



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Mutant Arena: Uma plataforma de jogos para o ensino
de testes unitários em *JavaScript*

Felipe Ribeiro Santos

Orientador

Prof. Dr. Márcio de Oliveira Barros

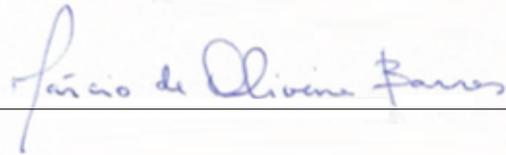
RIO DE JANEIRO, RJ - BRASIL
DEZEMBRO DE 2020

Mutant Arena: Uma plataforma de jogos para o ensino de testes unitários em *JavaScript*

FELIPE RIBEIRO SANTOS

DISSERTAÇÃO APRESENTADA COMO REQUISITO PARCIAL PARA OBTENÇÃO DO TÍTULO DE MESTRE PELO PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA DA UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO (UNIRIO). APROVADA PELA COMISSÃO EXAMINADORA ABAIXO ASSINADA.

Aprovada por:



Prof. Dr. Márcio de Oliveira Barros — UNIRIO

Prof. Dra. Regina Maria Maciel Braga Villela — UFJF

Prof. Dr. Gleison dos Santos Souza — UNIRIO

RIO DE JANEIRO, RJ - BRASIL

DEZEMBRO DE 2020.

Santos, Felipe Ribeiro.

A005 Mutant Arena: Uma plataforma de jogos para o ensino de testes unitários em JavaScript
/ Felipe Ribeiro Santos Santos, 2020.
77f ; 30 cm

Orientador: Márcio de Oliveira Barros

Dissertação (Mestrado em Informática) - Universidade Federal do Estado do Rio de Janeiro,
Rio de Janeiro, 2020. 1. Educação 2. Ensino 3. Jogos

4. Engenharia de Testes 5. Testes de Mutação

I. Barros, Márcio de Oliveira. II. Universidade Federal
do Estado do Rio de Janeiro, Programa de Pós-Graduação em Informática

. IV. Título.

CDD - 004.678

A Deus, o criador de todo céu e toda terra!

Agradecimentos

Agradeço, primeiramente, ao meu Deus que me deu condições de chegar até aqui!

Agradeço aos meus pais que sempre sonharam comigo e me incentivaram a batalhar pela minha formação.

Agradeço à minha esposa Jaqueline Santos que esteve comigo a todo momento. Ao meu filho Daniel que ainda não nasceu, mas já me inspira com sua força em cada chute na barriga da mamãe.

Ao meu irmão Esdras, sua esposa Thaianne e minha sobrinha Maria Clara que só trouxeram alegria nos dias mais estressantes.

Aos meus amigos que me incentivaram a seguir em frente e sempre torcem pelo meu sucesso.

Agradeço ao meu orientador Prof. Dr. Márcio Barros por ter me guiado nesta pesquisa e também pela oportunidade de aprendizado ao longo desse processo.

Aos participantes da banca Prof. Dra. Regina Maria e Prof. Dr. Gleison dos Santos, meus sinceros agradecimentos.

E também a todos os participantes do experimento desta pesquisa, meu muito obrigado!

Santos, Felipe Ribeiro. **Mutant Arena: Uma plataforma de jogos para o ensino de testes unitários em *JavaScript***. UNIRIO, 2020. 77 páginas. Dissertação de Mestrado. Departamento de Informática Aplicada, UNIRIO.

RESUMO

O uso de testes no ciclo de desenvolvimento tem contribuído fortemente para o aumento da qualidade no software. Porém, criar casos de teste não é uma tarefa fácil. Novas tendências tem surgido na literatura acadêmica onde o uso de elementos de jogos tem tornado a tarefa de elaboração de testes mais envolvente. Uma dessas tendências é o uso dos jogos no processo do ensino-aprendizado.

Este trabalho trata a construção de uma plataforma que tem como objetivo utilizar elementos de jogos para proporcionar uma maneira divertida para o aprendizado, bem como auxiliar na fixação dos conceitos de testes. Para isto, foi criada a plataforma *Mutant Arena*. Após o desenvolvimento, avaliamos o uso do jogo em dois grupos de jogadores utilizando a Metodologia de Avaliação de Jogos Educacionais - MEEGA+. O estudo mostra que o jogo é eficaz como apoio ao ensino dos conceitos de testes na Engenharia de Software.

Palavras-chave: Educação, Ensino, Jogos, Engenharia de Testes, Testes de Mutação.

ABSTRACT

The use of tests in the development cycle has contributed strongly to the increase the software quality. However, creating test cases is not an easy task. New trends have emerged in the academic literature where the use of game elements has made the task of designing tests more engaging. One of these trends is the use of games in the teaching-learning process.

This research deals with the construction of a platform that aims to use elements of games to provide a fun way for learning, as well as assist in fixing the concepts of tests. For this, the *Mutant Arena* platform was created. After development, we evaluated the use of the game in two groups of players using the Educational Games Evaluation Methodology - MEEGA +. The study shows that the game is effective in supporting the teaching of test concepts in Software Engineering.

Keywords: Education, Teaching, Games, Test Engineering, Mutation Tests.

Sumário

1	Introdução	1
1.1	Contexto	1
1.2	Definição do problema	1
1.3	Objetivo da pesquisa	2
1.4	Questão de pesquisa	2
1.5	Contribuições esperadas	3
1.6	Metodologia de pesquisa	3
1.7	Estrutura do texto	4
2	Fundamentação Teórica e Trabalhos Relacionados	5
2.1	Gamificação	5
2.2	Testes de Mutação	8
2.3	Gamificação na Engenharia de Software	10
2.4	A Metodologia de Avaliação de Jogos Educacionais MEEGA+	14
2.5	Considerações Finais	18
3	Mutant Arena - Gamificando o ensino de testes	19
3.1	A Plataforma Mutant Arena	19
3.1.1	Modo Instrutor	20

3.1.2	Modo Jogador	22
3.2	O Processo de Geração dos Mutantes	24
3.2.1	Registro dos Testes Unitários	25
3.2.2	Geração de Mutantes	25
3.2.3	Teste de Mutação e Resultados	27
3.3	Considerações Finais	29
4	Avaliação do Mutant Arena	30
4.1	Planejamento	30
4.2	Execução	32
4.3	Resultados	33
4.3.1	Resultados do primeiro grupo	33
4.3.2	Resultados do segundo grupo	38
4.4	Discussão dos Resultados	44
4.5	Considerações Finais	46
5	Conclusões	48
5.1	Contribuição	49
5.2	Limitações	49
5.3	Trabalhos Futuros	49
	Referências Bibliográficas	50
A	Formulário de Avaliação de Jogo	54

Lista de Figuras

1.1	Metodologia usada na Dissertação	3
2.1	Decomposição dos fatores de qualidade no modelo MEEGA+ [1]	16
3.1	Tela de cadastro da plataforma Mutant Arena utilizada no modo instrutor	21
3.2	Do lado esquerdo; a arena, o placar e quantidade de jogadas realizadas. Do lado direito; um menu de navegação possibilita o jogador a escrever seu código fonte, visualizar o código-fonte original da arena, visualizar o seu histórico de jogadas e visualizar um ranking geral da arena.	23
3.3	Relatório fornecido ao jogador para observar os avanços na escrita do seu teste unitário.	24
3.4	Exemplo do fluxo de execução do teste de mutação baseado no código-fonte.	24
4.1	Distribuição de gênero entre os participantes do Grupo 1.	34
4.2	Distribuição da faixa etária dos participantes do Grupo 1.	34
4.3	Distribuição de frequência de uso de jogos dos participantes do Grupo 1. . .	34
4.4	Gráfico de experiência dos jogadores do Grupo 1.	36
4.5	Gráfico de usabilidade dos jogadores do Grupo 1.	37
4.6	Distribuição de gênero entre os participantes do Grupo 2.	39
4.7	Distribuição de faixa etária dos participantes do Grupo 2.	39
4.8	Frequência de uso de jogos digitais entre os participantes do Grupo 2. . .	40

4.9	Gráfico de experiência dos jogadores do Grupo 2.	42
4.10	Gráfico de usabilidade dos jogadores do Grupo 2.	43
4.11	Gráfico de usabilidade dos jogadores de todos os grupos.	45
4.12	Gráfico de experiência dos jogadores de todos os grupos.	46

Lista de Tabelas

- 3.1 Operações de mutação que são aplicadas no código-fonte apresentado pelo instrutor quando cria uma arena para gerar os códigos mutantes que deverão ser identificados pelos casos de teste escritos pelos jogadores. . . 28

Lista de Códigos

2.1	Código original em linguagem JavaScript	9
2.2	Código com as variações (mutantes) em linguagem JavaScript	9
3.1	Exemplo de caso de teste para identificação do triângulo	25
3.2	Código sem mutação em JavaScript	26
3.3	Código com mutação em JavaScript	27
4.1	Código da arena representando uma função de soma em JavaScript	31
4.2	Código da arena com algoritmo para identificação do tipo de um triângulo em JavaScript	31
4.3	Código da arena com algoritmo para soma dos n primeiros termos de uma progressão aritmética em JavaScript	32

Lista de Abreviaturas

MEEGA Model for the Evaluation of Educational Games

UNIRIO Universidade do Estado do Rio de Janeiro

JS JavaScript

ES Engenharia de Software

1. Introdução

1.1 Contexto

O interesse pelo uso de jogos na educação se baseia na hipótese de que o uso de elementos de gamificação ajuda a motivar os alunos e, desta maneira, pode levar a melhores processos e resultados de aprendizado [2]. Diversas disciplinas da Engenharia de Software já contam hoje com o apoio de jogos que utilizam elementos de gamificação para o ensino [3].

Por outro lado, desenvolver software oferecendo confiabilidade ainda é um desafio. O teste de software desempenha um papel fundamental para aumentar a confiança nos produtos de software. Segundo Fraser et al. [4], 1,7 trilhões de dólares foram gastos com problemas de software em 2017 e poderiam ser evitados com mais testes. Parte deste desperdício está relacionado com o despreparo dos desenvolvedores em escrever testes de software, pois a disciplina de testes é pouco abordada na educação e poderia fazer parte do currículo nos cursos de Computação.

