados no Quiz 2 - GC e GE1

Fonte: Dados da pesquisa

The figure shows a screenshot of the Mann-Whitney test results window. The window title is 'Mann-Whitney: amostras independentes'. It displays a table comparing two samples (Amostra 1 and Amostra 2) across various statistical measures.

Resultado	Amostra 1	Amostra 2
Tamanho da amostra	22	11
Soma dos Postos (Ri)	431.5	129.5
Mediana =	80.00	70.00
U =	63.50	
Z(U) =	2.1959	
p-valor (unilateral) =	0.0140	
p-valor (bilateral) =	0.0281	

Figura 5.11: Teste de Mann-Whitney para o Quiz 2 - GC vs. GE1

Fonte: Dados da pesquisa

5.3.4 Análise Quantitativa 2 - (GC vs GE2)

Nesta segunda análise foram comparados os resultados das execuções experimentais realizadas com o grupo de controle (GC) na IES C com os resultados do grupo experimental (GE2) da IES D.

Na Figura 5.12 são representados os resultados do questionário *Intrinsic Motivation Inventory* (IMI) aplicado aos alunos de ambos os grupos, sendo considerados 22 alunos do grupo de controle e 16 alunos do grupo experimental. O grupo experimental (GE2) obteve uma pontuação menor do que o grupo de controle (GC) nas categorias motivacionais positivas, sendo *Interesse/diversão*, *Escolha percebida* e *Competência percebida*. Por outro lado, na categoria *Pressão/tensão*, os alunos do grupo de controle emitiram uma pontuação menor do que o grupo experimental. Dessa forma, acredita-se que o grupo experimental tenha se sentido menos livre e competente, além de ter se interessado/divertido menos ao aprender o conteúdo de teste de software por meio da abordagem gamificada do que o grupo de controle com a abordagem convencional. Além disso, o resultado também sugere que o grupo experimental se sentiu mais pressionado e tenso durante a aula do que o grupo de controle.

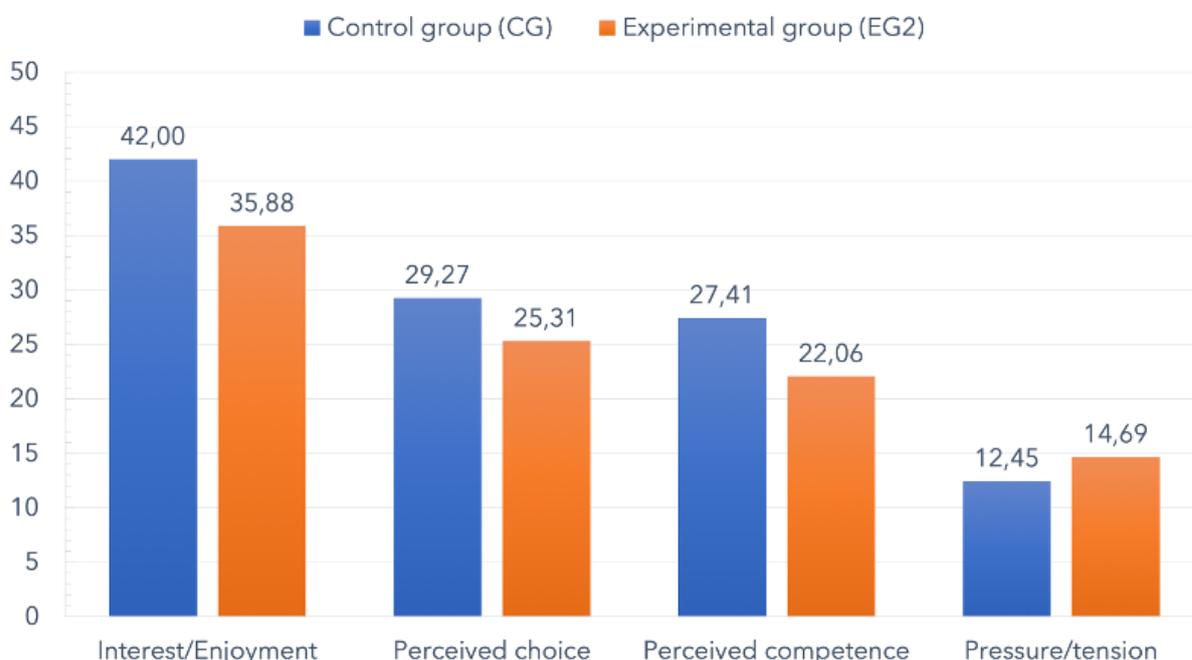


Figura 5.12: Resultados da Motivação - GC vs GE2

Fonte: Dados da pesquisa

Em relação ao desempenho dos alunos, conforme representado na Figura 5.13, ambos os grupos de controle (GC) e experimental (GE2) obtiveram a mesma pontuação mínima e máxima no pré-teste, sendo 0 e 2 respectivamente. No pós-teste, ambos os grupos obtiveram nota máxima igual a 3 pontos, porém o grupo experimental (GE2)

obteve nota mínima superior ao grupo de controle (GC) em 1 ponto. A mediana do grupo experimental (GE2) também foi superior à do grupo de controle (GC), sendo 1 e 0,5 no pré-teste, e 3 e 2 no pós-teste, respectivamente. Estes resultados sugerem que os alunos do grupo experimental (GE2) atingiram um nível de conhecimento superior ao grupo de controle (GC), e que a abordagem gamificada promoveu uma maior aprendizagem.

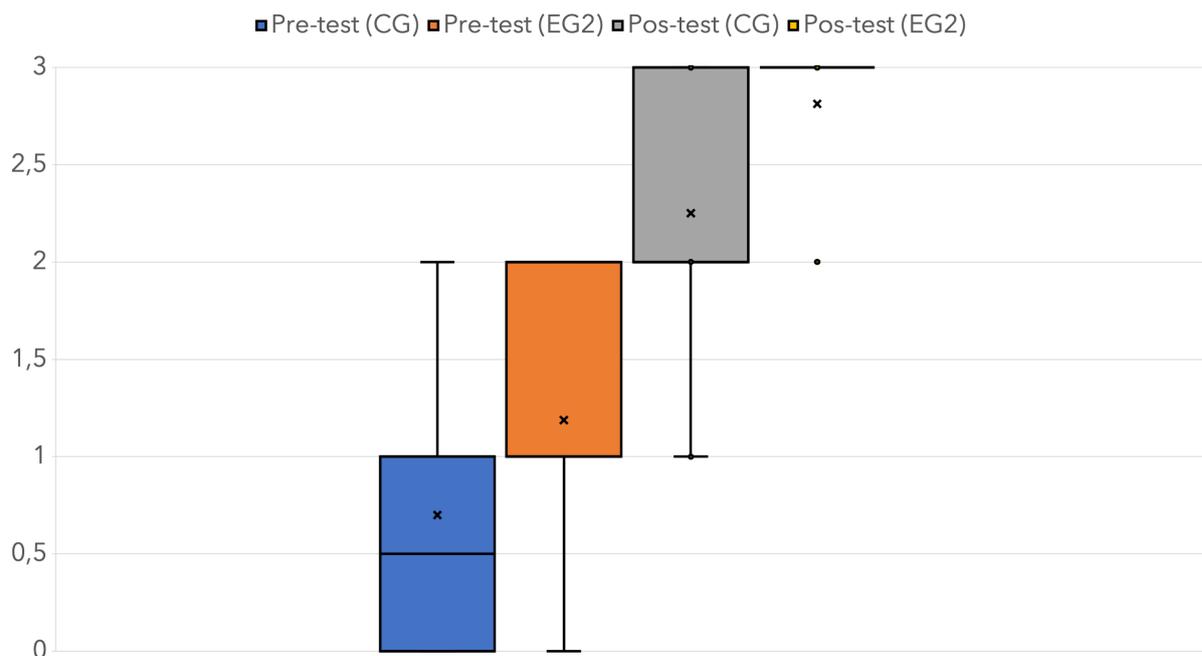


Figura 5.13: Resultado dos testes de conhecimento - GC vs GE2

Fonte: Dados da pesquisa

Para averiguar qual abordagem de ensino promoveu resultados melhores em relação à eficácia na aprendizagem dos alunos, calculou-se o valor de Δ . Conforme observado na Figura 5.14, nota-se uma variação similar entre os valores de Δ em ambos os grupos, e, assim, nenhuma tendência para um valor particular. A mediana demonstra que o grupo experimental (GE2) obteve valores Δ superiores que o grupo de controle (GC), sendo 2 e 1,5, respectivamente.

Na verificação da normalidade na distribuição dos dados, constatou-se que tanto o grupo de controle (GC) quanto o grupo experimental (GE2) possuem distribuição não gaussiana, uma vez que $p - value < 0,05$ em ambos os grupos, conforme representado na Figura 5.15. Dessa forma, embora o cálculo da mediana reforce o resultado de que o grupo experimental tenha obtido pontuações superiores, o teste de Mann-Whitney foi aplicado resultando em $p = 0,4180$, conforme representado na Figura 5.16, o que indica que não há diferença com significância estatística entre os grupos GC e GE2 em relação ao nível de aprendizagem quando a abordagem adotada é a convencional ou a gamificada.

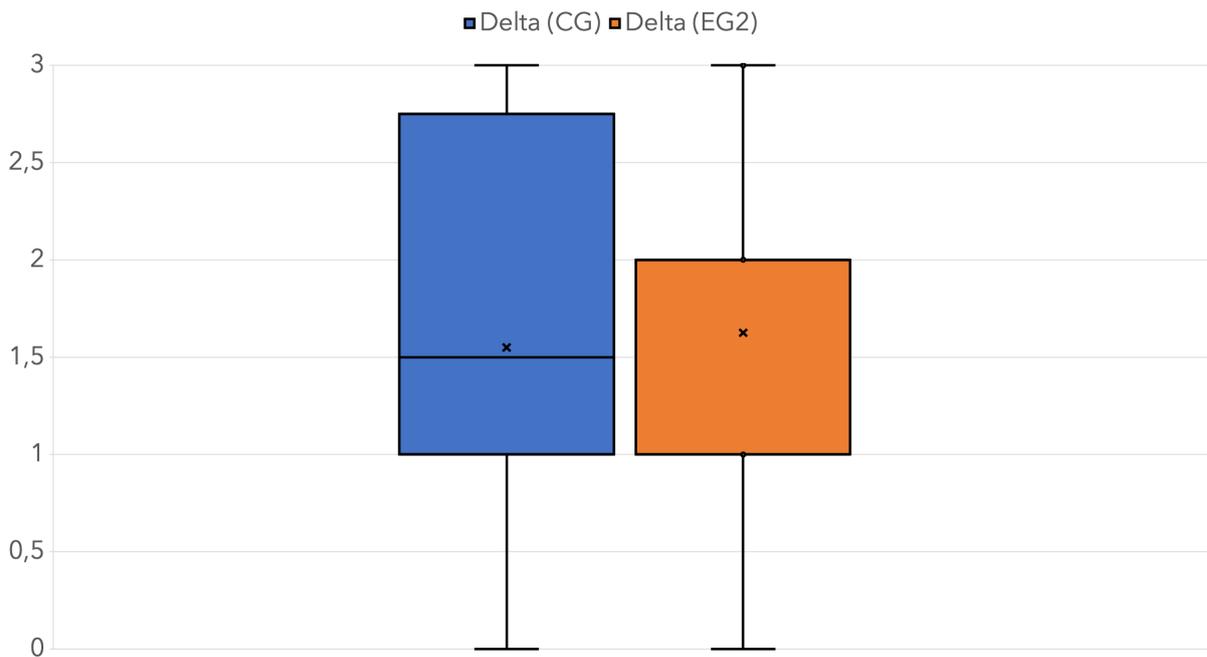


Figura 5.14: Boxplot com os valores Δ s dos testes de conhecimento - GC vs GE2

Fonte: Dados da pesquisa

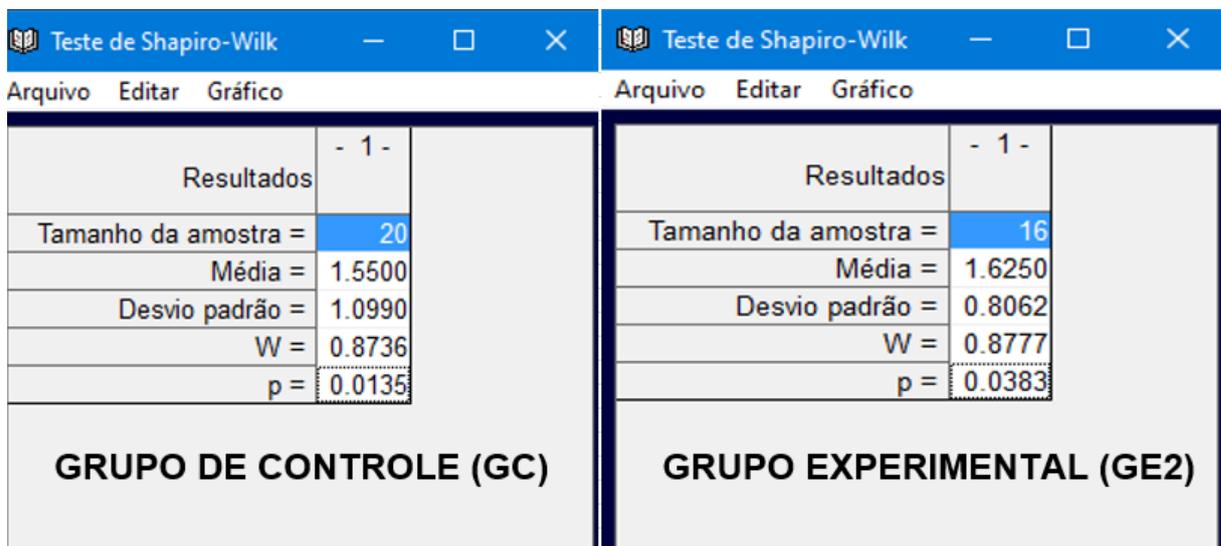
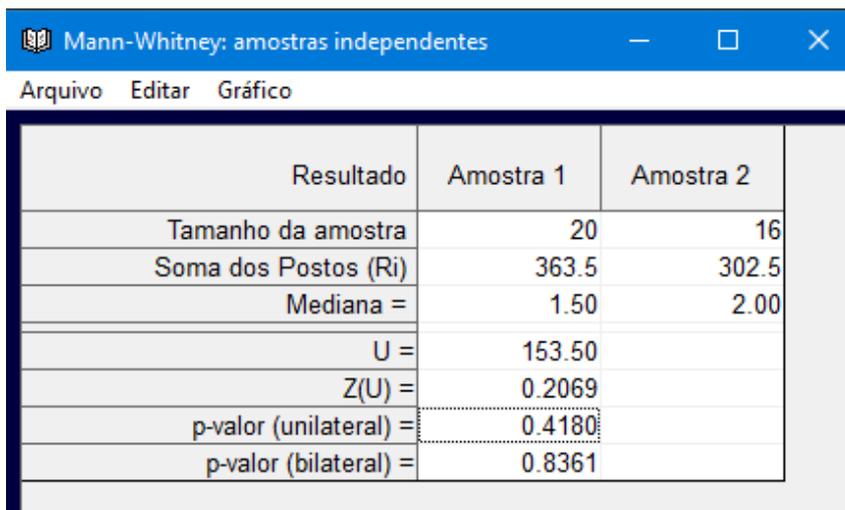


Figura 5.15: Resultado do teste de normalidade da distribuição dos dados no pré- e pós-teste - GC e GE2

Fonte: Dados da pesquisa

Em relação à eficácia dos alunos em responder corretamente as questões dos conceitos básicos de teste de software no *quiz* 1, nota-se que o grupo experimental (GE2) obteve uma mediana de 80% de eficácia, enquanto a mediana do grupo de controle (GC) foi de 90%, conforme representado na Figura 5.6. Em relação aos valores mínimos, o grupo de controle obteve 50% de eficácia, contra 40% do grupo



Resultado	Amostra 1	Amostra 2
Tamanho da amostra	20	16
Soma dos Postos (Ri)	363.5	302.5
Mediana =	1.50	2.00
U =	153.50	
Z(U) =	0.2069	
p-valor (unilateral) =	0.4180	
p-valor (bilateral) =	0.8361	

Figura 5.16: Teste de Mann-Whitney para o pré- e pós-teste - GC vs. GE2

Fonte: Dados da pesquisa

experimental (GE2). Por outro lado, o valor máximo obtido em ambos os grupos foi de 100% de eficácia.