A motivação em buscar novas técnicas que auxiliem no aumento da qualidade dos software tem sido combustível para a criação de jogos que auxiliem os desenvolvedores a compreender como produzir testes [5]. A elaboração do teste é um fator que consome muito tempo e esforço no processo de desenvolvimento de software. Os testes podem consumir até 40% do orçamento do projeto [6], além de ser considerada uma tarefa tediosa e pouco motivadora.

1.2 Definição do problema

Um fator que contribui para a baixa qualidade do software está relacionado a maneira como os testes são abordados na educação. Segundo Fraser et al., os cursos de Enge-

nharia de Software tendem a focar nos aspectos de *design* e codificação [4]. Como consequência, diante de um problema em um trecho de código, se torna mais convidativo ao desenvolvedor reconstruir o código-fonte ao invés de criar testes para identificar os erros e corrigi-los.

Uma das alternativas para melhoria o ensino de testes de software é adotar ferramentas que introduzam conceitos de gamificação. O uso da gamificação e seus impactos têm sido amplamente discutidos na Academia e o uso de jogos educativos foi considerado uma das principais tecnologias adotadas nesta década como fonte de apoio à Educação [7].

Nesta Dissertação discutiremos o modelo, o desenvolvimento e a avaliação de um jogo com foco no ensino de testes de mutação para códigos escritos na linguagem de programação JavaScript. *JavaScript* é atualmente a linguagem de programação mais utilizada de acordo com uma pesquisa realizada com mais de 56 mil desenvolvedores conduzida pelo *Stack Overflow* [8]. A linguagem *JavaScript* é utilizada para construir aplicativos web, desktop e *mobile*, bem como software desenvolvidos para executar do lado do servidor.

1.3 Objetivo da pesquisa

O objetivo desta pesquisa é construir e avaliar a eficácia (em termos de engajamento e transferência de conhecimento) de um jogo com foco na disciplina de testes de software. A intenção é avaliar se o uso dos elementos de jogos proporciona uma maneira divertida para o aprendizado dos conceitos e processos relacionados a testes.

Como público alvo, miramos o uso do jogo por alunos de cursos de Computação e profissionais da área de Engenharia de Software onde não é necessário ter um conhecimento prévio sobre como construir testes em JavaScript.

1.4 Questão de pesquisa

A questão de pesquisa a seguir foi derivada do problema apresentado na Seção 1.3, estabelecendo o escopo e as direções de investigação utilizadas durante o desenvolvimento desta pesquisa.

- RQ1: Qual é a eficácia de um jogo com foco na aprendizagem de desenvolvimento de testes na linguagem *JavaScript*?

Após o estudo da literatura que versa sobre o uso de jogos no ensino de testes, foi percebido que havia uma oportunidade de tratar este problema com o uso de um jogo baseado na web que trouxesse uma boa experiência ao jogador combinada a boa usabilidade no jogo.

1.5 Contribuições esperadas

É possível destacar as seguintes contribuições desta Dissertação:

- A criação de um jogo (Mutant Arena) que tem o objetivo de ensinar conceitos relacionados ao desenvolvimento de testes para alunos e desenvolvedores de software, introduzindo esses conceitos de maneira descontraída no processo de aprendizado, reforçando o aprendizado e sendo uma ferramenta de apoio ao ensino.
- A avaliação do engajamento e o aprendizado de alunos de computação e profissionais da indústria de software.

1.6 Metodologia de pesquisa

A metodologia de pesquisa utilizada na execução deste trabalho pode ser resumida no esquema mostrado na Figura 1.1.



Figura 1.1: Metodologia usada na Dissertação

Inicialmente, foi realizada uma consulta na literatura de maneira *ad-hoc* para compreender como o uso da gamificação tem sido abordada na Engenharia de Software. A literatura revelou a oportunidade de criação de um jogo que incorporasse elementos de jogos com o objetivo de auxiliar na aprendizagem de testes de mutação utilizando a linguagem de programação *JavaScript* para a escrita dos casos de testes.

Considerando o estudo realizado para compreender os mecanismos de jogos utilizados nos jogos digitais, elaborou-se os requisitos para o projeto e construção do jogo. *Mutant Arena* se baseia em uma aplicação web, sendo desenvolvida em linguagem JavaS-

cript (Node.JS), utilizando banco de dados MongoDB, HTML e CSS para construção de páginas dinâmicas e módulos de processamento assíncronos.

Após a construção do jogo, realizou-se a avaliação do jogo utilizando o modelo de avaliação MEEGA+ com alunos de Mestrado e profissionais da indústria de software. Esta avaliação produziu diversos dados, que foram analisados segundo as dimensões e subdimensões do modelo MEEGA+ para avaliar a experiência do usuário e a usabilidade do jogo educativo.

Por fim, foi escrito este documento, que relata tanto o processo através do qual a pesquisa foi desenvolvida quanto as suas conclusões.

1.7 Estrutura do texto

Esta Dissertação está organizada em cinco capítulos. O primeiro capítulo compreende esta introdução.

O Capítulo 2 apresenta os fundamentos teóricos e trabalhos relacionados, discutindo os mecanismos de jogos, testes de mutação, como a gamificação tem sido abordada na Engenharia de Software e a metodologia de avaliação de jogos educacionais MEEGA+.

No Capítulo 3 é apresentada o jogo Mutant Arena, os modos de interação com o jogo, a mecânica de geração de mutantes e os elementos de jogos utilizados.

O Capítulo 4 apresenta os resultados da avaliação do jogo através do MEEGA+. Este Capítulo descreve o planejamento, a execução, os resultados e a discussão destes resultados segundo o modelo de avaliação de jogos selecionado.

O Capítulo 5 fecha a Dissertação com as conclusões da pesquisa, limitações do trabalho e perspectivas de trabalhos futuros.

2. Fundamentação Teórica e Trabalhos Relacionados

Este capítulo apresenta os conceitos utilizados no jogo *Mutant Arena*. Na Seção 2.1 apresentamos a gamificação e os elementos envolvidos no uso de jogos na educação. Os testes de mutação são tratados na Seção 2.2. Na seção 2.3 discutimos como o uso de jogos educativos tem sido abordado na Engenharia de Software. Em seguida na seção 2.4 apresentamos a metodologia de avaliação de jogos educacionais chamado MEEGA+. Por fim, a Seção 2.5 apresenta as considerações finais deste capítulo.

2.1 Gamificação

No contexto desta pesquisa utilizaremos o termo *gamificação* pois este foi utilizado em um maior número de trabalhos relacionados [9]. Gamificação, segundo Deterding et al. [10], é definido como “o uso de elementos de *design* de jogos em contextos não relacionados a jogos”. Isto significa dizer que a ação de utilizar esses elementos tem a função de se pensar como em um ambiente de jogo, utilizar as mecânicas e elementos de um jogo, porém em um contexto diferente de um jogo. Segundo Zichermann et al. [11] a gamificação utiliza a filosofia, elementos e mecânicas de um jogo em um contexto não relacionado a jogo para induzir o comportamento de uma pessoa, melhorando a sua motivação e o seu engajamento em uma determinada tarefa.

Para promover este engajamento, é importante considerar que tarefas que antes eram tratadas como entediadas passam a ter uma nova abordagem. O fator *tédio*, que antes poderia comprometer a motivação pelo sucesso, passa a ser tratado como um desafio. Com o uso da gamificação, estes desafios podem ser classificados em prioridades e ordenados em missões destinadas a ser cumpridas com uso de habilidades e o esforço coletivo. Por exemplo, podemos considerar que, para realizar uma atividade, um jogador precisa aprender novas habilidades no domínio da Engenharia de Software e usá-las de maneira combinada para conquistar um desafio e obter recompensas [12].

O uso da gamificação e seus impactos têm sido amplamente discutidos na Academia e o uso de jogos educativos foi considerado uma das principais tecnologias adotadas nesta década [7]. O termo “gamificação” teve sua origem na indústria de mídia digital, tendo o primeiro trabalho publicado em 2008 [10]. A comunidade científica passou a adotar este termo a partir de 2010, quando entendeu que havia um aumento na adoção e institucionalização dos jogos sociais e também o reconhecimento da sua influência nos jogadores [10].

A avaliação de quais elementos utilizar para a criação de um jogo na engenharia de teste pode ser uma tarefa difícil já que o uso de um determinado elemento pode descaracterizar o objetivo final do jogo. Nah et al. [13] identifica os principais elementos utilizados para fins educacionais na engenharia de software, porém destaca a falta de estudo em compreender a relação entre os elementos e o impacto que eles causam no processo de aprendizagem.

Werbach et al. [14] apresentam os elementos comumente utilizados na gamificação.

- **Pontos:** este elemento pode ser utilizado com diversos propósitos e possibilita o acompanhamento dos jogadores durante a interação com o sistema. Pode servir para estimular o jogador ou como parâmetro para que a equipe que desenvolveu o jogo possa acompanhar os resultados do jogador;
- **Níveis:** indicam o progresso do jogador dentro do jogo e podem ser utilizados como forma de controlar o ganho dos níveis de habilidade e conhecimento do indivíduo no sistema;
- **Placar:** o principal objetivo deste elemento é fornecer um meio para obter comparações. Geralmente é apresentado por um *ranking* (uma lista ordenada de dados com o nome dos jogadores e suas pontuações);
- **Divisas:** são elementos simbólicos – como distintivos – com o objetivo de marcar os objetivos e constantes progressos realizados por um jogador dentro do sistema;
- **Integração:** este elemento possibilita que um jogador inexperiente se sinta pertencente ao sistema. Este é um indicador que sinaliza o engajamento do jogador ao experimentar o jogo pela primeira vez. Os autores indicam que nos primeiros minutos dentro do ambiente deve-se: (a) revelar lentamente a complexidade do sistema; (b) reforçar o usuário de forma positiva; (c) criar um ambiente em que haja baixa possibilidade de falhas; e (d) deve ser possível aprender algo sobre o jogador. Os benefícios de se acertar neste processo podem garantir o engajamento em

longo prazo, pois este elemento tem o intuito de cativar e encorajar o indivíduo a permanecer no sistema, princípio este que é desconhecido nos primeiros contatos;

- **Desafios e missões:** são as informações dadas ao jogador para que um objetivo seja alcançado. O ideal é que os indivíduos, ao experimentar este ambiente, possam ter sempre algo interessante para realizar;
- **Loops de engajamento:** a criação e manutenção de emoções motivadoras sucessivas contribuem para que o jogador tenha um contínuo processo de engajamento na experiência do jogo;
- **Personalização:** pode ser caracterizada de várias formas e possibilita a transformação de itens do sistema pelo jogador;
- **Reforço e *feedback*:** servem para fornecer dados ao jogador, informando-o sobre onde se encontra no ambiente e o resultado de suas ações.

Os autores ainda apresentam doze mecânicas de jogos que são utilizadas em profundidade nos jogos atuais [14]:

1. **Reconhecimento de padrões:** é uma forma de interação dinâmica entre o indivíduo e o sistema. Durante o desenrolar do jogo, o jogador desvenda a complexidade que ambiente o proporciona. Para desvendar esta complexidade, o jogador busca reconhecer padrões que o auxiliem a organizar o ambiente do jogo;
2. **Coleta:** tem como base a característica humana de acumular materiais possivelmente úteis no futuro;
3. **Surpresa e prazer inesperado:** surpresas positivas durante o desenrolar do jogo geram sensação de prazer no indivíduo;
4. **Organização e criação de ordem:** alguns perfis de jogadores são atraídos com sistemáticas de organização de elementos ou conjuntos, principalmente quando são recompensados por isso;
5. **Presentes:** muitos jogos, principalmente sociais, utilizam presentes como mecanismo para promoção ou recrutamento de novos jogadores. Muitas dessas mecânicas consistem em dar algo a alguém para ganhar algo em troca;
6. **Flerte e romance:** servem como forma interessante (e simpática) de promover o envolvimento entre jogadores. Qualquer tipo de interação social, desde que não haja confronto, pode ser classificada neste item;

7. **Reconhecer para realizar:** consiste em compreender como funcionam as regras do ambiente para então realizar uma determinada tarefa;
8. **Liderança:** ferramentas em que a recompensa foca no poder de liderança do jogador;
9. **Fama:** a métrica deste item está ligada ao número de pessoas que veem, assinam, falam ou compartilham algo de alguém. Necessariamente, esta mecânica está ligada a influência sobre outros jogadores;
10. **Tornando-se herói:** esta mecânica é baseada em jogos populares baseados no conceito de resgate à princesa ou similares; O objetivo desta mecânica é oferecer desafios de complexidade gradual que ao final da missão central o jogador receba um prêmio;
11. **Status:** define hierarquias dentro dos sistemas. Essas hierarquias podem ser definidas por distintivos ou pela possibilidade de escolhas. De qualquer forma, o status pode ser utilizado como forma de recompensa;
12. **Nutrição de conhecimento:** conceito onde os jogadores devem ter que cuidar de certo ambiente para florescê-lo. Ou seja, a ideia é que o indivíduo invista e controle em determinadas ações para ter um resultado futuro. Por exemplo, jogos que simulam um ambiente de fazenda exigem que os jogadores plantem, cultivem e colham a safra para conquistar uma missão macro.

2.2 Testes de Mutação

Testes de mutação têm como objetivo adicionar falhas artificiais em um software, gerando versões deste software que são chamadas de *mutantes*. Dados um software A e sua suíte de testes de unidade T , são inicialmente gerados $m_A \in M_A$ mutantes introduzindo pequenas variações no código do software. Por exemplo, uma possível variação seria substituir um sinal de maior ($>$) por maior ou igual (\geq) em uma condição ou adicionar uma unidade a uma expressão aritmética.

Em seguida, os testes de unidade $t \in T$ são executados em cada um dos mutantes de M_A . O objetivo da suíte de teste T é “matar” os mutantes, fazendo com que todos os mutantes $m_A \in M_A$ produzam uma saída diferente do esperado pelo caso de teste quando executado sobre A . Dado um mutante $m_A \in M_A$ e um caso de teste $t \in T$, diz-se que t mata m_A se e somente se a saída de t em A for diferente da saída de t em m_A .

O escore da mutação é calculado como a razão do número de mutantes vivos sobre o número total dos mutantes. Ele fornece uma estimativa da abrangência e a qualidade da suíte de testes. Conhecendo o conjunto de mutantes vivos, é possível evoluir a suíte de testes para abranger mutantes que não foram descobertos. O teste de mutação foi proposto como uma alternativa às métricas tradicionais de cobertura de código para avaliar a qualidade da suíte de testes, julgando se é necessário criar novos testes ou aprimorar os existentes [15] para garantir uma boa cobertura da suíte.

A premissa subjacente ao uso de testes de mutação é que desenvolvedores tendem a escrever programas quase corretos, mas pequenas mudanças sintáticas podem representar falhas complexas no mundo real [15]. Assim, a criação automática de variações do software a partir de pequenas mudanças pode ajudar a medir a qualidade da suíte de testes na cobertura de “programas próximos” ao que foi desenvolvido. Um conjunto bem projetado de operadores de mutação pode resultar em testes poderosos, mas um conjunto mal projetado pode resultar em testes ineficazes. Existem evidências de que suítes de testes que conseguem encontrar mutantes também são boas para encontrar falhas reais [15].

A mutação é baseada em um conjunto de operadores relacionados à gramática da linguagem de programação [16]. Um exemplo de operador de mutação é o operador aritmético que altera uma declaração de atribuição como “ $x = a + b$ ” para uma variedade de alternativas possíveis, incluindo “ $x = a - b$ ”, “ $x = a \times b$ ” e “ $x = a / b$ ”. Para exemplificar os testes de mutação, o Código 2.1 apresenta um algoritmo para encontrar o menor valor entre duas entradas e o Código 2.2 apresenta um mutante deste código.

```
1 function min(valorA, valorB) {
2     let minVal;
3     minVal = valorA;
4     if (valorB < valorA) {
5         minVal = valorB;
6     }
7     return minVal;
8 }
```

Código 2.1: Código original em linguagem JavaScript

```
1 function min(valorA, valorB) {
2     var minVal;
3     minVal = valorB; // Mutacao
4     if (valorB < valorA) {
5         minVal = valorB;
6     }
7     return minVal;
```

Código 2.2: Código com as variações (mutantes) em linguagem JavaScript

2.3 Gamificação na Engenharia de Software

O potencial do uso de jogos na educação é baseado na hipótese que o uso dos elementos de gamificação ajuda a motivar os alunos e, portanto, pode levar a melhores processos e resultados de aprendizado [2]. Testes de software, que antes eram ensinados de maneira tradicional (isto é, sem o apoio de uma tecnologia voltada para o ensino), já contam com diversas experiências de ensino de maneira gamificada. Alguns trabalhos importantes foram publicados para identificar quais tem sido as principais dificuldades que a comunidade tem enfrentado ao lidar com as técnicas de gamificação no processo de aprendizagem [3].

Moreno et al. [3] mapearam publicações que abordaram o uso da gamificação no ensino das disciplinas na Engenharia de Software utilizando jogos digitais e não-digitais e com foco no ensino de processos. Eles analisaram cursos que não abordavam o uso da tecnologia como suporte ao aprendizado e que agora observam a satisfação dos alunos ao aprender de maneira divertida.

Os autores classificaram as formas de implementação da gamificação pela academia em quatro grupos: (i) quando um novo jogo é desenvolvida; (ii) quando se utiliza de um jogo já existente; (iii) o desenvolvimento de um *plug-in* para uma ferramenta que não possui fins de gamificação; e (d) quando nenhuma ferramenta gamificada (quando apresentava o uso de elementos de jogos) ou não-gamificada foi utilizada. O trabalho desenvolvido no contexto desta Dissertação se enquadra no primeiro grupo, com o desenvolvimento de um novo jogo.

CODE DEFENDERS¹ é uma aplicação Web gamificada usada para ensinar testes de software com base nos conceitos de teste de mutação. A aplicação foi apresentada em quatro estudos [4, 15, 17, 18]. Em cada publicação se percebe a busca dos autores pela compreensão do uso dos elementos de gamificação no jogo, que é avaliada através dos *feedbacks* dos alunos depois de utilizada. O principal objetivo do jogo é criar um ambiente onde duas equipes competem com perfis diferentes: enquanto uma equipe lidera o ataque ao software, a outra equipe é responsável pela defesa. Os invasores visam criar variantes de uma classe Java (ou seja, mutantes), com as quais eles “atacam” a capacidade

¹<http://code-defenders.org/> - último acesso em 15-Março-2020.

de detecção de falhas do conjunto de testes do software. Os defensores têm como desafio proteger a classe Java, escrevendo testes unitários que detectam os mutantes (a Seção 2.2 explica os testes de mutação).

Os autores estavam interessados em avaliar se o uso do jogo possibilitava os alunos produzirem testes melhores (com maior cobertura) e se preferiam escrever testes sem o uso do jogo. Para validar esses pontos, um estudo controlado foi conduzido com a participação de alunos de Graduação, Pós-Graduação (Mestrado e Doutorado) e profissionais da indústria. Antes de iniciar o experimento, os jogadores foram submetidos a uma sessão de treinamento, que consistia em introduzir conceitos sobre testes unitários, testes de mutação e uma visão geral do uso da CODE DEFENDERS.

O experimento foi conduzido em sessões onde cada participante atuava com um perfil diferente: em uma sessão atuava escrevendo testes unitários manualmente, em outra como atacante ou defensor dada uma coleção de classes Java. Os autores concluíram que os jogadores que utilizaram o jogo escreveram testes com cobertura em média 34.7% maior do que os testes gerados manualmente. Ao final, um questionário foi utilizado para compreender se os alunos preferiam utilizar o jogo para escrever os casos de teste. Os autores concluíram que mais de 60% dos alunos concordaram que escrever testes utilizando o jogo é mais divertido.

Em um outro estudo, Clegg et al. [18] exploram o uso da gamificação para ensinar testes de mutação e apresentam maneiras práticas sobre como utilizar elementos da gamificação (por exemplo tutoriais, recompensas e duelos) no contexto da educação. Os autores avaliaram se, através do uso de um jogo de teste de mutação, os alunos seriam capazes de entender os conceitos relevantes do assunto enquanto se divertem e, no final, se tornariam melhores desenvolvedores e testadores de software.