Na verificação da normalidade na distribuição dos dados obtidos no *quiz 1*, constatou-se que ambos os grupos (GC e GE2) possuem distribuição não gaussiana, com $p - value = 0,0124$ e $p - value = 0,0476$ respectivamente, conforme representado na Figura 5.17. Como os grupos apresentaram distribuição não normal, aplicou-se o teste não paramétrico de Mann-Whitney resultando em $p - value = 0,0536$, conforme representado na Figura 5.18. Este resultado indica que não há diferença com significância estatística entre a eficácia dos grupos GC e GE2. Dessa forma, conclui-se que não é possível inferir que os alunos que aprenderam os conceitos básicos de teste de software por meio da abordagem gamificada foram mais eficazes em responder corretamente as questões do *quiz 1* do que os alunos que aprenderam por meio da abordagem convencional.

Em relação ao *quiz 2*, a mediana para o grupo de controle (GC) foi 80%, contra 85% no grupo experimental (GE2). Em relação aos valores mínimos e máximos, ambos os grupos (GC e GE2) obtiveram 60% de eficácia mínima e 100% de eficácia máxima, conforme representado na Figura 5.9. Para ambos os grupos, observou-se distribuição não gaussiana (Figura 5.19), e o teste de Mann-Whitney resultou em $p - value = 0,3614$ (Figura 5.20). Dessa forma, conclui-se que não há diferença com significância estatística entre a eficácia dos grupos GC e GE2.

- 1 -		- 3 -	
Resultados		Resultados	
Tamanho da amostra =	22	Tamanho da amostra =	16
Média =	81.3636	Média =	72.5000
Desvio padrão =	16.1232	Desvio padrão =	16.1245
W =	0.8814	W =	0.8850
p =	0.0124	p =	0.0476
GRUPO DE CONTROLE (GC)		GRUPO EXPERIMENTAL (GE2)	

Figura 5.17: Resultado do teste de normalidade da distribuição dos dados no Quiz 1 - GC e GE2

Fonte: Dados da pesquisa

Resultado	Amostra 1	Amostra 2
Tamanho da amostra	22	16
Soma dos Postos (Ri)	483.5	257.5
Mediana =	90.00	80.00
U =	121.50	
Z(U) =	1.6113	
p-valor (unilateral) =	0.0536	
p-valor (bilateral) =	0.1071	

Figura 5.18: Teste de Mann-Whitney para o Quiz 1 - GC vs. GE2

Fonte: Dados da pesquisa

Teste de Shapiro-Wilk		Teste de Shapiro-Wilk	
- 1 -		- 3 -	
Resultados		Resultados	
Tamanho da amostra =	22	Tamanho da amostra =	16
Média =	79.0909	Média =	81.2500
Desvio padrão =	16.3034	Desvio padrão =	15.0000
W =	0.7488	W =	0.8631
p =	0.0085	p =	0.0202
GRUPO DE CONTROLE (GC)		GRUPO EXPERIMENTAL (GE2)	

Figura 5.19: Resultado do teste de normalidade da distribuição dos dados no Quiz 2 - GC e GE2

Fonte: Dados da pesquisa

Mann-Whitney: amostras independentes			
Arquivo Editar Gráfico			
Resultado	Amostra 1	Amostra 2	
Tamanho da amostra	22	16	
Soma dos Postos (Ri)	417.0	324.0	
Mediana =	80.00	85.00	
U =	164.00		
Z(U) =	0.3548		
p-valor (unilateral) =	0.3614		
p-valor (bilateral) =	0.7227		

Figura 5.20: Teste de Mann-Whitney para o Quiz 2 - GC vs. GE2

Fonte: Dados da pesquisa

5.3.5 Análise Quantitativa 3 - (GC vs (GE1 + GE2))

Nesta terceira (e última) análise foram comparados os resultados das execuções experimentais realizadas com o grupo de controle (GC) na IES C e o terceiro grupo experimental (GE1 + GE2), que é a junção dos grupos experimentais da IES B e D.

Na Figura 5.21 são representados os resultados do questionário *Intrinsic Motivation Inventory* (IMI) aplicado aos 22 alunos do grupo de controle (GC) e à junção dos grupos experimentais GE1 e GE2, resultando em 27 alunos considerados. Nota-se que o grupo de controle (GC) foi superior em todas as categorias. Nota-se também que embora o grupo experimental (GE1) na análise 1 tenha obtido melhores resultados em relação à motivação quando comparado com o grupo de controle, percebe-se que a junção do GE1 com o GE2 resultou em pontuações inferiores nos aspectos motivacionais positivos (*Interesse/aproveitamento, Escolha, Competência*), e superior no aspecto motivacional negativo (*pressão/tensão*).

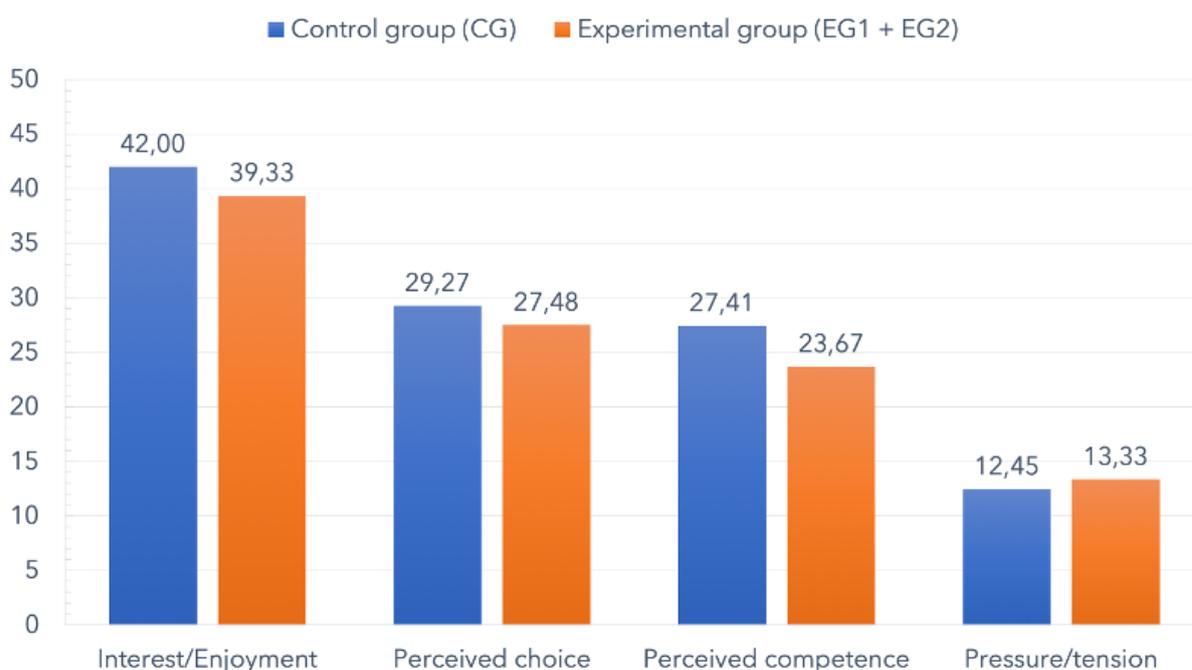


Figura 5.21: Resultados da Motivação - GC vs (GE1 + GE2)

Fonte: Dados da pesquisa

Em relação ao desempenho dos alunos, enquanto a pontuação máxima do grupo de controle (GC) foi de 2 pontos, no grupo experimental (GE1 + GE2) alguns alunos atingiram 3 pontos no pré-teste. Já as pontuações mínimas e máximas dos dois grupos foram similares no pós-teste, conforme representado na Figura 5.22. Tanto no pré-teste quanto no pós-teste, a mediana do grupo experimental (GE1 + GE2) foi superior ao grupo de controle (GC), sendo 1 e 0,5, e 3 e 2, respectivamente.

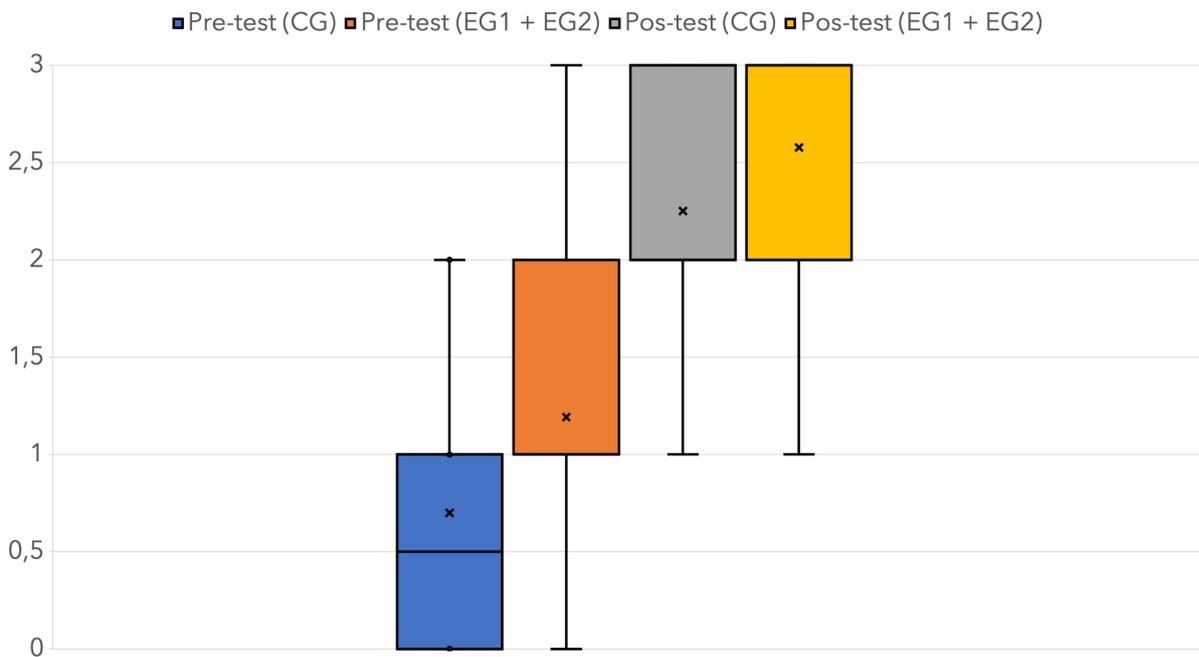


Figura 5.22: Resultado dos testes de conhecimento - GC vs (GE1 + GE2)

Fonte: Dados da pesquisa

Conforme observado no cálculo do Δ (referente à eficácia na aprendizagem, resumizada na Figura 5.23), nota-se uma variação similar em ambos os grupos e, assim, nenhuma tendência para um valor particular. A mediana demonstra que o grupo experimental (GE1 + GE2) obteve valores Δ inferiores ao grupo de controle (GC), sendo 1 e 1,5, respectivamente.

Distribuição não gaussiana foi novamente observada para ambos os grupos (Figura 5.24), e o teste de Mann-Whitney (Figura 5.25) indica que não há diferença com significância estatística entre os grupos GC e (GE1 + GE2) com relação ao nível de aprendizagem quando ambas as abordagens de ensino são consideradas.

A análise de eficácia com base no *quiz 1* (Figura 5.26) indica que o grupo experimental (GE1 + GE2) obteve uma mediana de 70% (mínimo de 40%, máximo de 100%), enquanto a mediana do grupo de controle (GC) foi de 90% (mínimo de 50%, máximo de 100%).

Os valores para ambos os grupos possuem distribuição não gaussiana (Figura 5.27), e o teste de Mann-Whitney (Figura 5.28) indicou que há uma diferença com significância estatística ($p - value = 0,0128$) entre a eficácia dos grupos GC e (GE1 + GE2). Dessa forma, conclui-se que é possível inferir que os alunos que aprenderam os conceitos básicos de teste de software por meio da abordagem convencional foram mais eficazes em responder corretamente as questões do *quiz 1* do que os alunos que aprenderam por meio da abordagem gamificada.