Diferente do CODE DEFENDERS, que foi aplicado em sala de aula em modo *multi-player*, Clegg et al. recomendam o uso em modo *single-player*, com o objetivo de incentivar que os jogadores progridam por níveis de dificuldade crescentes, onde cada nível cobre diferentes conceitos de teste. Cada nível consiste em uma coleção de quebra-cabeças sobre testes unitários, que são resolvidos usando ações de atacantes ou defensores – ações que fazem parte da mecânica do jogo. Cada quebra-cabeça, por sua vez, lida com um conceito de teste específico (por exemplo, cobertura de testes, fluxo de testes e testes de mutação), fornecendo uma falha que ilustra a insuficiência de um determinado conjunto de casos de teste e requer que o jogador forneça um teste adicional que resolva aquela falha ou que o jogador forneça uma falha que exponha a insuficiência do teste sendo ana-

lisado. Neste estudo, os autores apenas sugerem o uso das técnicas propostas, sem realizar análises exploratórias para apresentar conclusões.

Outro estudo que aborda o uso da gamificação no ensino de testes de software foi elaborado por Anderson et al., que apresentam um jogo chamado Learn2Mine ² com o objetivo de facilitar o ensino de programação em cursos de Ciências de Dados. O jogo pode ser utilizada tanto na academia como na indústria [19]. A sua mecânica insere os alunos em desafios nos quais é necessário resolver lições que são registradas como sub-problemas. Conforme os alunos resolvem os problemas, novos problemas mais difíceis lhes são apresentados. Resolvidos todos os problemas, a lição objetiva é considerada cumprida.

Embora o jogo não seja desenvolvida especificamente para o ensino de testes, os sub-problemas podem ser compreendidos como conceitos de teste de unidade e, desta maneira, os alunos acabam aprendendo e aplicando testes unitários em seu código para resolver os problemas e ganhar recompensas. Uma solução enviada pelo aluno pode não ser adequada para resolver determinado desafio e o jogo, utilizando um elemento de gamificação (o *feedback*), apresenta ao jogador uma avaliação do artefato que foi enviado.

Os autores utilizam outros elementos de gamificação (como pontos, emblemas e *rankings*) com o intuito de promover a satisfação ao jogar, o engajamento para aprender e a motivação. Para entender como o jogo influencia no aprendizado dos alunos, os autores elaboraram um *survey* para avaliar a usabilidade pedagógica e a utilidade das diferentes características do Learn2Mine. Todas as perguntas formuladas receberam resposta positiva e os autores concluíram que o jogo apresenta um ambiente benéfico para ministrar cursos de Ciência de Dados.

Hoyama et al. [20] apresentam um jogo gamificada, chamada CleanGame, que tem como objetivo capacitar alunos para a identificação de *Code Smells*. *Code Smells* são estruturas de código que devem ser removidas por atividades de refatoração para tornar o código mais legível, fácil de entender e simples de modificar. CleanGame é composto por dois módulos: (i) um questionário com perguntas de múltipla escolha para a identificação de *Code Smells* com o objetivo de ensinar aos alunos os principais conceitos; e (ii) um módulo com tarefas práticas para identificação de *Code Smells* no código-fonte.

Os autores avaliaram se o uso da gamificação causava um impacto positivo em como os estudantes identificavam *Code Smells*. O jogo foi avaliado com dois grupos de alunos depois que o conteúdo sobre *Code Smells* foi lecionado em sala de aula. O primeiro grupo

²<http://learn2mine.appspot.com/> - último acesso em 15-Março-2020

utilizou uma IDE, enquanto o outro grupo usou o jogo gamificada. Ao final, os autores solicitaram aos alunos que respondessem a um questionário dividido em três seções: (i) informações sobre os participantes; (ii) dados sobre a identificação de *Code Smells* (com e sem o CleanGame); e (iii) experiência dos alunos ao usar o CleanGame. Os autores concluíram que os alunos que utilizaram o jogo foram capazes de identificar aproximadamente o dobro de *Code Smells* do que os alunos que utilizaram apenas a IDE.

Moreira et al. [21] relatam a experiência do uso de um jogo não digital direcionado ao ensino do *framework* Scrum e práticas ágeis em sala de aula, onde o foco era abordar desafios sobre tomada de decisões e resolução de problemas em um contexto ágil de desenvolvimento de software. A adoção do *Scrum Card Game* permitiu simular um ambiente de projetos ágeis de software de curta duração. Cada equipe possuía de quatro a seis pessoas e assuntos como estimativa de requisitos, priorização e problemas com fatores técnicos e humanos eram tratados no desenrolar do jogo. Embora o conteúdo ensinado com o uso do jogo não estivesse no domínio da qualidade de software, o jogo tem como objetivo abordar técnicas e soluções para obter melhor qualidade do produto, como testes automatizados, programação em pares e colaboração com o *Product Owner*.

O jogo foi utilizado em sala de aula da seguinte forma: a instrutora criou uma tabela com duas colunas “Planejado” e “Realizado” no quadro branco e coletou de cada equipe as estimativas da primeira *sprint*, incluindo o número de histórias a serem concluídas e a produtividade (definida pela soma dos pontos de cada história). Depois que todas as equipes concluíram a *sprint*, a instrutora coletou o número de histórias concluídas e a produtividade real. No final do jogo, os instrutores conduziram uma discussão sobre os principais fatores que provocaram impacto no resultado das equipes e de que forma o jogo simula experiências de equipes trabalhando com metodologias ágeis.

Andrade et al. [22] apresentam um jogo não digital baseado em cartas, chamado *Gre-atTest*, que tem como objetivo desafiar os alunos, que assumem o papel de analistas de testes, a identificar o melhor tipo de teste em cenários de uso de aplicações. Os tipos de testes abordados foram: (i) teste de aceitação; (ii) teste de desempenho; (iii) teste de estresse; (iv) teste funcional; (v) teste de segurança; e (vi) teste de usabilidade.

O jogo é composto de três *decks* de carta, sendo um deles com desafios a serem solucionados, outro referente a bônus para resolução bem sucedida de desafios e um *deck* de jogo. O *deck* de jogo apresenta as cartas contendo os tipos de testes. A dinâmica envolve os alunos resolverem cinco desafios por rodada. Para isto, o aluno precisa selecionar uma carta de teste que seja compatível com a situação descrita no desafio escolhido. Se o teste estiver condizente, o jogador ganha pontos.

Os autores buscavam compreender se o uso do jogo apoiava no processo de aprendizagem e se despertava o interesse dos discentes pela área de teste de software. Para isto, realizaram dois tipos de avaliação: (i) aplicaram o jogo em sala de aula para alunos de Graduação e Pós-Graduação; e (ii) aplicaram o jogo com quatro professores especialistas no ensino de Engenharia de Software com o objetivo de coletar *feedback* sobre diferentes aspectos do jogo (como jogabilidade, conteúdo e apresentação das cartas), assim como sua aplicabilidade no ensino de teste de software.

O jogo permitiu que os alunos aprendessem e ganhassem confiança no aprendizado com o passar das rodadas. Alguns alunos afirmaram também que não apenas aprenderam, mas se sentiram motivados a buscar mais conhecimento sobre o tema. Outro aspecto analisado, e que teve resultados positivos foi a indicação do jogo para colegas, foi a possibilidade de jogar novamente. Os professores também relataram uso positivo em sua avaliação. Houve unanimidade nas repostas de todos quando se questionou se o jogo prendia a atenção dos participantes. Assim como no caso dos alunos, os professores concordaram que o jogo permitiu a cooperação e a competição entre os participantes. Os professores também concordaram que o jogo trata de um conteúdo relevante para o aprendizado.

Motivados pelo desafio do ensino de boas práticas em processos de desenvolvimento de software e as dificuldades de compreensão das principais normas e modelos que ajudam a criar esses processos (ISO/IEC 12207, ISO/IEC 29110, MR-MPS-SW e CMMI-Dev), Santos et al. [23] criaram um jogo não-digital chamado ProcSoft. O jogo é baseado em um tabuleiro e tem como objetivo capacitar os participantes a compreender quais são as atividades comuns a um processo de software (em relação com os papéis existentes em projeto de software) e as ferramentas de apoio que podem ser utilizadas. O *design* do jogo foi inspirado em um jogo popular, chamado Banco Imobiliário (*Monopoly*), e utiliza elementos de jogos como cartas, dados, peões e peças para marcação de propriedades.

Após a experimentação do jogo em sala de aula foi utilizada a metodologia de avaliação de jogos MEEGA. Os autores buscavam coletar feedbacks sobre a mecânica do jogo, design e a importância do seu uso no processo de aprendizagem. Os alunos responderam o questionário e os autores concluíram que houve impacto positivo na capacidade de aprendizagem e identificaram desafios que provocam a imersão dos alunos ao conteúdo.

2.4 A Metodologia de Avaliação de Jogos Educacionais MEEGA+

MEEGA+ [24] é um modelo para avaliação de jogos direcionados ao ensino no domínio da Ciência da Computação, incluindo jogos digitais e não digitais, como jogos de

carta ou tabuleiros. O seu objetivo é possibilitar a análise de jogos educacionais com o propósito de avaliar a percepção da qualidade em fatores.

MEEGA+ foi desenvolvido a partir do modelo MEEGA, elaborado por Savi et al. [25]. O modelo MEEGA consiste na aplicação de um questionário padronizado após os estudantes utilizarem um jogo. O questionário utiliza a abordagem GQM (*Goal/Question/Metric*) para definir um programa de medição para avaliar os jogos em três fatores: motivação, experiência do usuário e aprendizagem do ponto de vista do aluno. No questionário, cada fator é dividido em sub-dimensões relacionadas à percepção de uso do jogo [25]. Os fatores de interesse são descritos como:

1. **Motivação:** identifica se o jogo motivou o aluno nos estudos do conteúdo abordado. Inclui questões como: “O jogo prendeu minha atenção”, “O conteúdo é relevante ao aprendizado”, “O design do jogo é intuitivo e atraente”, entre outras. Este fator verifica o quanto a utilização do jogo engajou os alunos a utilizarem o jogo como fonte de aprendizado;
2. **Experiência do usuário:** analisa a interação do indivíduo com o produto por inteiro. Inclui questões como “Este jogo é adequadamente desafiador, as tarefas não são muito fáceis nem muito difíceis”, “Recomendaria esse jogo aos meus amigos”, “Pude interagir com as pessoas enquanto jogava”, entre outras. Entende-se a caracterização do envolvimento do aluno (jogador) com as tarefas do jogo, incluindo sua percepção de aprendizagem, sentimentos, prazeres e interações com o jogo, ambiente e outros jogadores;
3. **Aprendizagem:** avalia a aprendizagem dos alunos ao utilizar o jogo proposto. Inclui questões como “Quanto você acha que o jogo contribuiu para sua aprendizagem na disciplina?”, “Você acha que a experiência com o jogo vai contribuir para seu desempenho na vida profissional?”, entre outras. Envolve as percepções do efeito geral do jogo na aprendizagem dos alunos na disciplina.