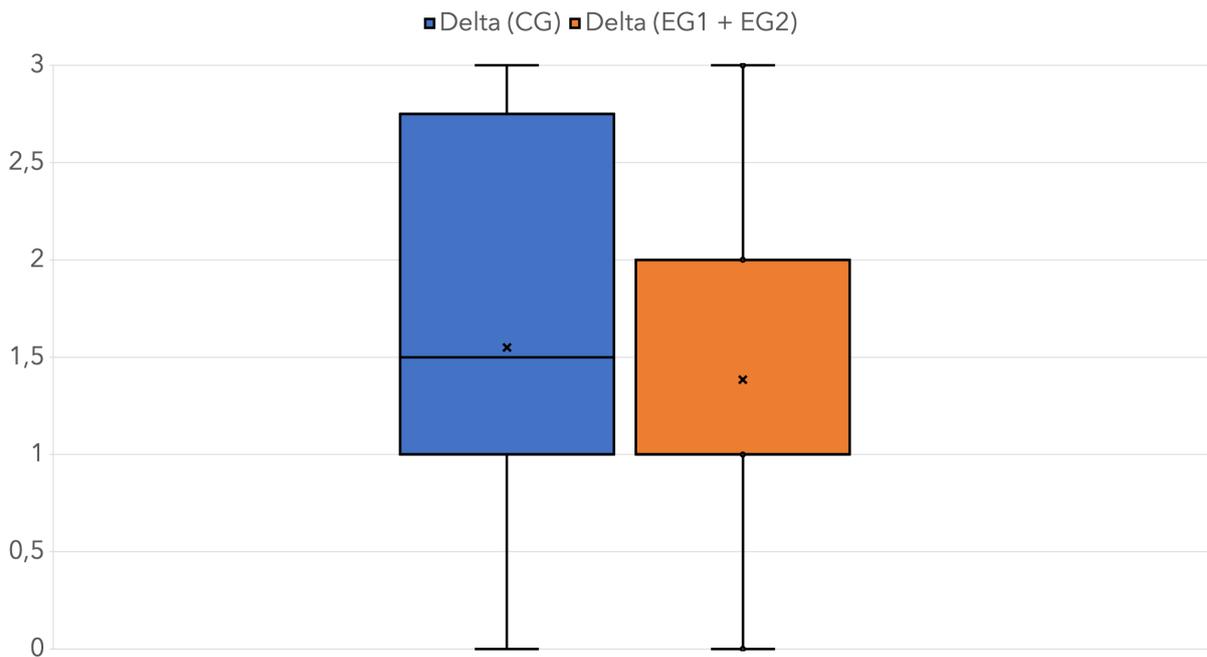


Figura 5.23: Boxplot com os valores Δ s dos testes de conhecimento - GC vs (GE1 + GE2)

Fonte: Dados da pesquisa

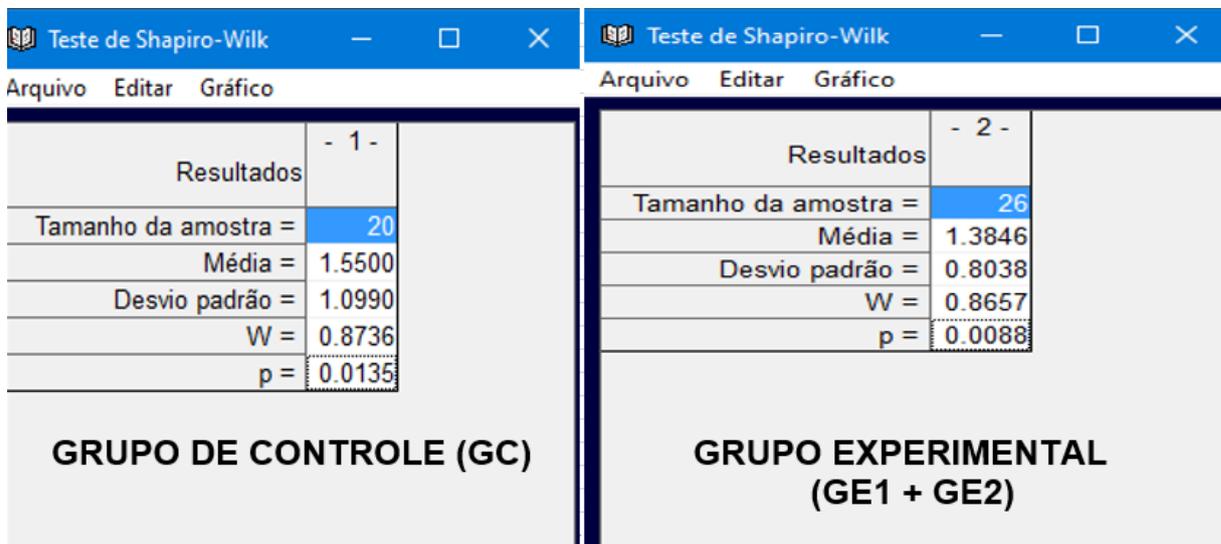


Figura 5.24: Resultado do teste de normalidade da distribuição dos dados no pré- e pós-teste - GC e (GE1 + GE2)

Fonte: Dados da pesquisa

O segundo *quiz* foi aplicado após o ensino dos critérios do teste funcional. A mediana de ambos os grupos (GC, e GE1 + GE2) foi de 80%. Em relação aos valores mínimos e máximos, ambos os grupos também obtiveram 60% de eficácia mínima e 100% de eficácia máxima, conforme representado na Figura 5.29.

Mann-Whitney: amostras independentes			
Arquivo Editar Gráfico			
Resultado	Amostra 1	Amostra 2	
Tamanho da amostra	20	26	
Soma dos Postos (Ri)	492.5	588.5	
Mediana =	1.50	1.00	
U =	237.50		
Z(U) =	0.4986		
p-valor (unilateral) =	0.3090		
p-valor (bilateral) =	0.6181		

Figura 5.25: Teste de Mann-Whitney para o pré- e pós-teste - GC vs (GE1 + GE2)

Fonte: Dados da pesquisa

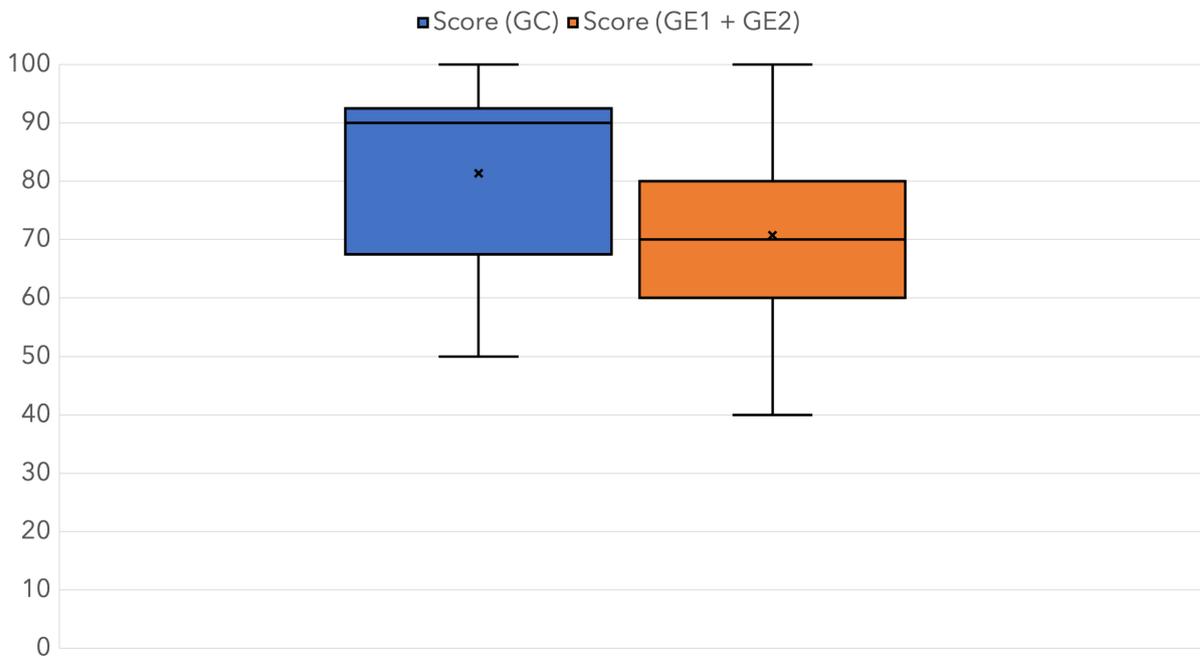


Figura 5.26: Eficácia dos alunos no *quiz* 1 - GC vs (GE1 + GE2)

Fonte: Dados da pesquisa

Na verificação da normalidade na distribuição dos dados, constatou-se que ambos os grupos (GC, e GE1 + GE2) possuem distribuição não gaussiana, com $p - value = 0,0085$ e $p - value = 0,0098$ respectivamente, conforme representado na Figura 5.30. O teste de Mann-Whitney (Figura 5.31, $p - value = 0,1802$) indica que não há diferença com significância estatística entre a eficácia dos grupos GC e (GE1 + GE2). Considerando esta perspectiva, conclui-se que não é possível inferir que os alunos que aprenderam teste funcional por meio da abordagem gamificada foram mais eficazes

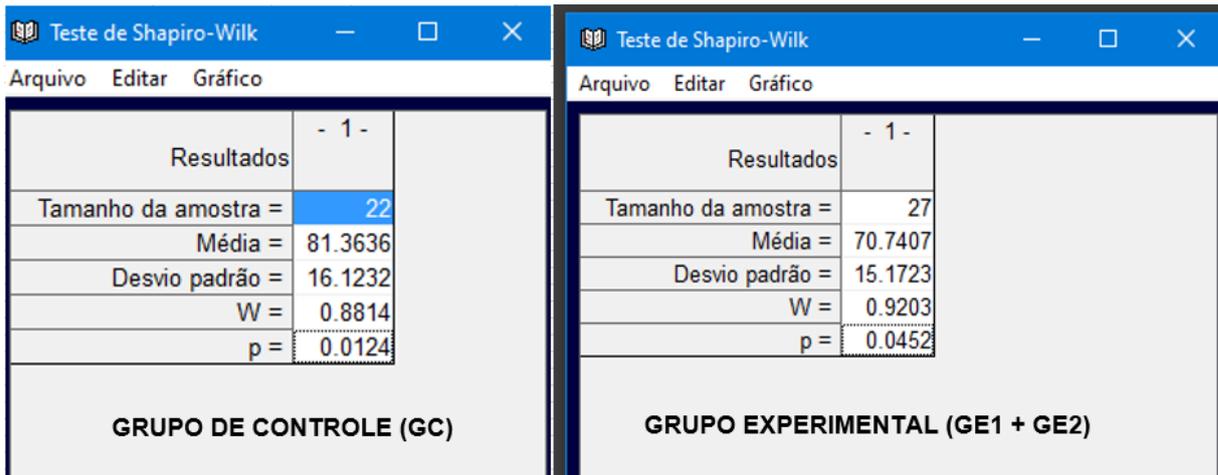


Figura 5.27: Resultado do teste de normalidade da distribuição dos dados no Quiz 1 - GC e (GE1 + GE2)

Fonte: Dados da pesquisa

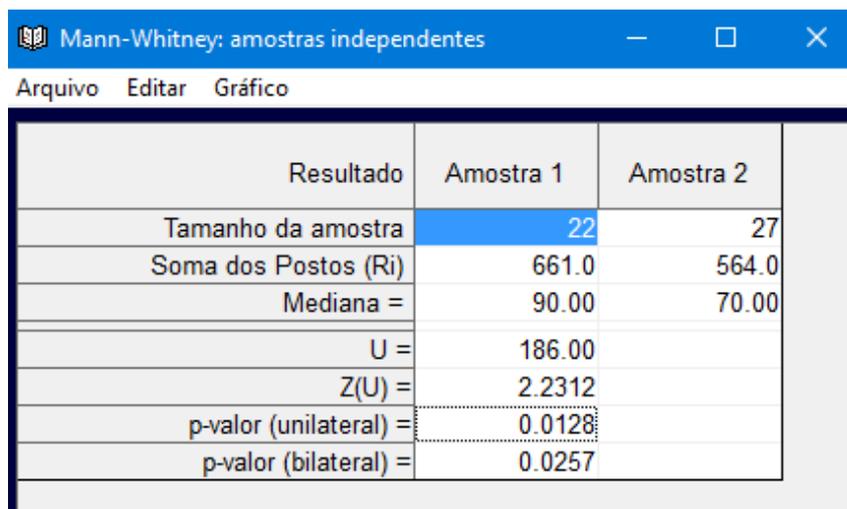


Figura 5.28: Teste de Mann-Whitney para o Quiz 1 - GC vs (GE1 + GE2)

Fonte: Dados da pesquisa

em responder corretamente as questões do *quiz 2* do que os alunos que aprenderam por meio da abordagem convencional.

5.3.6 Percepção dos Alunos

Para esta seção, apresentam-se na Tabela 5.2 as questões (e notas) do questionário *Avaliando o Minicurso*, cuja intenção foi obter o feedback dos alunos sobre os problemas enfrentados pelo ensino de teste de software que se buscou minimizar com o uso da gamificação.

Revisitando os problemas apresentados na Tabela 4.1, fundamentou-se nas consequências que cada questão poderia causar. Por exemplo, se a abordagem conven-

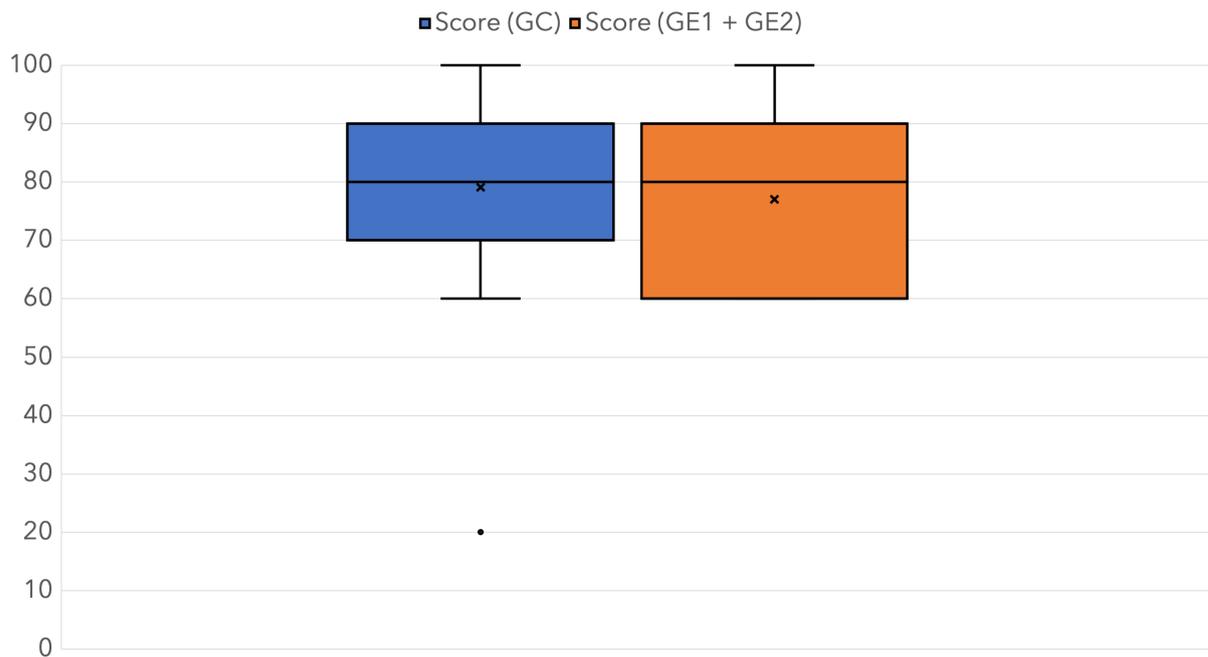


Figura 5.29: Eficácia dos alunos no *quiz 2* - GC vs (GE1 + GE2)