No entanto, foi identificada uma sobreposição de conceitos entre os fatores de motivação e experiência do usuário na concepção do modelo MEEGA, além da falta de compreensão dos alunos em alguns itens do instrumento de medição. Assim Gresse et al. [1] elaboraram o modelo MEEGA+ e propuseram o uso de novos fatores e sub-dimensões conforme podemos observar na Figura 2.1. São eles:

1. **Atenção focada:** Avaliar a atenção, concentração, absorção e dissociação temporal dos alunos;

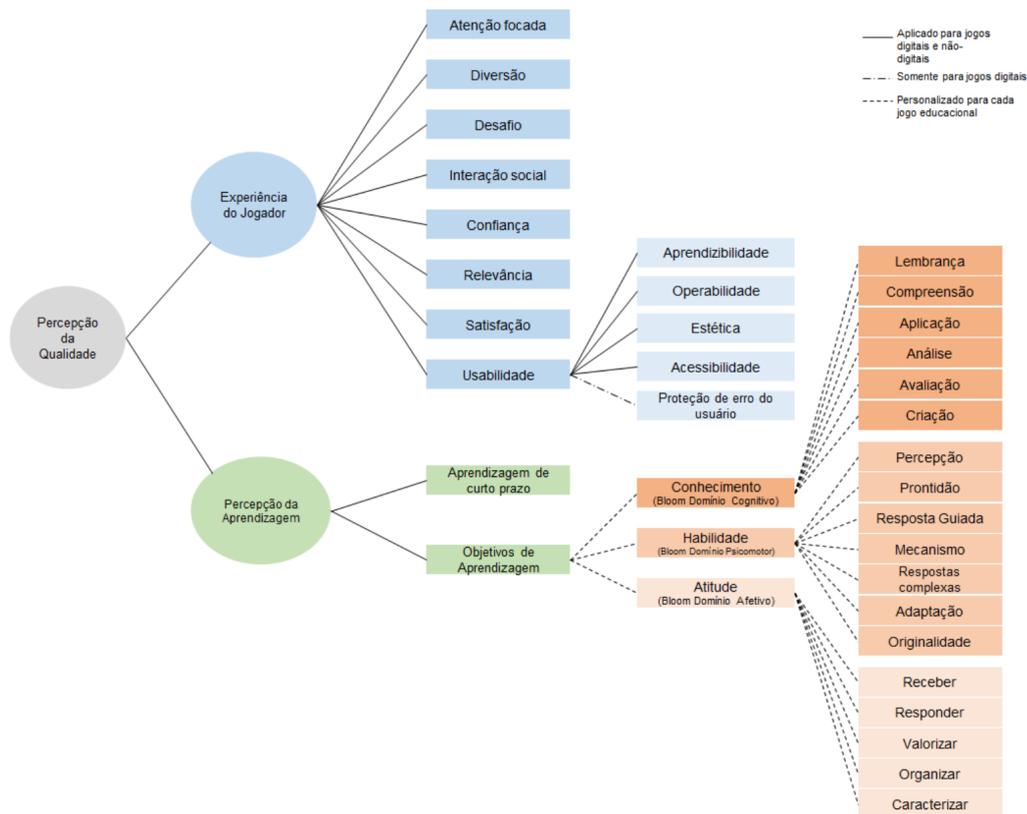


Figura 2.1: Decomposição dos fatores de qualidade no modelo MEEGA+ [1]

2. **Diversão:** Avaliar a sensação de prazer, felicidade, relaxamento e distração dos alunos;
3. **Desafio:** Avaliar se o jogo é suficientemente desafiador em relação ao nível de competência do aluno. Novos obstáculos e situações devem ser apresentados ao longo do jogo para minimizar a fadiga e manter os alunos interessados;
4. **Interação Social:** Avaliar se o jogo promove a sensação de um ambiente compartilhado e conexão com outras pessoas em atividades de cooperação ou competição;
5. **Confiança:** Avaliar se os alunos são capazes de progredir no estudo do conteúdo educacional por meio de seu esforço e habilidade;
6. **Relevância:** Avaliar se os alunos percebem que a proposta educacional é consistente com seus objetivos e vinculam o conteúdo ao futuro profissional/acadêmico;
7. **Satisfação:** Avaliar se os alunos sentem que o esforço dedicado ao jogo resulta em aprendizagem;
8. **Usabilidade:** Este fator divide-se nas subdimensões aprendizabilidade, operabilidade, estética, acessibilidade e proteção contra erros do usuário. Essas subdimensões avaliam a facilidade de uso do jogo;

9. **Aprendizagem percebida:** Avaliar as percepções do efeito geral do jogo na aprendizagem dos alunos na disciplina.

Para cada fator e subdimensão, os autores apresentam os questionários utilizados como instrumento de medição para avaliar jogos digitais e não-digitais. No entanto, o fator usabilidade é aplicável apenas para jogos digitais e não deve ser considerado em jogos não-digitais. Abaixo, exemplificamos uma parte do questionário do fator “Confiança” [1]. Os questionários são respondidos em uma escala Likert de 5 pontos, com alternativas de resposta que variam de discordo totalmente a concordo totalmente.

- “Quando olhei pela primeira vez o jogo, eu tive a impressão de que seria fácil para mim.”
- “A organização do conteúdo me ajudou a estar confiante de que eu iria aprender com este jogo.”

Muitos estudos que abordam o uso de jogos como fonte de apoio ao aprendizado têm utilizado o modelo MEEGA+ para avaliação da sua qualidade [20–22]. No estudo elaborado por Rossana et al. [22] o modelo MEEGA é aplicado para alunos e professores com diferentes objetivos de compreensão. Para isto, os autores propuseram variações no formulário utilizado pelos professores.

A abordagem com o corpo docente buscava compreender se a jogabilidade e o conteúdo abordado no jogo tinham real aplicação no ensino de testes de software. Os professores relataram que o jogo fornecia meios complementares de aprendizado ou fixação do conhecimento adquirido e que recomendaria o seu uso para outros professores aplicarem em suas disciplinas. Eles também concordaram que o *design* do jogo é intuitivo e atraente, prende a atenção, permite cooperação e competição entre os participantes. Para estes casos, fatores como *confiabilidade*, *usabilidade* e *atenção focada* foram utilizados no formulário fornecido aos professores.

A partir do formulário preenchido pelos alunos, os autores concluiriam que os alunos se sentiram motivados, que o jogo tinha um *design* intuitivo e atraente e que os alunos foram direcionados a lidar com um conteúdo relevante ao aprendizado, dado que puderam sentir confiança no aprendizado com o passar das rodadas. Estes pontos positivos levaram segurança aos alunos para indicar o uso do jogo para os seus colegas.

Moreira et al. [21] procuraram compreender a percepção dos alunos após utilizar o *Scrum Card Game* com o uso do MEEGA+. A avaliação da experiência dos jogadores foi,

de maneira geral, positiva. Quando houve indicação de discordância pelos estudantes, esta porcentagem foi baixa, menor do que 17%. Como exemplo, o fato do jogo seguir a mesma dinâmica ao longo das *sprints* gerou nos alunos a percepção de não serem apresentados a novos desafios e atividades. O *feedback* coletados pela a avaliação possibilitou os autores planejar as mudanças necessárias para o aumento do potencial de uso do jogo. Os fatores como usabilidade, confiança, desafio, interação social foram utilizados no questionário.

Hoyama et al. [20] investigaram a percepção dos alunos no uso do jogo *CleanGame* utilizando o MEEGA+. Por se tratar de uma aplicação *Web*, elementos como *ranking*, barras de progresso, pontuações e *status* foram utilizados em seu *design*. A análise dos dados permitiu concluir que os alunos tiveram percepções negativas em relação a interface visual do jogo – segundo os alunos um erro ocorria ao exibir a resposta correta e a quantidade de pontos de cada jogador.

O modelo MEEGA+ permite uma avaliação com pouca interrupção durante o processo instrucional, fornecendo um rápido *feedback* sem exigir conhecimento detalhado do instrutor sobre a teoria educacional, medição ou estatística. Para facilitar o seu uso, o modelo fornece um planilha que contém o questionário para coleta de dados e o resultado da análise desses dados permite entender a percepção dos alunos após utilizarem o jogo educativo. Estas informações, conforme observamos nos trabalhos mencionados, permitem conhecer os pontos a ser melhorados e compreender se o jogo atingiu o objetivo planejado.

2.5 Considerações Finais

O uso da gamificação na Engenharia de Software tem permitido a criação de jogos que contribuem com o aprendizado e o engajamento dos jogadores em contextos que antes eram ensinados de maneira tradicional. Sendo estes jogos digitais ou não-digitais, eles têm proporcionado uma nova forma dos alunos compreenderem os temas relacionados à Engenharia de Software.

A utilização destes jogos não se limita à academia, sendo elas também utilizadas na indústria de software. As publicações que utilizam jogos no apoio do aprendizado concluem que a utilização destes na educação proporcionam um ambiente divertido e engajador.

No próximo capítulo, será apresentada a proposta do *Mutant Arena*, um jogo que utiliza elementos da gamificação com o objetivo de engajar os alunos no aprendizado de testes de mutação em um ambiente desafiador e divertido.

3. Mutant Arena - Gamificando o ensino de testes

Esse capítulo apresenta a plataforma Mutant Arena, desenvolvida com o objetivo de promover o aprendizado sobre testes de mutação utilizando elementos de jogos em seu *design*. Na Seção 3.1 apresentamos a plataforma, os modos de uso do instrutor e do jogador e os elementos que caracterizam a plataforma como um jogo, seus mecanismos e telas. Na Seção 3.2 descrevemos o processo de geração dos mutantes utilizados como parte do jogo. Finalmente, na Seção 3.3 apresentamos as considerações finais deste capítulo.

3.1 A Plataforma Mutant Arena

A qualidade de um software é diretamente afetada pela criação ou ausência de casos de teste durante o seu desenvolvimento. Por outro lado, criar casos de teste não é uma tarefa fácil. Estudos recentes mostram que a elaboração de testes na forma de jogos tornou esta atividade mais envolvente do que abordar os testes de forma tradicional no ciclo de desenvolvimento de software [4, 17, 26, 27].

Um importante fator que contribui para a dificuldade de criação dos casos de teste está relacionado a como os testes de software são abordados na educação. Geralmente, os cursos de Engenharia de Software tendem a focar nos aspectos da criatividade em *design* e codificação de software, segundo Fraser et al. [26]. Assim, os esforços para codificação de um novo trecho de código, em vez de criar testes unitários para identificar erros e corrigi-los, se tornam muito mais convidativos.

O Mutant Arena é um jogo educacional que tem como objetivo ensinar conceitos relacionados ao desenvolvimento de testes para alunos e desenvolvedores de software. A plataforma introduz uma maneira informal e descontraída no processo de aprendizado com o uso de elementos de jogos. A ideia geral é consolidar os conceitos sobre testes e reforçar o aprendizado, fornecendo uma ferramenta de apoio ao ensino.

A plataforma foi elaborada baseando-se na motivação pelo aprendizado de testes de mutação, apoio na absorção do conteúdo e o despertar da disciplina de testes ao aluno. O jogo foi desenvolvido como uma plataforma Web com o objetivo de oferecer um *design* atraente no qual os jogadores possam aprender sobre o desenvolvimento de testes de maneira divertida em um ambiente competitivo.

O Mutant Arena, idealizado para uso em uma sala de aula, foi desenvolvido em linguagem JavaScript (Node.JS), utilizando banco de dados MongoDB, HTML e CSS para construção de páginas dinâmicas e módulos de processamento assíncronos para viabilizar o uso da plataforma sem demandar muitos recursos de processamento. O código está disponível em <https://github.com/gohackfelipe/mutantarena>.

Baseado no *design* do jogo “Space Invaders”, o objetivo do jogo é destruir uma série de mutantes de um código-fonte JavaScript escrevendo um código de teste unitário capaz de identificar estes mutantes. Os mutantes são gerados pela plataforma a partir de um código-fonte de referência apresentado por um instrutor. O jogador que submete mais rápido o teste unitário com maior cobertura se destaca em um *ranking*.

O elemento central do jogo é uma arena que possui um código-fonte principal e seus mutantes. Para isso, a plataforma trabalha com dois perfis de usuário: jogador e instrutor. O jogo não apresenta arenas previamente cadastradas, mas é possível configurar as arenas através da interface com o usuário no modo instrutor. As arenas devem ser configuradas a partir de um código na linguagem JavaScript. Após o registro, os mutantes são gerados de forma assíncrona e apresenta-se a quantidade de mutantes na tela de cadastro de arenas. Esta quantidade será utilizada como desafio para o jogador, na medida em que ele pode comparar o número de mutantes identificados pelos seus casos de teste contra o número total de mutantes.

3.1.1 Modo Instrutor

O modo instrutor é o perfil da plataforma Mutant Arena responsável pela configuração das arenas. Nesta interface (apresentada na Figura 3.1), a plataforma permite o cadastro, edição, atualização e exclusão das arenas.

O objetivo é permitir que o instrutor entre com um código-fonte JavaScript para o qual os jogadores deverão criar casos de teste. O jogador responsável pelo conjunto de casos de teste de melhor qualidade será considerado vencedor. Para medir a qualidade dos casos de teste, a plataforma recebe o código-fonte e gera um conjunto de mutantes, que devem ser identificados pelos casos de teste dos jogadores.

Para oferecer uma experiência agradável na escrita do código JavaScript que é a base para a criação dos mutantes, a plataforma possui uma área de edição de código-fonte utilizando IntelliSense¹. O instrutor também pode desenvolver o código-fonte JavaScript em uma ferramenta externa, copiar e colar em sua arena.

Fraser et al. [4, 15, 17, 18] propôs a plataforma Code Defenders com o objetivo de criar um ambiente onde duas equipes competem com perfis diferentes: enquanto uma equipe lidera o ataque ao software, a outra equipe é responsável pela defesa. Os invasores criam variantes de uma classe Java (ou seja, mutantes), com as quais eles “atacam” a capacidade de detecção de falhas dos casos de teste do software. Do outro lado, os defensores têm como desafio proteger a classe Java escrevendo testes unitários que detectam os mutantes.

Diferente da proposta elaborada por Fraser et al. [4, 15, 17, 18], a proposta do Mutant Arena é gerar os mutantes baseado nas operações de mutação listadas na Tabela 3.1. Os mutantes são gerados de maneira automática, sem a intervenção do jogador ou do instrutor. O processo de geração destes mutantes será apresentado na seção 3.2.

Para proporcionar uma experiência de desafio ao jogador, a plataforma permite o cadastro de múltiplas arenas com o objetivo na exploração da mecânica de missões (explicada na Seção 2.1). A combinação das arenas pode proporcionar ao jogador níveis progressivos de dificuldade, trazendo a sensação de objetivo alcançado (caso seja) e a configuração da competitividade entre os jogadores.

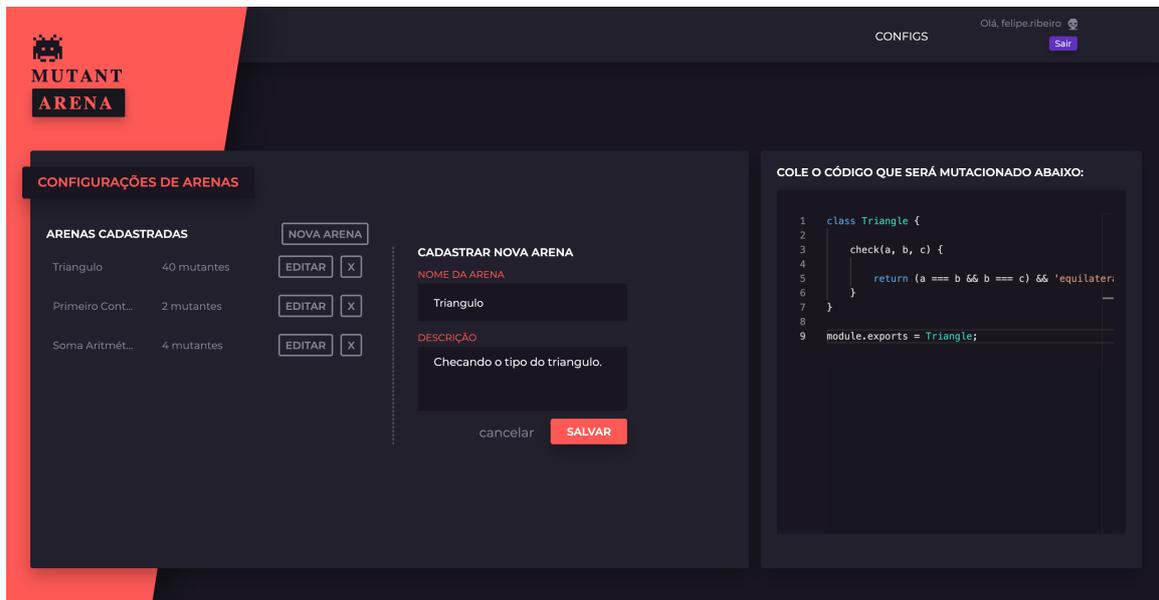


Figura 3.1: Tela de cadastro da plataforma Mutant Arena utilizada no modo instrutor

¹https://en.wikipedia.org/wiki/Intelligent_code_completion

3.1.2 Modo Jogador

O perfil do jogador apresenta a tela de jogo simbolizando uma arena de competição, com uma interface dinâmica com *design* de elementos que caracterizam a função de um jogo digital [10, 13, 14] e que foram apresentados na Seção 2.1. Para o Mutant Arena, foram desenvolvidos os seguintes elementos para caracterizá-lo como um jogo digital:

- O placar indica a quantidade de mutantes mortos pelo caso de teste apresentado pelo jogador em sua última jogada, assim como o número de sobreviventes;
- O número de jogadas é indicado juntamente com a relação entre os mutantes mortos e sobreviventes, isto é, uma fração entre mutantes e sobreviventes. Em um ambiente de sala de aula, acreditamos que esse dado fomenta a criação de um ambiente competitivo entre os jogadores, que poderão comparar o seu desempenho;
- O uso do *ranking* geral da arena permite o jogador não só saber se algum jogador eliminou todos os mutantes, mas se os seus casos de teste estão melhor ou pior comparado a outro adversário;
- Através do histórico de jogadas, o jogador compreende os avanços na escrita de seus testes unitários. Nesta aba, o Mutant Arena fornece um relatório detalhado sobre a execução do teste de mutação e os mutantes descobertos.

No lado esquerdo da Figura 3.2 apresentamos uma seção chamada *mutantes*, onde cada *card* representa um mutante gerado. Este *card* apresenta os mutantes com nomes ilustrativos seguindo o *design* do jogo “Space Invaders” e o seu estado atual, isto é, se o mutante está vivo ou morto. Ainda nesta seção, um *ranking* possibilita o jogador entender qual foi a relação de mutantes vivos e mortos na sua última jogada.

Do lado direito, conforme pode ser visto na Figura 3.2, o menu fornece ferramentas para o jogador criar as suas estratégias de jogo. O primeiro item do menu, chamado “Código”, possibilita ao jogador escrever o seu código de teste utilizando o *intelliSense*² em JavaScript e enviá-lo, através do botão “Enviar”, para análise pela plataforma. O objetivo do uso do *intelliSense* é proporcionar ao jogador um mecanismo de correção de sintaxe do código-fonte em JavaScript. Os recursos e técnicas de JavaScript como *promises*, *prototypes* e orientação a objetos, podem ser utilizados sem restrições.

O item “Arena” no menu permite ao jogador entender o código-principal da arena, já que os casos de testes precisam ter cobertura no código principal.