Fonte: Dados da pesquisa

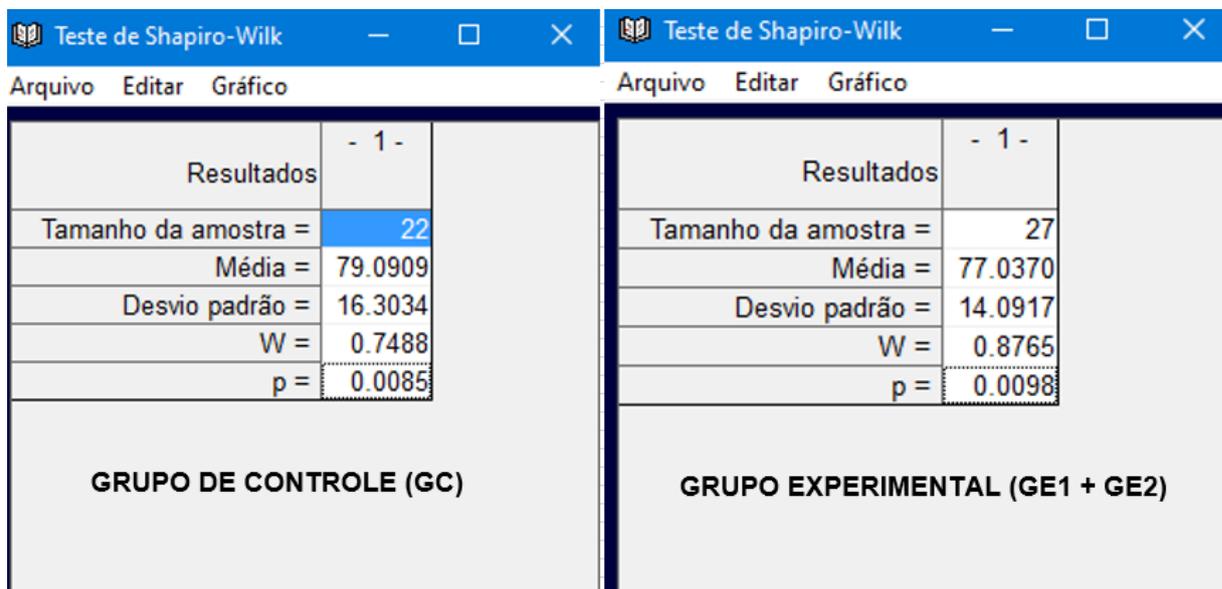


Figura 5.30: Resultado do teste de normalidade da distribuição dos dados no Quiz 2 - GC e (GE1 + GE2)

Fonte: Dados da pesquisa

cional tem sido ineficiente, uma possível consequência seria a distração dos alunos. Dessa forma, propôs-se o uso da gamificação visando tornar as aulas mais agradáveis, atraindo assim a atenção dos alunos para os conceitos aos quais estavam sendo expostos. Nesse sentido, a questão #1 do questionário visou verificar se este objetivo foi atingido, e o resultado foi que a abordagem gamificada aplicada nos grupos experi-

Resultado	Amostra 1	Amostra 2
Tamanho da amostra	22	27
Soma dos Postos (Ri)	595.5	629.5
Mediana =	80.00	80.00
U =	251.50	
Z(U) =	0.9146	
p-valor (unilateral) =	0.1802	
p-valor (bilateral) =	0.3604	

Figura 5.31: Teste de Mann-Whitney para o Quiz 2 - GC vs. (GE1 + GE2)

Fonte: Dados da pesquisa

Tabela 5.2: Resultados do feedback dos alunos

#	Questões	GC	GE1	GE2	GE1 + GE2
1.	A maneira como teste de software foi ensinado atraiu a minha atenção	5,30	6,55	5,50	5,93
2.	A maneira como teste de software foi ensinado tornou o aprendizado mais divertido.	4,65	6,64	5,38	5,89
3.	Os <i>quizzes</i> reforçaram o aprendizado do conteúdo abordado	6,35	6,73	6,00	6,30
4.	O exercício reforçou o aprendizado do conteúdo abordado.	6,45	6,36	5,63	5,93
5.	O exercício em grupo me motivou a trabalhar em equipe colaborando com os meus colegas.	6,50	5,55	5,50	5,52
6.	O nível de dificuldade nos <i>quizzes</i> e exercício foi coerente ao conteúdo ensinado.	6,05	6,82	6,63	6,70
7.	Ainda consigo lembrar do conteúdo ensinado durante o minicurso	5,85	5,55	5,81	5,70
8.	Depois deste minicurso eu tenho confiança de que consigo realizar as atividades de teste de software utilizando os critérios do Teste Funcional ensinados	5,25	5,36	4,56	4,89
9.	Pretendo realizar as atividades de teste de software em meus projetos futuros (mesmo se não forem requeridos).	6,55	5,91	5,06	5,41
10.	Após o minicurso eu compreendi a importância da atividade de teste.	7,00	6,82	6,25	6,48

Fonte: Elaborado pela autora

mentais (GE1 da IES B e GE2 da IES D) ajudou a atrair mais a atenção dos alunos do que a abordagem convencional. A comparação combinada dos grupos GE1 + GE2 também demonstra uma avaliação melhor neste quesito.

Outro resultado positivo obtido foi que o uso da gamificação tornou a aula mais divertida, conforme visto na questão #2 da Tabela 5.2. Mesmo que o desempenho dos alunos dos grupos experimentais (GE1, GE2, e GE1 + GE2) tenha sido inferior ao

grupo de controle (GC), destaca-se que ainda assim é considerado um desempenho satisfatório, tendo em vista que as notas obtidas são suficientes para aprovação dos alunos, considerando-se uma média igual a 7 pontos. Dessa forma, acredita-se que este trabalho é um primeiro passo em direção a uma abordagem mais eficiente de modo que o fator *diversão* seja mantido, e o desempenho dos alunos aumente.

Já a questão #5 refere-se ao comportamento esperado de *colaboração*. Embora esperava-se que a gamificação pudesse motivar a colaboração entre os alunos, especialmente quando trabalhando em equipes durante o exercício final, o resultado analisado sugere o oposto: o grupo de controle (GC) se sentiu mais motivado a trabalhar em equipe do que os grupos experimentais (GE1, GE1, e GE1 + GE2). Além disso, o grupo de controle (GC) também se sentiu mais motivado a utilizar os critérios funcionais em projetos futuros (mesmo se não forem requeridos) do que os grupos experimentais, conforme visto na questão #9.

Estes resultados levantam duas possibilidades: (i) ou a abordagem gamificada criada não foi suficiente para reduzir os desafios enfrentados pelo ensino de teste de software, ou (ii) o simples fato de se aplicar *quizzes* e um exercício prático (ambos sem elementos de jogos) depois de cada conceito ensinado foi tão bom quanto o uso de uma abordagem de ensino alternativa à convencional (neste caso, a abordagem gamificada). Dessa forma, sugere-se que mais investigação seja realizada para se obter respostas a estas questões.

Outro questionário (*Avaliando a Plataforma*) também foi aplicado aos dois grupos experimentais (GE1 e GE2) visando obter a opinião dos alunos em relação à plataforma *Bug Hunter*. Em uma das questões, perguntou-se “*O que mais te agradou na plataforma Bug Hunter*”. Dentre as respostas, foram mencionados o aspecto competitivo gerando um estímulo no aprendizado, feedback imediato, ambiente simples de entender, uso em tempo real (durante a aula), a evolução do avatar, e possibilidade de acompanhar o desenvolvimento de cada colega. Quando perguntado o que menos agradou, obtiveram-se como respostas a dificuldade em comprar itens para o avatar, poucas opções de recompensas para comprar com as moedas virtuais, e muita informação duplicada (por exemplo, abas redundantes). Em outra questão perguntou-se quais elementos de jogos foram mais motivadores na plataforma, e os três mais votados foram *competição*, *quadro de classificação*, e empatados *conquista* e *desafio*.

5.3.7 Análise Qualitativa

Durante as execuções experimentais, a pesquisadora observou os comportamentos, comentários, e semblantes dos alunos que participaram do experimento.

Atenção dos alunos: Embora os alunos dos dois grupos experimentais (GE1 e GE2) não tenham tido conversas paralelas durante a aula, notou-se que constantemente

se encontravam distraídos navegando pela plataforma *Bug Hunter*. Acredita-se que, possivelmente, o ambiente gamificado tenha causado um desvio de atenção, resultando assim em um desempenho menor quando comparado ao desempenho do grupo de controle. Além disso, percebeu-se certa tensão e distração no grupo experimental (GE1) da IES B, pois constantemente os alunos comentavam sobre a monografia de trabalho de conclusão de curso que deveria ser entregue na semana corrente, motivo pelo qual alguns alunos não participaram do experimento.

Participação dos alunos: Apenas 4 dos alunos do grupo experimental (GE1) participaram ativamente durante o ensino dos conceitos básicos de teste de software. No entanto, quando deu-se início ao primeiro *quiz* competitivo, toda a turma demonstrou euforia e diversão. Depois disso, quase todos os alunos participaram fazendo e/ou respondendo as perguntas, o que impactou no tempo disponível para a execução do restante do conteúdo. No segundo grupo experimental (GE2), os alunos participaram menos que o esperado, tanto antes quanto depois da aplicação dos *quizzes*. Por outro lado, os alunos do grupo de controle (GC) foram mais curiosos e participativos ao longo de toda sessão. Como destacado pelo coordenador do curso, a IES C fica localizada em um distrito muito pequeno do estado do Espírito Santo, e esta foi a primeira oportunidade que os alunos tiveram de participar de algo diferente, colaborando assim com a sua formação. O oposto é observado no grupo experimental (GE2) da IES D, uma vez que está localizada em São Carlos, uma cidade no interior de São Paulo conhecida como a "capital da tecnologia", sendo assim um local onde acontecem frequentes eventos e conta com inúmeras oportunidades na área da computação. Dessa forma, acredita-se que fatores regionais podem ter afetado os resultados de um experimento dessa natureza.

Colaboração entre os alunos: A plataforma *Bug Hunter* contém uma aba chamada *Fórum*, na qual os alunos podem postar suas dúvidas ou responder as perguntas dos colegas. A intenção desta aba é oferecer um espaço onde os alunos com perfil *socializador* possam interagir com os demais. No entanto, nenhum aluno de ambos os grupos (controle ou experimentais) utilizou o Fórum. Outra forma de colaboração esperada foi durante o exercício final, onde os alunos foram divididos em grupos no qual deveriam colaborar entre si durante a criação e execução dos casos de teste, semelhantemente como ocorre em cenários reais da indústria. Apesar disso, notou-se que este exercício foi a atividade menos agradável para os grupo experimentais (GE1 e GE2), contrário ao grupo de controle (GC) que apontou como sendo a melhor atividade durante a sessão experimental.

Aplicação dos conceitos aprendidos: Era esperado que os alunos aplicassem os critérios Análise de Valor Limite e Particionamento de Equivalência para a criação dos casos de teste durante o exercício final. Nesta atividade, 6 das 7 equipes do grupo

de controle (GC), e 3 das 4 equipes do grupo experimental (GE2) utilizaram os dois critérios. No entanto, no grupo experimental (GE1) apenas 1 das 4 equipes utilizou os dois critérios, 1 equipe utilizou um dos critérios; 1 equipe não utilizou critério e 1 equipe não realizou a atividade (encontrou dificuldades em colocar em prática o que aprendeu durante a aula). Destaca-se que os alunos do grupo de controle (GC) constantemente solicitaram o auxílio da pesquisadora durante o exercício, enquanto os alunos dos grupos experimentais (GE1 e GE2) buscaram resolver sozinhos por terem se sentido “desafiados”, conforme mencionaram.

Comentários dos alunos: Um aspecto negativo preocupou todos os grupos (controle e experimentais): o conteúdo ensinado foi muito extenso para ser aprendido em um curto período de tempo. Outro aspecto negativo observado no grupo de controle foi a conversa paralela entre alguns alunos durante toda a aula. Como pontos positivos, no grupo experimental (GE1), comentou-se abertamente que *“todas as aulas deveriam ser tão dinâmicas quanto esta”*. Os alunos do GE2 também afirmaram que os *quizzes* reforçaram o conhecimento adquirido, e que foi positivo a possibilidade de colocar em prática, através do desafio final, o que aprenderam sobre teste funcional em um *website* já conhecido e utilizado por eles.

Expressões faciais dos alunos: Em nenhum momento foi percebido nos alunos dos grupos experimentais (GE1 e GE2) expressões de tédio ou fadiga. Por outro lado, no grupo de controle (GC) percebeu-se tais expressões e sonolência, fatos confirmados no questionário de avaliação do minicurso no qual 6 alunos do GC afirmaram que a mesma forma de aula teórica (e massante) aplicada nos cursos foi o que menos agradou no minicurso, e que deveriam ser acrescentadas mais atividades práticas.

5.4 Revisitando as Questões de Pesquisa

Visando verificar o impacto do uso da gamificação na motivação e desempenho dos estudantes no contexto de ensino de teste de software, foram executadas duas sessões experimentais e uma replicação. Com base nos resultados obtidos após a análise dos dados, pode-se concluir:

- **QP1.** *A motivação dos alunos que aprendem teste de software por meio de uma abordagem gamificada é maior do que a motivação dos alunos que aprendem por meio da abordagem convencional?*

Resposta para Análise 1: Como 2 dos 3 aspectos motivacionais positivos (sendo eles *Interesse/diversão, escolha percebida, competência percebida*) no grupo experimental (GE1) obtiveram maior pontuação, e o aspecto negativo (isto é,

Pressão/Tensão) obteve pontuação menor do que no grupo de controle (GC), conclui-se que a motivação dos estudantes que aprenderam teste de software por meio da abordagem gamificada foi **maior** do que a motivação dos alunos que aprenderam por meio da abordagem convencional. Portanto, rejeita-se a hipótese nula (**H1.0**).

Resposta para Análise 2: O grupo experimental (GE2) obteve pontuação menor do que o grupo de controle (GC) nos 3 aspectos motivacionais positivos (sendo eles *Interesse/diversão, escolha percebida, competência percebida*), e maior pontuação no aspecto negativo (isto é, *Pressão/Tensão*). Dessa forma, conclui-se que a motivação dos estudantes que aprenderam teste de software por meio da abordagem gamificada foi **menor** do que a motivação dos estudantes que aprenderam por meio da abordagem convencional. Portanto, rejeita-se a hipótese nula (**H1.0**).

Resposta para Análise 3: Nota-se que embora o grupo experimental (GE1) da análise 1 tenha obtido melhores resultados em relação à motivação quando comparado ao grupo de controle, percebe-se que a junção do GE1 com o GE2 resultou em pontuações inferiores nos aspectos motivacionais positivos (*Interesse/aproveitamento, Escolha, Competência*), e superior no aspecto motivacional negativo (*Pressão/tensão*). Dessa forma, combinando GE1 + GE2, conclui-se que a motivação dos estudantes que aprenderam teste de software por meio da abordagem gamificada foi **menor** do que a motivação dos estudantes que aprenderam por meio da abordagem convencional. Portanto, rejeita-se a hipótese nula (**H1.0**).

Destaca-se que a análise da hipótese foi baseada no valor da média obtida em cada aspecto motivacional, ou seja, não foi realizado teste estatístico para o teste de hipótese da QP1. Dessa forma, considerou-se que a motivação foi maior no grupo que apresentou média superior na maioria dos aspectos motivacionais positivos (sendo, *Interesse/diversão, escolha percebida, competência percebida*), e inferior no aspecto motivacional negativo (isto é, *Pressão/Tensão*). Por exemplo, como o GE1 obteve médias superiores na maioria dos aspectos positivos e inferior no aspecto negativo (considerando a análise 1 da QP1.), considera-se como sendo o grupo que apresentou maior motivação.

- **QP2.** *O desempenho dos alunos que aprendem teste de software por meio de uma abordagem gamificada é maior do que o desempenho dos alunos que aprendem por meio da abordagem convencional?*

Resposta para Análise 1: Não há uma diferença com significância estatística entre o grupo de controle (GC) e o grupo experimental (GE1) em relação ao nível de aprendizagem verificado através dos pré- e pós- testes entre as abordagens

adotadas, seja a gamificada ou a convencional. No entanto, observou-se em ambos os *quizzes* uma diferença com significância estatística na eficácia de cada grupo ($p - value = 0,0155$ no primeiro *quiz*, e $p - value = 0,0140$ no segundo), revelando que o desempenho dos alunos que aprenderam teste de software por meio da abordagem gamificada foi **menor** do que o desempenho dos alunos que aprenderam por meio da abordagem convencional. Portanto, rejeita-se a hipótese nula (**H2.0**).

Resposta para Análise 2: Não houve diferença com significância estatística entre o grupo de controle (GC) e o grupo experimental (GE2) em relação ao nível de aprendizagem. Além disso, observou-se que em nenhum dos dois *quizzes* houve uma diferença com significância estatística na eficácia de cada grupo, o que indica que o desempenho dos alunos que aprenderam teste de software por meio da abordagem gamificada foi **similar** ao que o desempenho dos alunos que aprenderam por meio da abordagem convencional. Portanto, não é possível rejeitar a hipótese nula (**H2.0**).

Resposta para Análise 3: Não houve diferença com significância estatística entre o grupo de controle (GC) e a junção dos grupos experimentais (GE1 + GE2) em relação ao nível de aprendizagem. Além disso, observou-se que no primeiro *quiz* houve uma diferença com significância estatística na eficácia entre GC e GE1 + GE2. No entanto, a junção dos dois grupos experimentais resultou em uma diferença não significativa no segundo *quiz*. Dessa forma, pode-se considerar que, em geral, o desempenho dos alunos que aprenderam por meio da abordagem gamificada foi **similar** ao desempenho dos alunos que aprenderam por meio da abordagem convencional. Portanto, não é possível rejeitar a hipótese nula (**H1.0**).

5.5 Ameaças à Validade

As ameaças à validade identificadas são classificadas em quatro categorias, sendo elas: interna, externa, constructo, e conclusão (Wohlin et al., 2012). Além disso, as ações tomadas a fim de mitigar as ameaças também são apresentadas.

Validade interna: As ameaças identificadas são: (1) influência da pesquisadora; (2) influência da regionalidade; (3) conteúdo ensinado, e questões e requisitos utilizados nos *quizzes* e exercício final, respectivamente; e (4) fadiga. Em relação à (1), a mesma pesquisadora executou todas as sessões experimentais e as análises foram conduzidas e discutidas com outros pesquisadores. Em relação à (2), as Instituições de Ensino Superior (IES) participantes do experimento estão localizadas em regiões distintas

do país, embora todas na região Sudeste. A principal ameaça identificada se refere às oportunidades que os alunos destas regiões têm em participarem de eventos e pesquisas relacionados à computação. Dessa forma, após análise dos resultados, observou-se que o contexto da regionalidade pode ter influenciado na motivação e desempenho dos participantes deste estudo de mestrado. Sobre a ameaça (3), o conteúdo de teste de software preparado para ser apresentado aos alunos foi revisado por um professor PhD que tem ensinado a disciplina de engenharia de software há mais de 8 anos em uma universidade federal brasileira. Além disso, as questões adicionadas nos *quizzes* foram selecionadas dentro dos conceitos que foram ensinados, e os requisitos do *website* apresentados para a criação dos casos de teste foram definidos considerando os que frequentemente são utilizados em exercícios das disciplinas de engenharia de software. Por fim, em relação à ameaça (4), embora houvesse ciência de que sessões experimentais com quatro horas de duração pudessem causar fadiga, houveram limitações (discutidas na Seção 6.2) que impediram a divisão das sessões nas IES C e D em dois dias distintos. Ainda assim, entende-se que os alunos destas duas IES estão acostumados à realidade de aulas com quatro horas seguidas de duração.

Validade externa: As ameaças externas identificadas são: (1) conhecimento prévio; (2) formação das equipes; (3) questões inseridas nos *quizzes* e requisitos no exercício final. Em relação à (1), os participantes já haviam tido disciplinas que abordaram tópicos de teste de software. Em relação à (2), foi permitido que os próprios alunos se dividissem em grupos (equipes) a fim de se obter um maior engajamento e colaboração. Por fim, em relação à (3), as questões e requisitos utilizados não representam todos os possíveis cenários.

Validade de constructo: Esta ameaça diz respeito às métricas utilizadas para analisar a motivação e o desempenho dos alunos. Em relação à motivação, utilizou-se o *Intrinsic Motivation Inventory* (IMI), proposto e aplicado por Deci e Ryan (2011) para medir a motivação intrínseca dos participantes de experimentos. Por outro lado, em relação ao desempenho, utilizou-se a estatística descritiva e, adicionalmente, aplicou-se o teste de Shapiro-Wilk e Mann-Whitney para comparar dois grupos independentes (controle e experimental).

Conclusão: A amostra da pesquisa é limitada (11 alunos no GE1, 22 no GC, e 16 no GE3). Como 7 dos 18 alunos do GE1 não participaram completamente das sessões, as amostras ficaram mais desbalanceadas. Destaca-se que, além do número de alunos matriculados em disciplinas avançadas ser baixo quando comparado às disciplinas iniciais dos cursos de computação, há também uma certa dificuldade em atraí-los a participarem deste tipo de experimento. Além disso, a principal dificuldade encontrada foi o lento processo de obtenção da aprovação dos comitês de ética (que pode levar

meses). Tendo estes pontos em consideração, acredita-se que a abordagem desenvolvida e os resultados obtidos podem servir de base para que este experimento seja replicado⁵ (eventualmente com refinamentos) por outros pesquisadores e com uma amostra mais robusta.

5.6 Considerações Finais

Sumarizando os resultados obtidos com as sessões experimentais conduzidas, não é possível afirmar, em geral, que a motivação e o desempenho dos alunos que aprendem por meio da abordagem gamificada é maior do que a motivação e o desempenho dos alunos que aprendem por meio da abordagem convencional, tendo em vista que para cada grupo experimental (e para a junção de ambos) o resultado foi diferente quando comparado com o mesmo grupo de controle (GC). No entanto, considerando-se a experiência vivida durante o experimento e as lições aprendidas, afirma-se que o uso de uma abordagem alternativa à convencional foi positiva e válida, especialmente por ter proporcionado um ambiente de aprendizagem mais agradável e divertido. Dessa forma, sugere-se que a abordagem desenvolvida neste trabalho seja refinada e que mais investigações sejam realizadas, idealmente através de experimentos mais próximos da realidade, como no decorrer de todo o semestre letivo.

No próximo capítulo são apresentadas as conclusões deste trabalho, assim como as principais contribuições, limitações e possibilidades de trabalhos futuro. Publicações obtidas e materiais em elaboração também são mencionados.

⁵Note que todo material experimental está disponível em <<https://bit.ly/2NqKdUr>> - Acessado em 12 de dezembro de 2019.

Capítulo 6

CONCLUSÃO

Desde que os computadores se tornaram populares, houve um tremendo crescimento na indústria de software. Sistemas de software inicialmente restritos aos domínios de computação científica e processamento de dados agora permeiam o cotidiano. Dada a onipresença dos sistemas de software, é discutível que a qualidade desses sistemas possa ter consequências de longo alcance, afetando diretamente as atividades diárias. Neste contexto, o teste de software desempenha um papel essencial na melhoria da qualidade dos sistemas, sendo realizado na expectativa de garantir que o software faça o que fora projetado para fazer, enquanto verifica também se não executa algo não intencional (Myers et al., 2011).

Apesar da importância da qualidade do software, a educação no contexto de teste sofre com alguns problemas, como o tempo insuficiente dedicado ao ensino dos conceitos, desalinhamento entre o que se ensina e o que se aplica na indústria, ineficiência das aulas teóricas, aulas pouco atrativas, e falta de atividades práticas. Tais problemas podem gerar como consequência o desinteresse, desatenção e desmotivação dos alunos. Visando mitigar tais problemas, abordagens de ensino alternativas à convencional vêm sendo investigadas, e uma delas é a gamificação que, conforme observado no mapeamento sistemático descrito no Capítulo 3 desta dissertação, tem sido aplicada tanto em contextos educacionais quanto industriais visando, dentre outros, aumentar a motivação e desempenho das pessoas. Os benefícios alcançados nos estudos analisados inspiraram este trabalho de mestrado a investigar se o uso da gamificação no ensino de teste de software aumenta a motivação e o desempenho dos alunos quando comparado com a abordagem convencional de ensino.

Dada a importância das atividades de teste de software e os desafios enfrentados em seu contexto educacional, neste trabalho seguiu-se uma série de passos para a criação de uma abordagem gamificada visando mitigar tais desafios. Para verificar o impacto do uso da gamificação na motivação e desempenho dos alunos em aprender teste de software, conduziram-se duas execuções e uma replicação experimental para abordar as questões de pesquisa deste trabalho, respondidas na Seção 5.4.

Embora a análise descritiva tenha revelado um desempenho dos grupos experimentais (GE1 e GE2) inferior ao grupo de controle, acredita-se que o uso da gamificação no ensino de teste de software é válido. Para embasar esse pensamento, destacam-se os diversos benefícios positivos alcançados com a abordagem desenvolvida, tais como a motivação e participação e atenção dos alunos, além de ter proporcionado um ambiente mais agradável e divertido, tanto para ensinar (do ponto de vista da pesquisadora) quanto para aprender (do ponto de vista dos alunos).

As principais descobertas obtidas com este trabalho são:

- Embora corrobore com resultados positivos em termos de motivação e criação de um ambiente mais divertido e agradável, o uso da gamificação nem sempre é a solução adequada para os problemas que se deseja resolver;
- Incluir *quizzes* e atividades práticas após o ensino de conceitos pode auxiliar a atrair a atenção dos alunos e a fixar o conteúdo aprendido;
- A construção de um ambiente gamificado é um processo incremental complexo e desafiador, especialmente na fase de definição de um sistema de recompensas e *range* de pontuações e níveis. Este processo necessita de estudos piloto com objetivo de obter feedback dos participantes para, em seguida, criar um ambiente e todo um material didático mais adequado.

Um resumo das contribuições mais gerais obtidas neste trabalho é apresentado na próxima seção.

6.1 Contribuições

As principais contribuições deste trabalho são:

- Caracterização do estado da arte sobre o uso da gamificação em teste de software;
- Identificação de alguns dos problemas presentes no contexto de ensino de teste de software;
- Criação de uma abordagem gamificada baseada em um conjunto de passos seguidos, além da possibilidade de adaptá-la a diversos contextos, seja no ensino de outras disciplinas ou treinamentos na indústria;
- Desenvolvimento da abordagem gamificada em uma plataforma que também pode ser utilizada (e adaptada) em outros contextos;

- Análise da viabilidade do uso da abordagem gamificada desenvolvida;

Além das contribuições principais, consideram-se as seguintes contribuições secundárias deste trabalho de mestrado:

- Disponibilização de todo material¹ utilizado nesta pesquisa permitindo a replicabilidade;
- Colaboração na formação acadêmica dos alunos que participaram da pesquisa, especialmente por se tratar de um assunto pouco abordado na academia, mas o mais utilizado na indústria.
- Levantamento de novos questionamentos, reforçando a necessidade de mais investigações e projetos de pesquisa sobre ensino de teste de software e abordagens pedagógicas;

6.2 Limitações e Trabalhos Futuros

Algumas limitações foram identificadas nesta pesquisa. Considera-se que o ideal seria a realização do experimento ao longo de todo o semestre acadêmico a fim de se aproximar do cenário real de ensino. No entanto, apenas 4 horas foram oferecidas em cada instituição nas quais as sessões experimentais foram executadas. Dessa forma, o experimento foi projetado em formato de minicurso a fim de ser possível apresentar os conceitos mais importantes de teste de software e focar no ensino mais aprofundado sobre teste funcional com exemplos práticos.

O grupo experimental (GE1) teve duas sessões de duas horas divididas em dois dias consecutivos (totalizando quatro horas), enquanto o grupo de controle (GC) e o grupo experimental (GE2) tiveram as quatro horas de sessão em um único dia, com um intervalo de 20 minutos. Embora haja uma recomendação de que sessões experimentais não ultrapassem duas horas a fim de evitar fadiga, (Siegmund e Schumann, 2015), o grupo de controle (GC) se encontrava na última semana de aula do semestre acadêmico, não tendo outro dia disponível para a execução do experimento. Além disso, um ponto interessante é que mesmo com a possibilidade de fadiga, o GC obteve um desempenho melhor quando comparado ao GE1. O lado positivo do grupo experimental (GE2) também ter ocorrido em quatro horas seguidas foi a possibilidade de comparar os grupos (GC e GE2) de forma mais homogênea.

Acredita-se que a falta de estudos que apoiem o processo de *design* de uma abordagem gamificada que se encaixasse nas necessidades deste trabalho também

¹Disponível em <<https://bit.ly/2N9KdUr>> - Acessado em 12 de dezembro de 2019.

pode ter impactado nos resultados, principalmente quando se leva em consideração que o desenvolvimento deste tipo de abordagem é um desafio. Dessa forma, para trabalhos futuros, sugerem-se refinamentos e melhorias na gamificação implementada neste trabalho, como a criação de um sistema de pontos cujo funcionamento seja mais claro para os participantes, criação de selos com *design* mais atrativo, e criação de novos *bens virtuais*. Além disso, sugere-se a remoção de informações redundantes na plataforma gamificada (por exemplo, as regras para as conquistas apresentadas nas abas *Selos* e *Regras*), a possibilidade de interação dos participantes com a ferramenta, e o desenvolvimento de uma aplicação com *design (aesthetic)* mais atrativo e adaptativo, baseando-se nos perfis dos usuários que a utilizarão. Por fim, novas execuções experimentais ao longo do semestre letivo e com um número maior de alunos podem (e devem) ser executadas a fim de se obter uma amostragem mais representativa.