²https://en.wikipedia.org/wiki/Intelligent_code_completion

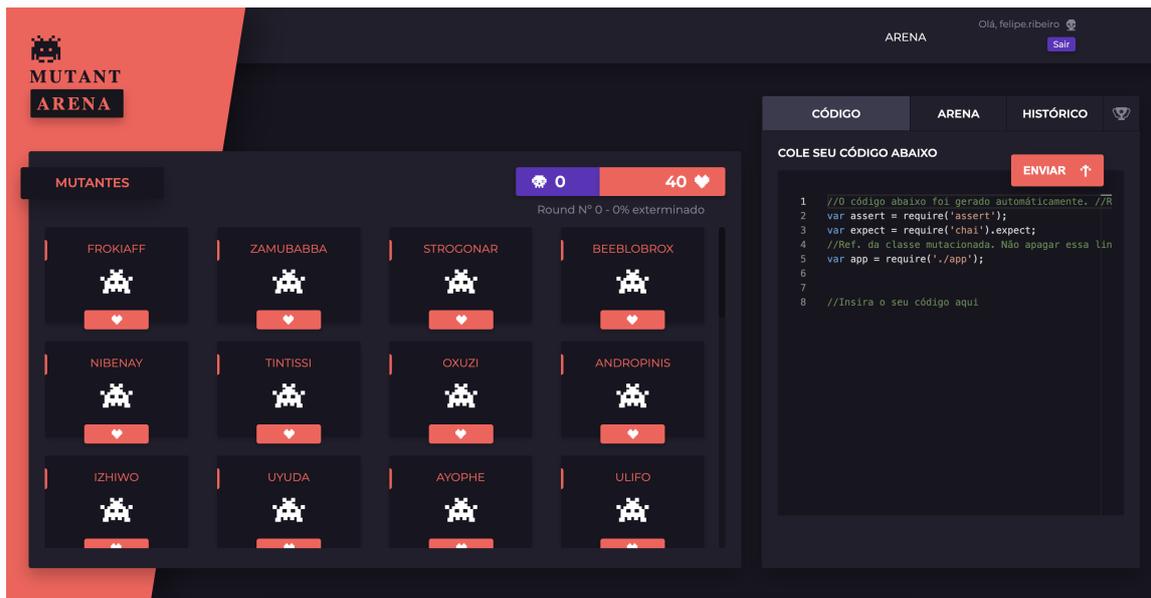


Figura 3.2: Do lado esquerdo; a arena, o placar e quantidade de jogadas realizadas. Do lado direito; um menu de navegação possibilita o jogador a escrever seu código fonte, visualizar o código-fonte original da arena, visualizar o seu histórico de jogadas e visualizar um ranking geral da arena.

O item “Histórico” no menu exibe uma lista com todas as jogadas do jogador, distinguindo-as entre jogada com “erro no código” ou processada com sucesso. O código é categorizado como “erro no código” quando o interpretador do Node.JS encontra um erro na sintaxe do código JavaScript. Por outro lado, quando a rodada é processada com sucesso, o botão “Ver” é habilitado e sua ação é redirecionar para a tela de relatório da Figura 3.3. Mais detalhes sobre os resultados desse relatório serão explicados na Seção 3.2.3.

O último botão do menu é um ícone em forma de troféu conforme a Figura 3.3. Nessa seção, a plataforma apresenta um ranking geral de todas as jogadas dos jogadores participantes da arena. Esse ranking é gerado baseado no jogador que submeteu, em menos tempo, o caso de teste com a maior cobertura.

O jogo exige que o jogador tenha uma experiência mínima na escrita de códigos em JavaScript. Porém, o uso de algumas técnicas mais avançadas dependerá da complexidade do código em JavaScript que o instrutor configurar na arena. Dado que o instrutor pode planejar as arenas de forma que técnicas mais sofisticadas sejam introduzidas conforme o avanço das missões, abordando assim muito mais que o ensino de testes unitários.

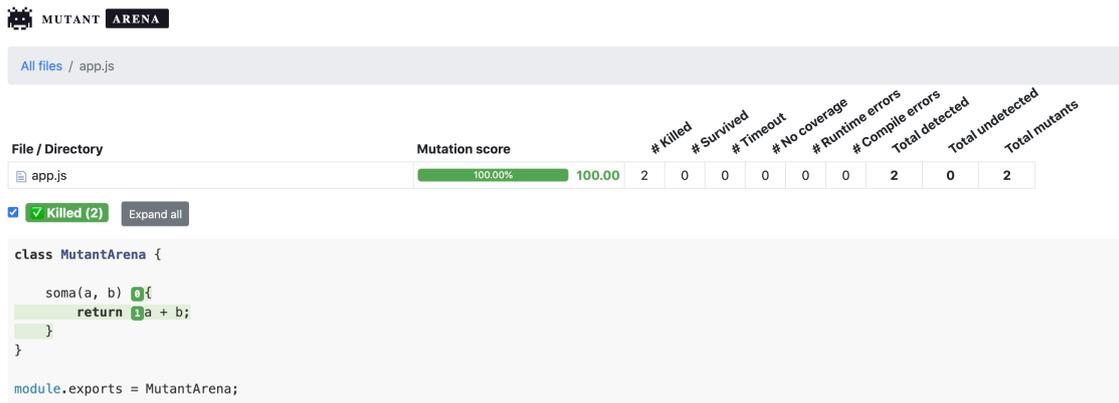


Figura 3.3: Relatório fornecido ao jogador para observar os avanços na escrita do seu teste unitário.

3.2 O Processo de Geração dos Mutantes

O processo de geração dos mutantes tem como objetivo inserir falhas artificiais (chamadas de mutantes) em um código-fonte. Os mutantes são gerados usando diferentes operadores de mutação, que sistematicamente adicionam modificações no código-fonte original e cada modificação adicionada resulta em um novo mutante.

A plataforma utiliza uma biblioteca chamada `stryker`³ para gerar os mutantes e processar os testes de mutação. A Figura 3.4 apresenta uma visão de alto nível deste processo, que será detalhado nas próximas subseções.

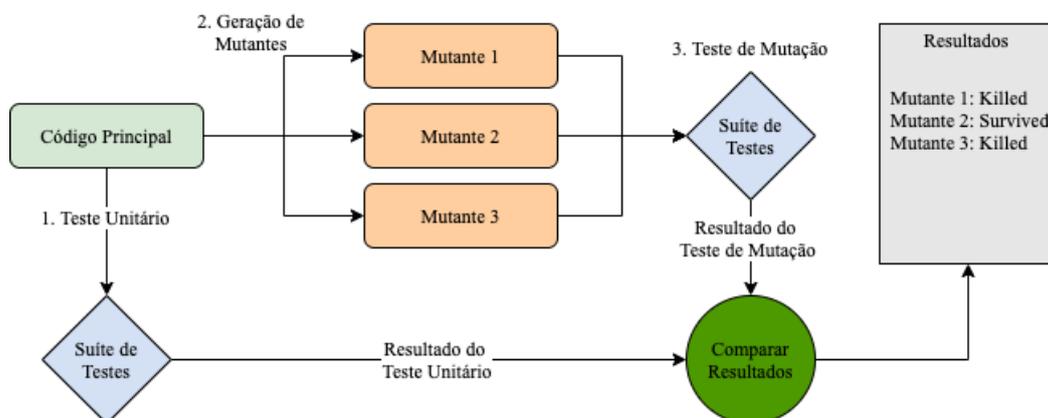


Figura 3.4: Exemplo do fluxo de execução do teste de mutação baseado no código-fonte.

³<https://github.com/stryker-mutator/>

3.2.1 Registro dos Testes Unitários

Conforme detalhamos na Seção 3.1.2, dado que o jogador submeteu os seus casos de teste na plataforma, estes serão utilizados para avaliar a qualidade da cobertura sobre os mutantes. O JavaScript possui diversas bibliotecas para escrita de testes. Para o uso no Mutant Arena selecionamos as bibliotecas de teste mais populares: *mocha*⁴ e *chai*⁵ [8]. No Código 3.1, temos um exemplo de casos de teste para um código que identifique o tipo de um triângulo dadas as dimensões dos seus três lados no Mutant Arena.

```
1 var assert = require('assert');
2 var expect = require('chai').expect;
3 var app = require('./app');
4
5 const geometry = new app();
6 describe('Trigonometry', function() {
7
8     it('testing - equilateral', function() {
9         expect(geometry.triangle(1,1,1)).equal('equilateral');
10    });
11
12    it('testing - isosceles', function() {
13        expect(geometry.triangle(1,2,2)).equal('isosceles');
14    });
15
16    it('testing - scalene', function() {
17        expect(geometry.triangle(1,2,3)).equal('scalene');
18    });
19 });
```

Código 3.1: Exemplo de caso de teste para identificação do triângulo

3.2.2 Geração de Mutantes

Uma vez que o código-fonte é fornecido pelo instrutor para configurar uma arena, conforme detalhado na Seção 3.1.1, a plataforma gera os mutantes sem que haja intervenção do instrutor. Para informar a quantidade de mutantes da arena na tela de instrutor

⁴<https://mochajs.org/>

⁵<https://www.chaijs.com/>

e jogador, conforme as Figuras 3.1 e 3.2, durante a configuração no modo instrutor a plataforma aplica o teste de mutação usando um código-fonte escrito em JavaScript.

Os mutantes são gerados usando diferentes operadores de mutação que sistematicamente adicionam modificações. Cada modificação adicionada representa um mutante. Para gerar esses mutantes, um processo é executado no *core* da plataforma com o intuito de buscar ocorrências de sintaxe no código da arena para cada item descrito na Tabela 3.1. Isto é, o código-fonte contido na arena é interpretado a fim de verificar a aplicabilidade das operações de mutação e, caso existam, a falha é adicionada. Porém, para cenários em que o código-fonte não seja trivial, muitos mutantes podem ser gerados.

Apesar de esforços na comunidade acadêmica e na indústria para a redução do número de mutantes produzidos, segundo Fraser et al. [4] as bibliotecas ainda geram uma quantidade grande de mutantes, o que demanda um grande tempo de processamento da detecção dos mutantes. Para reduzir o número de mutantes produzidos, a plataforma possibilita configurar a quantidade máxima de mutantes e os tipos de operadores de mutação a ser aplicados. Assim, além de reduzir o tempo de processamento necessário para a detecção dos mutantes, possibilita-se o uso da plataforma por um número maior de jogadores com uma infraestrutura de servidores reduzida.

Para exemplificar um cenário de mutação, usaremos o Código 3.2 com o propósito da identificação do tipo de um triângulo a partir dos tamanhos dos seus três lados. No Código 3.3 podemos observar uma variação baseada na operação de mutação binária onde os operadores “===” foram substituídos por operadores “!===”.

```
1 class Triangle {
2
3   check(a, b, c) {
4
5     return (a === b && b === c) && 'equilateral' ||
6           (a === b || a === c || b === c) && 'isosceles' ||
7           'scalene';
8   }
9 }
10 module.exports = Triangle;
```

Código 3.2: Código sem mutação em JavaScript

```

1 class Triangle {
2
3   check(a, b, c) {
4
5     return (a !== b && b !== c) && 'equilateral' ||
6           (a !== b || a !== c || b !== c) && 'isosceles' ||
7           'scalene';
8   }
9 }
10 module.exports = Triangle;

```

Código 3.3: Código com mutação em JavaScript

3.2.3 Teste de Mutação e Resultados

O teste de mutação consiste em executar a suíte de testes sobre os mutantes. Dizemos que um mutante sobreviveu quando sua falha não é identificada durante a execução de um caso de teste, ou seja, o caso de teste indica sucesso em um código que foi alterado por uma operação de mutação e é funcionalmente diferente do código original.