6.3 Publicações

No decorrer do desenvolvimento deste trabalho de mestrado foram publicados dois artigos, sendo “Gamification in Software Testing: A Characterization Study” (Jesus et al., 2018), e “Is It Worth Using Gamification on Software Testing Education? An Experience Report” (Jesus et al., 2019).

O primeiro artigo foi publicado no III Simpósio Brasileiro de Teste de Software Sistemático e Automatizado (SAST’18)² e se trata de um mapeamento sistemático da literatura conduzido visando caracterizar o uso da gamificação como uma estratégia para apoiar (e reforçar a qualidade) das atividades de teste de software. O MS realizado proveu uma visão mais concreta do contexto pesquisado e serviu de subsídio para o desenvolvimento deste projeto de mestrado.

O segundo artigo, por sua vez, foi publicado no XVIII Simpósio Brasileiro de Qualidade de Software (SBQS’19)³, onde foram reportados os resultados parciais deste trabalho.

Além dos artigos já publicados, um outro trabalho em colaboração com colegas do Laboratório de Pesquisa em Engenharia de Software (LAPES) foi submetido para o periódico *Journal of Systems and Software*, o qual encontra-se em segunda rodada de avaliação pelo corpo editorial da revista. Por fim, um último artigo está sendo escrito em

²Disponível em <<http://cbsoft2018.icmc.usp.br/#/sast/>> - Acessado em 12 de dezembro de 2019.

³Disponível em <<http://sbqs.sbc.org.br/index.php/pt/anais>> - Acessado em 12 de dezembro de 2019.

paralelo a esta dissertação com submissão prevista para um workshop internacional de educação em teste de software.

REFERÊNCIAS

- ANDERSON, P. E.; NASH, T.; MCCAULEY, R. Facilitating Programming Success in Data Science Courses Through Gamified Scaffolding and Learn2Mine. In: *ITiCSE*, Vilnius, Lithuania: ACM, p. 99–104, 2015.
- ANDERSON, P. E.; TURNER, C.; DIERKSHEIDE, J.; MCCAULEY, R. An Extensible Online Environment for Teaching Data Science Concepts Through Gamification. In: *IEEE FIE Conference*, Madrid, Spain: IEEE, p. 1–8, 2014.
- BARBOSA, E. F.; MALDONADO, J. C. Collaborative development of educational modules: a need for lifelong learning. *E-Infrastructures and technologies for lifelong learning: next generation environments*, 2011.
- BARTLE, R. Hearts, Clubs, Diamonds, Spades: Players Who Suit MUDs. *Journal of MUD Research*, 1996.
- BELL, J.; SHETH, S.; KAISER, G. Secret Ninja Testing with HALO Software Engineering. In: *SSE Workshop*, Szeged, Hungary: ACM, p. 43–47, 2011.
- BENITTI, F. B. V. As a Teacher, I Want to Know What to Teach in Requirements Engineering So That Professionals Can Be Better Prepared. In: *SBES'17*, New York, NY, USA: ACM, p. 318–327, 2017.
- BUDD, T. A. *Mutation Analysis of Program Test Data*. PhD Thesis, New Haven, CT, USA, aI8025191, 1980.
- CAPGEMINI GROUP; SOGETI; HP *World quality report 2015-16*. Tech. Report RT-ES 679/05, Cap Gemini S.A., Sogeti and Hewlett-Packard, disponível em <https://www.capgemini.com/thought-leadership/world-quality-report-2015-16> - Acessado em 28 de novembro de 2019., 2016.
- CARRINGTON, D. Teaching Software Testing. In: *Proceedings of the 2nd Australasian Conference on Computer Science Education (ACSE)*, Melbourne, Australia: ACM, p. 59–64, 1996.
- CHEIRAN, J. F. P.; RODRIGUES, E. M.; CARVALHO, E. L. S.; SILVA, J. P. S. Problem-Based Learning to Align Theory and Practice in Software Testing Teaching. In: *Proceedings of the 31st Brazilian Symposium on Software Engineering*, New York, NY, USA: ACM, p. 328–337, 2017.

- CLARKE, P. J.; DAVIS, D.; M., T.; KING; PAVA, J.; JONES, E. L. Integrating Testing into Software Engineering Courses Supported by a Collaborative Learning Environment. *Trans. on Comp. Education*, v. 14, n. 3, p. 18:1–18:33, 2014.
- CLEGG, B. S.; ROJAS, J. M.; FRASER, G. Teaching Software Testing Concepts Using a Mutation Testing Game. In: *ICSE SEER Track*, Buenos Aires, Argentina: IEEE, p. 33–36, 2017.
- CSIKSZENTMIHALYI, M. *Flow: The Psychology of Optimal Experience*. 1st ed. Harper Perennial Modern Classics, 1990.
- DAL SASSO, T.; MOCCI, A.; LANZA, M.; MASTRODICASA, E. How to Gamify Software Engineering. In: *SANER Conf.*, Klagenfurt, Austria: IEEE, p. 261–271, 2017.
- DECI, E. L.; RYAN, R. M. Levels of Analysis, Regnant Causes of Behavior and Well-being: The Role of Psychological Needs. *Psychological Inquiry*, v. 22, n. 1, p. 17–22, 2011.
- DELAMARO, M.; JINO, M.; MALDONADO, J. *Introdução ao Teste de Software*. 2nd. ed. Elsevier Editora Ltda., 2017.
- DEMILLO, R. A.; LIPTON, R. J.; SAYWARD, F. G. Hints on test data selection: Help for the practicing programmer. *Computer*, v. 11, n. 4, p. 34–41, 1978.
- DETERDING, S.; DIXON, D.; KHALED, R.; NACKE, L. From Game Design Elements to Gamefulness: Defining “Gamification”. In: *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, Tampere, Finland: ACM, p. 9–15, 2011.
- DICHEVA, D.; DICHEV, C.; AGRE, G.; ANGELOVA, G. Gamification in Education: A Systematic Mapping Study. *Journal of Educational Technology & Society*, v. 18, n. 3, p. 75–88, 2015.
- DUBOIS, D. J.; TAMBURRELLI, G. Understanding Gamification Mechanisms for Software Development. In: *ESEC/FSE*, Saint Petersburg, Russia: ACM, p. 659–662, 2013.
- FABBRI, S. C. P. F.; FELIZARDO, K. R.; FERRARI, F. C.; HERNANDES, E. C. M.; OCTAVIANO, F. R.; NAKAGAWA, E. Y.; MALDONADO, J. C. Externalising Tacit Knowledge of the Systematic Review Process. *IET Software*, v. 7, n. 6, p. 298–307, 2013.
- FRANKL, P. G.; WEYUKER, E. J. Testing Software to Detect and Reduce Risk. *Journal of Systems and Software*, v. 53, n. 3, p. 275 – 286, 2000.

- FRASER, G. Gamification of Software Testing. In: *AST Workshop*, Buenos Aires, Argentina: IEEE Press, p. 2–7, 2017.
- GARCÍA, F.; PEDREIRA, O.; PIATTINI, M.; CERDEIRA-PENA, A.; PENABAD, M. A Framework for Gamification in Software Engineering. *Journal of Systems and Software*, v. 132, p. 21–40, 2017.
- GRAHAM, D.; VAN VEENENDAAL, E.; EVANS, I. *Foundations of Software Testing: ISTQB Certification*. Cengage Learning, 2008.
- GRESSE VON WANGENHEIM, C.; SILVA, D. A. Qual Conhecimento de Engenharia de Software é Importante para um Profissional de Software? In: *Fórum de Educação em Engenharia de Software (FEES)*, Fortaleza, CE. Brasil, p. 1–8, 2009.
- HAMARI, J.; KOIVISTO, J.; SARSA, H. Does Gamification Work? – A Literature Review of Empirical Studies on Gamification. In: *HICSS Conference*, Waikoloa, HI, USA: IEEE, p. 3025–3034, 2014.
- HARROLD, M. J. Testing: A Roadmap. In: *Conference on the Future of Software Engineering - joint with ICSE*, Limerick - Ireland: ACM, p. 61–72, 2000.
- HUIZINGA, J. *Homo Ludens: Study of the Play Element in Culture*. Routledge Kegan & Paul, 1980.
- IEEE *IEEE 829 Standard for Software and System Test Documentation*. Standard 610.12, Institute of Electric and Electronic Engineers, New York/NY - USA, 1990.
- JESUS, G. M.; FERRARI, F. C.; PASCHOAL, L. N.; SOUZA, S. R. Is It Worth Using Gamification on Software Testing Education? An Experience Report. In: *Proceedings of the 18th Annual ACM Brazilian Symposium on Software Quality (SBQS)*, Fortaleza, Brazil: ACM, 2019.
- JESUS, G. M.; FERRARI, F. C.; PORTO, D. P.; FABBRI, S. C. P. F. Gamification in Software Testing: A Characterization Study. In: *Proceedings of the 3rd Annual ACM Brazilian Symposium on Systematic and Automated Software Testing (SAST)*, São Carlos, São Paulo, Brazil: ACM, 2018.
- JONES, C.; BONSIGNOUR, O. *The Economics of Software Quality*. Pearson Education, 2011.
- KAPP, K. *The Gamification of Learning and Instruction: Game-based Methods and Strategies for Training and Education*. Wiley, 2012.
- KATIE SALEN, E. Z. *Rules of Play: Game Design Fundamentals*. The MIT Press, 2004.

- KITCHENHAM, B. *Procedures for Performing Systematic Reviews*. Technical Report, 2004.
- KITCHENHAM, B. A.; BUDGEN, D.; BRERETON, P. *Evidence-based Software Engineering and Systematic Reviews*. Boca Raton, FL, USA: Chapman & Hall/CRC, 2015.
- KOSCIANSKI, A.; DOS SANTOS SOARES, M. *Qualidade de Software - 2ª Edição: Aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software*. Novatec, 2007.
- KOSTER, R. *Theory of Fun for Game Design*. O'Reilly Media, 2013.
- LAKATOS, E. M.; MARCONI, M. D. A. M. *Fundamentos de Metodologia Científica*. Editora Atlas S.A., 2010.
- LAURENT, T.; GUILLOT, L.; TOYAMA, M.; SMITH, R.; BEAN, D.; VENTRESQUE, A. Towards a Gamified Equivalent Mutants Detection Platform. In: *ICST Conference - Posters Session*, Tokyo, Japan: IEEE, p. 382–384, 2017.
- LIECHTI, O.; PASQUIER, J.; REIS, R. Supporting Agile Teams with a Test Analytics Platform: A Case Study. In: *AST Workshop*, Buenos Aires, Argentina: IEEE, p. 9–15, 2017.
- LYRA, K. T.; ISOTANI, S.; REIS, R. C. D.; MARQUES, L. B.; PEDRO, L. Z.; JAQUES, P. A.; BITENCOURT, I. I. Infographics or Graphics+Text: Which Material is Best for Robust Learning? In: *ICALT'16*, Austin, TX, USA, p. 366–370, 2016.
- MALONE, T. W. Toward a Theory of Intrinsically Motivating Instruction. *Cognitive Science*, v. 5, n. 4, p. 333 – 369, 1981.
- MÄNTYLÄ, M. V.; SMOLANDER, K. Gamification of Software Testing - An MLR. In: *PROFES Internl. Conference – Poster Session*, Trondheim, Norway: Springer, (LNCS, v. 10027), p. 611–614, 2016.
- MASTRODICASA, E. S. *A Gamification Framework for Software Engineering*. MSc Dissertation, Faculty of Informatics, Università della Svizzera Italiana, Lugano, Switzerland, 2014.
- MATHUR, A. *Foundations of Software Testing*. 2nd ed. Pearson Education India, 2013.
- MCGONIGAL, J. *Reality Is Broken: Why Games Make Us Better and How They Can Change the World*. Penguin Group, 2011.

- MYERS, G. J.; BADGETT, T.; SANDLER, C. *The Art of Software Testing*. 3rd ed. Hoboken/NJ - USA: John Wiley & Sons, 2011.
- OFFUTT, A. J.; PAN, J. Automatically Detecting Equivalent Mutants and Infeasible Paths. *The Journal of Software Testing, Verification, and Reliability*, v. 7, n. 3, p. 165–192, 1997.
- PARIZI, R. M. On the Gamification of Human-centric Traceability Tasks in Software Testing and Coding. In: *SERA Conference*, Towson, MD, USA: IEEE, p. 193–200, 2016.
- PARIZI, R. M.; KASEM, A.; ABDULLAH, A. Towards Gamification in Software Traceability: Between Test and Code Artifacts. In: *ICSOFTE Conference*, Colmar, France: IEEE, p. 1–8, 2015.
- PASCHOAL, L. N.; SILVA, L. R.; SOUZA, S. R. S. Abordagem Flipped Classroom em Comparação com o Modelo Tradicional de Ensino: Uma Investigação Empírica no Âmbito de Teste de Software. In: *XXVIII Simpósio Brasileiro de Informática na Educação*, SBC, 2017.
- PASSOS, E. B.; MEDEIROS, D. B.; NETO, P. A. S.; CLUA, E. W. G. Turning Real-World Software Development into a Game. In: *SBGAMES Symposium*, Salvador, Brazil: IEEE, p. 260–269, 2011.
- PEDREIRA, O.; GARCÍA, F.; BRISABOA, N.; PIATTINI, M. Gamification in Software Engineering - A Systematic Mapping. *Information and Software Technology*, v. 57, p. 157–168, 2015.
- PEDRO, L. Z. *Uso de Gamificação em Ambientes Virtuais de Aprendizagem para Reduzir o Problema da Externalização de Comportamentos Indesejáveis*. MSc Dissertation, Instituto de Ciências Matemáticas e de Computação, University of São Paulo, 2016.
- PETERSEN, K.; FELDT, R.; MUJTABA, S.; MATTSSON, M. Systematic Mapping Studies in Software Engineering. In: *EASE Conference*, Bari, Italy: The British Computer Society, p. 1–10, 2008.
- PINTO, F. S.; SILVA, P. C. Gamification Applied for Software Engineering Teaching-learning Process. In: *SBES'17*, New York, NY, USA: ACM, p. 299–307, 2017.
- PRESSMAN, R. S. *Software engineering: A Practitioner's Approach*. 7th. ed. New York/NY - USA: McGraw-Hill, 2010.

- ROJAS, J. M.; FRASER, G. Code Defenders: A Mutation Testing Game. In: *Mutation Workshop*, Chicago, IL, USA: IEEE, p. 162–167, 2016a.
- ROJAS, J. M.; FRASER, G. Teaching Mutation Testing using Gamification. In: *ECSEE Conference*, Seeon Monastery, Germany: Shaker Publishing, p. 1–5, 2016b.
- ROJAS, J. M.; WHITE, T. D.; CLEGG, B. S.; FRASER, G. Code Defenders: Crowdsourcing Effective Tests and Subtle Mutants with a Mutation Testing Game. In: *ICSE Conference*, Buenos Aires, Argentina: IEEE, p. 677–688, 2017.
- RYAN, R. M.; DECI, E. L. Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions. *Contemporary Educational Psychology*, v. 25, n. 1, p. 54 – 67, 2000.
- SHEPARD, T.; LAMB, M.; KELLY, D. More Testing Should Be Taught. *Communications of the ACM*, v. 44, n. 6, p. 103–108, 2001.
- SHETH, S.; BELL, J.; KAISER, G. *Increasing Student Engagement in Software Engineering with Gamification*. Tech. Report CUCS-018-12, Columbia University, New York, NY, USA, 2012.
- SIEGMUND, J.; SCHUMANN, J. Confounding Parameters on Program Comprehension: A Literature Survey. *Empirical Software Engineering*, v. 20, n. 4, p. 1159–1192, 2015.
- SMITH, J.; TESSLER, J.; KRAMER, E.; LIN, C. Using Peer Review to Teach Software Testing. In: *ICER'12*, New York, NY, USA: ACM, p. 93–98, 2012.
- SOSKA, A.; MOTTOK, J.; WOLFF, C. An Experimental Card Game for Software Testing: Development, Design and Evaluation of a Physical Card Game to Deepen the Knowledge of Students in Academic Software Testing Education. In: *2016 IEEE Global Engineering Education Conference (EDUCON)*, IEEE, p. 576–584, 2016.
- SPILLNER, A.; LINZ, T.; SCHAEFER, H. *Software Testing Foundations: A Study Guide for the Certified Tester Exam*. Rocky Nook, 2014.
- THE JOINT TASK FORCE ON COMPUTING CURRICULA *Curriculum guidelines for undergraduate degree programs in software engineering*. Technical Report, 2015.
- VALLE, P. H. D. *Jogos Educacionais: Uma Contribuição para o Ensino de Teste de Software*. MSc Dissertation, ICMC/USP, São Carlos, SP, Brazil, 2016.
- VALLE, P. H. D.; BARBOSA, E. F.; MALDONADO, J. C. CS Curricula of the Most Relevant Universities in Brazil and Abroad: Perspective of Software Testing Education. In: *SIIE'15*, p. 62–68, 2015.

- WERBACH, K.; HUNTER, D. *For the Win: How Game Thinking Can Revolutionize Your Business*. Wharton Digital Press, 2012.
- WOHLIN, C. 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 (EASE)*, London, UK: ACM, p. 1–10, 2014.
- WOHLIN, C.; RUNESON, P.; HÖST, M.; OHLSSON, M.; REGNELL, B.; WESSLÉN, A. *Experimentation in Software Engineering*. Computer Science. Springer Berlin Heidelberg, 2012.
- YUJIAN FU, P. E.; CLARKE, P. J. Gamification-Based Cyber-Enabled Learning Environment of Software Testing. In: *ASEE Conference & Exposition*, New Orleans, LA, USA: American Society for Engineering Education, p. 1–15, 2016.

Apêndice A

VISÃO GERAL DOS ESTUDOS SELECIONADOS

Esta seção apresenta uma visão geral de como a gamificação foi proposta e utilizada pelos autores dos estudos selecionados.

(García et al., 2017) propuseram o *framework* GOAL (Gamification on Application Lifecycle Management) com o objetivo de orientar processos que apoiem a introdução da gamificação em qualquer fase de desenvolvimento de software. Os autores realizaram um estudo de caso em uma empresa de desenvolvimento de software para investigar a viabilidade e aplicar o GOAL para integrar gamificação em outros ambientes de engenharia de software. Três áreas de processo - gerenciamento de requisitos, gerenciamento de projetos, e teste de software - foram gamificados seguindo a metodologia presente no *framework*. Os elementos de jogos escolhidos para serem utilizados foram: pontos, níveis, selos, quadro de classificação, gráfico social, e desafio. Os objetivos da gamificação não foram definidos no estudo de caso, uma vez que o objeto de estudo foi o próprio *framework* GOAL, mas a metodologia visa gamificar sistemas para engajar usuários e melhorar sua performance. Teste de unidade foi o único nível de teste mencionado no estudo de caso. Apesar disso, entende-se que qualquer técnica, nível, e fase do processo de teste pode ser gamificado utilizando o GOAL. Os resultados foram positivos, revelando benefícios providos pelo *framework* avaliado no contexto industrial.

CODE DEFENDERS¹, um projeto *open-source* disponível para jogar online, é um sistema gamificado de teste mutante usado para ensinar teste de software baseado em princípios de teste mutante. Além disso, pode também ser utilizado como uma abordagem crowdsourcing, ou um *framework* de avaliação por educadores. O jogo foi apresentado em quatro estudos (Clegg et al., 2017; Rojas e Fraser, 2016a,b; Rojas et al., 2017). Os objetivos comuns de gamificação nos quatro estudos foram: aumentar o engajamento, a motivação e o prazer em aprender teste mutante. Outros objetivos

¹Disponível em <<http://code-defenders.org/>> - Acessado em 12 de dezembro de 2019.

foram mencionadas em um ou mais desses estudos: produzir testes mais fortes (Rojas et al., 2017), melhorar as habilidades dos estudantes (Clegg et al., 2017; Rojas e Fraser, 2016b), melhorar o entendimento dos alunos (Rojas e Fraser, 2016b), promover a adoção de teste mutante entre desenvolvedores e profissionais, e melhorar o desempenho do testador (Rojas e Fraser, 2016a). Os elementos de jogos utilizados nos quatro estudos foram: pontos, combates e níveis. O quadro de classificação foi usado apenas em um estudo (Rojas e Fraser, 2016b), e equipe foi usada em dois estudos (Rojas e Fraser, 2016b; Rojas et al., 2017). Em relação a teste de software, a técnica (teste baseado em defeitos), nível (teste de unidade), e fases do processo (projeto de casos de teste, execução e avaliação, monitoramento e controle, e manutenção) foram os mesmos nos quatro trabalhos. Em seguida, apresenta-se as características variadas desses quatro estudos.

Rojas e Fraser (2016a) apresentaram um protótipo do CODE DEFENDERS com o objetivo de explorar maneiras de tornar os testes de mutação divertidos para aprender e aplicar. As hipóteses dos autores eram de que os jogadores criariam testes e mutantes mais fortes e teriam melhor desempenho nas tarefas de teste; nenhum resultado para as hipóteses foi apresentado porque uma avaliação empírica foi planejada para ser realizada em futuras versões do jogo. Em outro estudo, **Rojas e Fraser (2016b)** exploraram o “uso da gamificação para ensinar conceitos de testes de mutação”. Para isso, eles descreveram maneiras de fornecer experiência prática (por exemplo, puzzles, combates, auto-tutoria, recompensas) em um contexto educacional. A hipótese de pesquisa apresentada foi: “através do uso de um jogo de teste de mutação, os alunos serão capazes de compreender todos os conceitos relevantes de teste de mutação enquanto se divertem e, ao final, se tornarão desenvolvedores e testadores de software melhores, produzindo software de maior qualidade”. Os autores apresentaram alguns exemplos educacionais para fornecer experiência prática. Eles também mencionaram que o jogo pode servir como uma ferramenta de avaliação para educadores permitindo extrair feedback relacionado aos alunos. Nenhuma avaliação foi realizada. Mais recentemente, **Rojas et al. (2017)** apresentaram uma abordagem baseada em gamificação e crowdsourcing para produzir e obter dos jogadores que usam o CODE DEFENDERS testes e mutantes mais fortes; isto inclui a possibilidade de uso do jogo em contexto industrial, além de permitir terceirizar parte do trabalho dos desenvolvedores. Esta versão do jogo compreende um modo *single-player*, usa as ferramentas automatizadas EvoSuite² para gerar testes automaticamente, Randoop³ para responder a uma questão de pesquisa, e Major⁴ para criar mutantes automaticamente. Os autores também investigaram a viabilidade de gamificar o teste de software usando o CODE DEFENDERS, e

²Disponível em <<http://www.evosuite.org/>> - Acessado em 12 de dezembro de 2019.

³Disponível em <<https://randoop.github.io/randoop/>> - Acessado em 12 de dezembro de 2019.

⁴Disponível em <<http://mutation-testing.org/>> - Acessado em 12 de dezembro de 2019.

avaliaram o jogo como uma abordagem crowdsourcing. Para a primeira investigação, os autores avaliaram se testadores produzem testes melhores usando o jogo, e se eles preferem escrever testes enquanto jogam o jogo. Para a segunda, avaliaram se crowdsourcing leva a criar suítes de teste e mutantes mais fortes que as criadas por ferramentas de testes automatizados, e se a pontuação dos mutantes criados usando a abordagem crowdsourcing tem relação com a pontuação de mutantes tradicionais. Os resultados foram positivos para todas as questões; além disso, os autores perceberam a necessidade de incorporar novos elementos de jogos para melhorar o engajamento e a diversão dos jogadores. Por fim, **Clegg et al. (2017)** propuseram uma abordagem para ensinar conceitos de teste sistematicamente, e apresentaram alguns conceitos de teste para demonstrar como eles podem ser usados em CODE DEFENDERS. Neste estudo, o modo *single-player* também foi utilizado jogo; combates ainda existem, mas jogadores virtuais automatizados substituem o segundo jogador presente no modo multijogador; por fim, duas ferramentas automatizadas foram mencionados (como em (Rojas et al., 2017)): EvoSuite, que desempenha o papel de defensor gerando casos de teste automaticamente; por outro lado, Major desempenha o papel de atacante criando mutantes automaticamente, dependendo de qual papel o jogador humano escolher. Nenhuma avaliação foi realizada neste trabalho.

Anderson et al. (2015) apresentaram um ambiente de aprendizado chamado Learn2Mine⁵. O objetivo dos autores foi prover aos professores um ambiente onde pudessem ensinar *data science* através de exercícios de programação, e prover aos estudantes (tanto em ambiente acadêmico ou industrial) um sistema útil e agradável que os guie durante o processo de aprendizagem. As lições são adicionadas no sistema divididas em subproblemas; os estudantes devem resolvê-los até que, de forma incremental, alcancem o objetivo final de cada lição. Embora teste de software não seja o foco do curso, os subproblemas abrangem conceitos de teste de unidade. Assim, os estudantes acabam por aprender e aplicar testes de unidade em seus códigos para resolver o problema principal. Os alunos são recompensados após completarem cada lição e recebem *feedback* imediato após submeterem (ou re-submeterem quantas vezes desejarem) as suas soluções. Nenhum detalhe sobre técnicas de teste, níveis (exceto teste de unidade), ou fases do processo de teste foram dados, mas compreende-se que qualquer técnica, nível e fase pode ser ensinado usando o Learn2Mine, uma vez que mais lições podem ser adicionadas nele. O ambiente é gamificado com o uso de pontos, selos, quadro de classificação, e combates como elementos de jogos a fim de aumentar a diversão, engajamento, e motivação. Os estudantes não interessados na experiência da gamificação podem simplesmente ignorar tais aspectos porque os elementos foram pensados e projetados para serem discretos. Uma pesquisa foi aplicada para avaliar a

⁵Disponível em <http://learn2mine.appspot.com/> e <https://github.com/Anderson-Lab/Learn2Mine-Main> - Acessado em 12 de dezembro de 2019

usabilidade pedagógica e utilidade das características do Learn2Mine; todas as questões receberam respostas positivas, e o sistema de re-submissões recebeu as notas mais altas. Outro experimento foi realizado para comparar resultados de dois grupos de estudantes: um com acesso ao ambiente de aprendizado guiado e gamificado, e outro grupo sem a gamificação (mas ambos com acesso às re-submissões, correções e notas automáticas, e feedback imediatos). Os resultados revelaram que a performance do primeiro grupo foi significativamente melhor na resolução das tarefas em geral, apoiando as hipóteses dos autores de que um ambiente de ensino guiado e gamificado aumentam a taxa de realizações das tarefas. Alguns estudantes deram feedback tanto positivos quanto negativos sobre o uso do sistema, mas nenhum dos que usaram a versão gamificada deram opiniões apenas negativas. Portanto, os autores concluíram que o Learn2Mine é um ambiente benéfico no contexto educacional para ensinar cursos de *data science*, o que é claramente extensível a teste de software.

Yujian Fu e Clarke (2016) propuseram, avaliaram, e analisaram a efetividade do WReSTT-CyLE⁶ (Web-Based Repository of Software Testing Tutorials - a Cyberlearning Environment). Ele é um ambiente de aprendizado cibernético que usa gamificação como estratégia para engajar e motivar estudantes para aprenderem metodologias e ferramentas de teste de software, além de poderem melhorar seus conhecimentos. Os três elementos de jogos principais são pontos, selos, e quadro de classificação, mas a presença de níveis e a possibilidade de jogar em equipes também foram identificadas durante a leitura do trabalho apresentado pelos autores. Em relação a teste de software, os autores mencionam que WReSTT-CyLE contém tutoriais de ferramentas baseadas em plataformas IDE (*Integrated Development Environment*), e conteúdos para ensinar técnicas de testes funcional e estrutural. Embora nenhuma outra informação tenha sido dada sobre nível de teste e fase do processo de teste, compreende-se que quaisquer deles possam ser ensinados usando WReSTT-CyLE. Yujian Fu e Clarke avaliaram se há um relacionamento entre: os pontos virtuais dos estudantes e suas notas reais (a); pontos ganhos e número de postagens (b); e pontos virtuais e frequência de acesso ao curso (c). Os resultados mostraram que há um relacionamento em a, mas dois vieses foram identificados. Em b, pouca ou nenhuma correlação foi encontrada. Finalmente, c foi positivo, mas também com dois vieses. Portanto, os autores concluíram que WReSTT-CyLE é um eficiente ambiente de aprendizado, e que há um relacionamento entre gamificação e um ambiente online para engajar e motivar estudantes para aprenderem teste de software. Uma observação interessante que os autores mencionaram foi que os estudantes mais motivados foram os mais eficientes.

Dal Sasso et al. (2017) fizeram diversas contribuições em seu trabalho. Primeiramente apresentaram uma revisão do estado da arte para investigar como os elementos de gamificação podem ser usados em engenharia de software. Em seguida, apresenta-

⁶Disponível em <<http://wrestt.cis.fiu.edu/>> - Acessado em 12 de dezembro de 2019.

ram algumas definições relacionadas a jogos, gamificação, comportamentos, e citaram um exemplo em que a gamificação foi um fracasso, discutindo se o assunto é ou não uma causa perdida. Após isso, os autores discutiram princípios, promessas e perigos da gamificação. Dal Sasso et al. também propuseram um *framework* para gamificar sistematicamente qualquer atividade de engenharia de software, inclusive testes (razão pela qual justifica-se a inclusão deste trabalho no conjunto de estudos seleccionados). Depois disso, eles apresentaram um conjunto de *building blocks* como sendo a base do *framework*. Por fim, os autores demonstraram como usar o *framework*. Para isso, gamificaram dois sistemas e propuseram cinco métodos para avaliar a performance da gamificação. O *framework* proposto apoia o *design* de uma camada de gamificação integrada às ferramentas de trabalho já existentes, tanto para contextos educacionais quando industriais. Ele contém quatro seções (Atividade, Análise, Implementação, Teste) que apoia o processo de *design*. A avaliação realizada pelos autores envolve uma seção chamada "Teste". Esta avaliação visa verificar o quanto de sucesso os elementos de jogos apresentam quando aplicados em cada atividade. Exemplos concretos foram fornecidos para demonstrar como gamificar dois sistemas: *The Myth and De-Bug*, e *The Empire of Gemstones*. O primeiro é um sistema de rastreamento de *bugs*, onde os objetivos da gamificação foram melhorar a eficiência, o desenvolvimento da criatividade, melhorar a qualidade dos registros de *bugs*, estimular a participação da comunidade, e facilitar o processo de correção. Os elementos de jogos utilizados foram avatar, selos, quadro de classificação, pontos, e bens virtuais. O segundo sistema está no contexto de engenharia de software para revisão de código moderno. A principal diferença em relação aos elementos de jogos é a introdução de níveis para estimular a competição entre os usuários. O objetivo da gamificação neste exemplo foi melhorar a qualidade do código. Embora os autores tenham proposto métodos de avaliação, nenhuma foi realizada por eles.

Parizi (2016) defendeu que abordagens de rastreabilidade entre artefatos de testes e código precisam ser criados ou melhorados. Assim, o autor propôs mudar a recuperação existente de *links* de rastreabilidade baseada na busca pela causa do problema, para uma construção pró-ativa de sistemas de rastreabilidade com os profissionais altamente engajados. GamiTracify, um *framework* conceitual gamificado, foi criado para engajar e motivar desenvolvedores em tarefas de rastreabilidade resultando na melhoria da qualidade de *links* de rastreamento. O *framework* pode ser integrado a outras ferramentas e aplicado em um contexto industrial. A ideia é que os desenvolvedores de testes incorporem no código, ou mantenham, informações de rastreabilidade no momento em que estiverem desenvolvendo os testes, evitando assim pós-verificações de *links* recuperados em fases posteriores. Os autores realizaram uma avaliação empírica para verificar como o GamiTracify impacta a habilidade de se ter acurácia nas informações de rastreabilidade comparado com SCOTCH, uma outra abordagem

não gamificada. Eles consideraram as métricas *recall* (o número de *links* corretos) e precisão (porcentagem de falsos positivos) para quantificar a acurácia. Os resultados revelaram que as métricas *recall* e precisão usando o GamiTracify foram maiores que quando usando SCOTCH. Isto sugere, para os autores, que gamificação pode ter impacto positivo na motivação das equipes para alcançar acurácia. Os elementos de jogos utilizados para engajar e motivar os desenvolvedores de teste foram pontos, quadro de classificação, avatar, e desafios. A técnica de teste estrutural e o nível de unidade foram abordados no estudo. Além disso, GamiTracify abrangeu as fases de desenvolvimento e manutenção no processo de teste, nas quais as informações de rastreabilidade podem ser incorporadas ou extraídas, respectivamente. Este foi o único estudo selecionado que explicitamente definiu quais fases do processo de teste foram gamificados.

Liechti et al. (2017) discutiram e apresentaram um estudo de caso industrial sobre *test analytics*, compreendendo melhoria contínua, mecanismos de feedback, e teste automatizado. Os autores definiram *test analytics* como “análise de dados relacionados a teste a fim de prover *insights* úteis sobre a qualidade dos produtos e práticas ágeis, com o objetivo de apoiar um processo de melhoria contínua” (tradução própria). A equipe envolvida no estudo de caso definiu seu próprio objetivo: se tornarem melhores e mais consistentes em teste de software, o que também trariam outros benefícios. Os autores usaram o *Fogg Behaviour Model* (FBM) para desencadear duas mudanças de comportamento: a adoção do *Behaviour-Driven Development* (BDD), e a manutenção rigorosa de uma suíte de teste automatizada. Liechti et al. também desenvolveram uma plataforma *open-source* de *test analytics* para tornar as lições aprendidas dos projetos disponíveis a outras equipes. Esta plataforma funciona integrada às outras ferramentas capturando os resultados de testes produzidos. Baseados nestes resultados, os membros das equipes são recompensados. Os elementos de jogos utilizados foram pontos, selos, e quadro de classificação. Gamificação foi introduzida na plataforma como um motivador extrínseco para aumentar o prazer na realização das tarefas. Após o prazer inicial e a motivação causada pelos motivadores extrínsecos (elementos de jogos), os desenvolvedores se tornaram intrinsecamente motivados pelo fato de terem *feedback* visível e regular, onde podiam ver seus progressos e os resultados dos testes. Ou seja, visualizar seus resultados e performance os motivavam a continuarem realizando as tarefas. Em relação a teste de software, as técnicas de teste funcional e estrutural foram mencionadas; além disso, qualquer nível de teste (unidade, integração, e sistema) pode ser utilizado. As fases do processo de teste gamificadas foram projeto de casos de teste, execução e avaliação, monitoramento e controle, e manutenção. Um resultado interessante observado pelos autores foi que um desenvolvedor iniciante em teste automatizado se tornou “viciado” em realizar testes. Além disso, os impactos causados pelos experimentos nas equipes foram considerados significativos e dura-

douros. Embora o estudo de caso tenha sido realizado em uma *start-up*, acredita-se na possibilidade de se estender tanto o FBM quanto a plataforma para um contexto educacional.

Laurent et al. (2017) propuseram um sistema gamificado como uma plataforma de *crowdsourcing* para rotular mutantes equivalentes e avaliar diversos parâmetros envolvidos no processo de detecção. O protótipo é intencionado a coletar informações relacionadas à detecção de mutantes equivalentes, assim como coletar informações relacionadas à influência de atributos de usuários e mutantes no processo (de detecção de mutantes equivalentes). Assim, apenas os mutantes que corresponderem ao nível de habilidade do jogador serão apresentados a ele. A gamificação foi usada como forma de manter os usuários envolvidos, e os elementos do jogos utilizados foram pontos e quadro de classificação. Além disso, o protótipo gamificado é focado na atividade de teste de software, mais especificamente na técnica de teste baseada em defeitos, e no nível de unidade. As fases do processo de teste em que os usuários executam suas tarefas são o projeto de casos de teste, e execução e avaliação. Os autores pretenderam usar o sistema como uma plataforma de *crowdsourcing*, e mencionaram que também pode ser utilizado por desenvolvedores e testadores como uma ferramenta não integrada. Entende-se que o sistema apresentado é independente de outros sistemas; além disso, também pode ser usado para fins educacionais (tanto em escolas quanto na indústria) e industriais. Os autores pretendem realizar avaliações futuras, portanto, nenhum resultado foi apresentado.

O estudo de **Sheth et al. (2012)** é um pôster que apresenta o ambiente HALO (Highly Addictive, socialLly Optimized), um protótipo desenvolvido para avaliar seu benefício e viabilidade para os estudantes. Como o ambiente foi discutido tanto por Sheth et al. (2012) quanto por Bell et al. (2011), serão apresentados os detalhes no resumo do trabalho de Bell et al. (próximo parágrafo). A seguir, resumem-se somente os resultados do estudo piloto com estudantes de graduação conduzido por Sheth et al. (2012). Foi oferecido aos estudantes usarem o HALO de forma opcional nos deveres de programação. Os resultados revelaram que os estudantes que fizeram uso do HALO mostraram uma melhoria maior do que os que não utilizaram. Além disso, alguns alunos se sentiram mais confiantes e confortáveis para submeterem suas atividades depois de terem testado seus códigos. Um detalhe que se destaca no trabalho de Sheth et al. é o fato de que, diferentemente de Bell et al. (2011), os autores não fizeram uso dos elementos de jogos *conquistas* e *quadro de classificação*.

Bell et al. (2011) apresentaram uma metodologia chamada “*secret ninja*”, e propuseram combinar isto ao HALO, um ambiente engajador e colaborativo. Este ambiente pode ser integrado a uma IDE como um *plugin*. Combinando o HALO a uma adaptação do *secret ninja*, os autores visam motivar estudantes a testarem seus códigos, mas disfarçando o ensino dos conceitos de teste de software em histórias e missões.

Outros objetivos de usar o HALO são aumentar o engajamento ao longo do ciclo de vida do desenvolvimento de software, e melhorar o aprendizado dos estudantes. O objetivo da gamificação é adicionar prazer ao processo de aprendizado; assim, os autores esperam o aumento da produtividade e satisfação. Os elementos de jogos utilizados foram pontos, níveis, desafios, conquistas, quadro de classificação, e equipes. (Bell et al., 2011) se preocuparam em investigar em trabalhos futuros como evitar o desencorajamento dos estudantes devido ao sistema de rastreamento de recompensas (quadro de classificação). Além disso, defenderam o benefício do “vício” controlado para aumentar a produtividade. Como o HALO é uma ferramenta para ensinar teste de software, entende-se que qualquer técnica, nível, e fase do processo de teste pode ser ensinado, encorajando assim “uma base sólida de hábitos de teste, levando à disposição futura de testar tanto no curso quanto na indústria” (tradução própria), como mencionado pelos autores. Seus trabalhos futuros visam avaliar o impacto e benefício que o HALO pode causar aos estudantes.

Passos et al. (2011) propuseram e avaliaram uma abordagem para gamificar o processo de desenvolvimento de software, incorporando mecanismos semelhantes a RPG nas atividades cotidianas. Os autores diferiram seu trabalho de pesquisas anteriores argumentando que sua proposta não é incorporar a gamificação em uma mídia interativa ou ensino, mas sim em atividades do mundo real. Além disso, destacam que sua “proposta não é pretendida ao uso em sala de aula” (tradução própria). O objetivo da gamificação foi tornar os desenvolvedores mais conscientes do seu progresso, visando engajá-los em suas tarefas. A abordagem proposta baseia-se no *loop* desafio-punição-recompensa, onde os elementos de jogos utilizados foram pontos, selos (medalhas), níveis, equipes, desafios e conquistas. Um estudo de caso foi realizado em uma empresa para avaliar um conjunto de conquistas pré-definidas. Conquistas tanto individuais quanto em equipes foram estabelecidas e, com base nos resultados, o desenvolvedor (ou equipe) recebe uma medalha. A avaliação realizada revelou que o elemento *conquistas* ajudou a aumentar o engajamento e estimular a competição entre os desenvolvedores; serviu como uma forma de monitorar e controlar o processo de desenvolvimento; e pode ser usado como uma medida para comparar o desempenho estimado com o tempo real para realizar as tarefas. Os autores também propuseram gamificar uma ferramenta de gerenciamento de tarefas incorporando elementos de DevRPG, cujos objetivos são fornecer *feedback* imediato visando engajar os desenvolvedores nos desafios e estimular a competitividade entre eles. Nota-se que este estudo (Passos et al., 2011) indiretamente aborda testes de software, pois a proposta é aplicável para gamificar o processo de desenvolvimento de software, incluindo atividades de teste. Além disso, entende-se que qualquer técnica de teste, nível, fase do processo de teste pode ser gamificada pela aplicação dessa abordagem.

Dubois e Tamburrelli (2013) compararam sua proposta com a abordagem proposta por Passos et al. (2011) (gamificar o processo de desenvolvimento de software), e argumentaram que a principal limitação da abordagem de Passos et al. foi a falta de explicação em como usar a gamificação. Essa limitação, segundo os próprios autores (Dubois e Tamburrelli, 2013), é resolvida em seu estudo. Dubois e Tamburrelli propuseram uma abordagem para gamificar fases do ciclo de vida da engenharia de software visando melhorar a educação de engenharia de software, e melhorar a qualidade das atividades de desenvolvimento. Outros objetivos com a gamificação aplicada neste estudo foram aumentar o engajamento e motivação, melhorar o treinamento, e reforçar o monitoramento de pessoas envolvidas nas atividades de desenvolvimento de software. A abordagem pode ser integrada com outras ferramentas ou abordagens, e também pode ser aplicada em contextos tanto educacionais quanto industriais. Comportamentos de trapaça foram desencorajados pela identificação de formas para preveni-los. Os autores discutiram uma possível forma de aplicar sua abordagem apresentando como utilizar a gamificação em um contexto educacional; no entanto, o exemplo foi genérico em termos de elementos de jogos; Dubois e Tamburrelli citaram o mecanismo de recompensa, mas não definiram quais elementos concretos poderiam ser utilizados para recompensar um estudante. Os autores também observaram resultados positivos em cursos ministrados com a abordagem de gamificação. Dessa forma, um experimento foi realizado para compreender o efeito da competitividade entre grupos em duas sessões. Os autores apresentaram os resultados e afirmaram que não há evidência suficiente para concluir o papel da competitividade. Embora o trabalho dos autores não tenha foco em teste de software, sua natureza genérica justificou sua inclusão no conjunto final dos trabalhos seleccionados. Além disso, compreende-se que qualquer técnica, nível, e fase do processo de teste pode ser gamificado aplicando esta abordagem proposta pelos autores.

Apêndice B

INTRINSIC MOTIVATION INVENTORY - IMI (INGLÊS)

For each of the following statements, please indicate how true it is for you, using the following scale:

1	2	3	4	5	6	7
not at all true			somewhat true		very true	

1. While I was working on the task I was thinking about how much I enjoyed it.
2. I did not feel at all nervous about doing the task.
3. I felt that it was my choice to do the task.
4. I think I am pretty good at this task.
5. I found the task very interesting
6. I felt tense while doing the task.
7. I think I did pretty well at this activity, compared to other students
8. Doing the task was fun.
9. I felt relaxed while doing the task.
10. I enjoyed doing the task very much.
11. I did not really have a choice about doing the task.
12. I am satisfied with my performance at this task
13. I was anxious while doing the task.

14. I thought the task was very boring.
15. I felt like I was doing what I wanted to do while I was working on the task.
16. I felt pretty skilled at this task.
17. I thought the task was very interesting.
18. I felt pressured while doing the task.
19. I felt like I had to do the task
20. I would describe the task as very enjoyable.
21. I did the task because I had no choice.
22. After working at this task for awhile, I felt pretty competent.

-
14. Eu achei essa atividade muito chata.
 15. Eu senti que eu estava fazendo o que eu queria, enquanto eu estava fazendo a atividade.
 16. Me senti muito competente nesta atividade.
 17. Eu achei a atividade muito interessante.
 18. Eu me senti pressionado enquanto fazia a atividade.
 19. Eu senti obrigação de fazer essa atividade.
 20. Eu descreveria essa atividade como muito agradável.
 21. Eu fiz essa atividade porque eu não tive escolha.
 22. Depois de fazer essa atividade por um tempo, me senti muito competente.