Por outro lado, um mutante é morto quando um caso de teste que é executado com sucesso no código-fonte original falha quando é executado no código-fonte do mutante. Neste caso, o caso de teste teve sucesso em identificar a divergência entre o código-fonte original e o código-fonte mutante gerado pela operação de mutação.

A pontuação de mutação, ou seja, a proporção entre mutantes mortos e sobreviventes fornece um indicador da qualidade do conjunto de testes. A tela contendo esses indicadores na plataforma no modo jogador é apresentada na Seção 3.1.2.

O relatório permite ao jogador: (i) entender o estado de cada mutante; (ii) filtrar a visualização por estado; e (iii) aperfeiçoar o seu caso de teste de acordo com os estados identificados. Cada métrica pode fornecer detalhes importantes para esse aperfeiçoamento e possui a seguinte definição:

- **Killed:** Quando pelo menos um teste falhou enquanto este mutante estava ativo, e assim, o mutante é morto;
- **Survived:** Quando todos os testes foram aprovados enquanto esse mutante estava ativo, e assim, o mutante sobreviveu;

Tabela 3.1: Operações de mutação que são aplicadas no código-fonte apresentado pelo instrutor quando cria uma arena para gerar os códigos mutantes que deverão ser identificados pelos casos de teste escritos pelos jogadores.

Tipo de Mutação	Exemplo	
	Código Original	Código Mutante
Aritmética		
Mutação em expressões com +, -	<code>a+b</code>	<code>a-b</code>
Mutação em expressões como *, / e %	<code>a/b</code>	<code>a*b</code>
Declaração de Arrays		
Arrays com tamanho são alterados com arrays vazios	<code>[a, 1 + 1]</code>	<code>[]</code>
Arrays vazios são preenchidos com arrays com tamanho	<code>[]</code>	<code>["value"]</code>
Construtores de arrays com tamanho são preenchidos com arrays vazios	<code>new Array(a, 1 + 1)</code>	<code>new Array()</code>
ArrowFunction		
Mutação de funções anônimas com somente uma linha de retorno	<code>const b = () =>4</code>	<code>const b = () =>undefined</code>
Block Statement		
Mutação do bloco de uma função com corpo vazio	<code>function() return 4;</code>	<code>function()</code>
Mutação de bloco simples	<code>const a = 3; const b = a;</code>	<code>const a = 3;</code>
Mutação do corpo de uma função anônima definida como bloco.	<code>const b = () =>return 4;</code>	<code>const b = () =>{}</code>
Mutação do corpo de uma instrução case se definida como bloco.	<code>switch (v) case 42: a = "spam"; break;</code>	<code>switch (v){ case 42: {} }</code>
Boolean		
Mutação de true para false	<code>true</code>	<code>false</code>
Variáveis com expressões !a	<code>!a</code>	<code>a</code>
Condições		
Operação ternária	<code>a < 3? b : c</code>	<code>false? b : c</code>
Mutação entre == e ===	<code>a == b</code>	<code>true</code>
Igualdade		
Mutação >e <	<code>a < b</code>	<code>a >= b, a <= b</code>
Mutação entre == e ===	<code>a == b</code>	<code>a != b</code>
Lógica		
Mutação entre && e	<code>a && b</code>	<code>a b</code>
Object		
Alteração na declaração de um objeto	<code>const o = { foo: "bar"}</code>	<code>const o = {}</code>
String		
Mutação em uma string com valor vazio.	<code>const b = "Hello world!"</code>	<code>const b = ;</code>
Unária		
Alteração em exp -a to +a	<code>~a</code>	<code>a</code>
Incremento/Decremento		
a++ para a-	<code>a++</code>	<code>a-</code>

- **Timeout:** A execução dos testes unitários com este mutante ativo atingiu um tempo limite. Por exemplo, o mutante entrou em um *loop* infinito durante a execução do teste;
- **No coverage:** Nenhum teste foi executado para este mutante;
- **Runtime Errors :** A execução dos testes resultou em um erro. Por exemplo, os casos de teste provocaram o lançamento de uma exceção `OutOfMemoryError`;
- **Compile Errors:** O mutante resultou em um erro de compilação. Esta métrica é utilizada apenas para linguagens compiladas.

3.3 Considerações Finais

Este capítulo apresentou a plataforma de jogo Mutant Arena, os perfis que podem ser assumidos por participantes do jogo e as principais funcionalidades disponíveis nos modos instrutor e jogador. As telas do jogo foram apresentadas e os elementos de jogos utilizados na plataforma. Também foi apresentado o processo de geração de mutantes.

O próximo capítulo apresentará a avaliação do jogo com desenvolvedores de software e o retorno dado por estes sobre a aplicação no ensino de testes. O método MEEGA+ será utilizado para realizar esta avaliação, de forma similar a outros jogos desenvolvidos no contexto da Educação em Engenharia de Software.

4. Avaliação do Mutant Arena

Neste capítulo, são apresentados o planejamento, a execução e os resultados obtidos na avaliação da plataforma Mutant Arena utilizando a metodologia MEEGA+. O objetivo desse modelo é analisar jogos educacionais digitais e não digitais com o propósito de avaliar a percepção da qualidade em termos de experiência do jogador e usabilidade do ponto de vista dos jogadores.

Na Seção 4.1 detalhamos o planejamento elaborado para aplicação do experimento aos jogadores. Na Seção 4.2 descrevemos a execução do experimento. Na Seção 4.3 apresentamos os resultados do experimento baseado no uso do MEEGA+. Na Seção 4.4 discutimos os resultados após a análise dos dados obtidos no experimento. Finalmente, na Seção 4.5 apresentamos as considerações finais deste capítulo.

4.1 Planejamento

Na etapa de planejamento, dada a situação do mundo em relação à pandemia pelo vírus SARS-COV-2 [28], estabelecemos o modelo de video-chamada para realização dos estudos experimentais do Mutant Arena. O grupo de participantes foi formado por alunos de Mestrado da UNIRIO e profissionais da indústria de software.

Foi elaborado um termo de consentimento livre esclarecido e um questionário relativo à qualidade do jogo (Apêndice A). Estes documentos foram desenvolvidos no Google Forms¹ e preenchidos pelos jogadores antes e depois de utilizar o jogo, respectivamente. Por se tratar de um jogo *multiplayer* (cada jogador joga de forma independente dos demais, mas existe competição entre os jogadores que estão na mesma arena), perguntas relacionadas a interação social foram incluídas no questionário esperando-se compreender uma avaliação dos alunos sobre a interação dos participantes no Mutant Arena.

¹<https://www.google.com/forms/about/>

A primeira seção do formulário consiste no termo de consentimento livre esclarecido. A segunda seção aborda questões sobre informações demográficas. A terceira seção apresenta uma breve descrição da proposta da plataforma e o link de acesso para jogá-lo. A quarta e última seção contém as perguntas referentes a qualidade do jogo (32 questões fechadas e 3 questões abertas). O participante deve responder às questões fechadas seguindo uma escala de Likert de cinco pontos, como “discordo totalmente” (-2), “discordo” (-1), “nem discordo, nem concordo” (0), “concordo (+1)” e “concordo totalmente” (+2).

Como o próprio autor desta Dissertação coordenou o experimento, com o objetivo de não expressar uma opinião próxima sobre a pesquisa, o questionário citado pelo ME-EGA+ [1] para uso do instrutor não foi utilizado.

Com o objetivo de prover uma experiência de aprendizado em sala de aula, utilizamos o *Google Meet*² para reunir os jogadores em uma sala virtual de modo que pudessem interagir e trocar experiências sobre o uso do jogo com suporte a video-chamada e compartilhamento de tela. Durante a experimentação, todo o contato foi *online* com o intuito de respeitar as regras de distanciamento estabelecidas pelos órgãos federais competentes.

Previamente foram cadastradas três arenas para proporcionar diferentes níveis de dificuldades para os jogadores. A arena com o código-fonte JavaScript apresentado no Código 4.1 tinha como objetivo apresentar as funcionalidades da plataforma ao jogador. A segunda e terceira arenas, com os códigos mais complexos apresentados nos Códigos 4.2 e 4.3, tinham como objetivo envolver o jogador no uso da plataforma de modo a que este pudesse, em seguida, informar as suas impressões em relação a este uso.

```
1 class MutantArena {
2   soma(a, b) {
3     return a + b;
4   }
5 }
6 module.exports = MutantArena;
```

Código 4.1: Código da arena representando uma função de soma em JavaScript

```
1 class Triangle {
2   check(a, b, c) {
3     return (a === b && b === c) && 'equilateral' ||
4           (a === b || a === c || b === c) && 'isosceles' ||
5           'scalene';
6   }
}
```

²<https://meet.google.com/>

```
7 }  
8 module.exports = Triangle;
```

Código 4.2: Código da arena com algoritmo para identificação do tipo de um triângulo em JavaScript

```
1 class PA {  
2     somaAritmetica(a1, an, n) {  
3         return n * (a1 + an) / 2  
4     }  
5 }  
6 module.exports = PA;
```

Código 4.3: Código da arena com algoritmo para soma dos n primeiros termos de uma progressão aritmética em JavaScript

4.2 Execução

A avaliação do jogo ocorreu em dois dias e com dois grupos diferentes. A primeira avaliação ocorreu no dia 31 de agosto de 2020 e contou com a participação de três jogadores e o instrutor responsável por orientar os alunos durante a execução da avaliação. Esta teve a duração de 70 minutos, dividida em 60 minutos para o jogo e 10 minutos para o questionário. A segunda avaliação aconteceu no dia 1º de outubro de 2020 e contou com a participação de quatro jogadores e o instrutor. Esta avaliação também teve a duração de 70 minutos, dividida em 60 minutos para o jogo e 10 minutos para responderem o questionário. É importante ressaltar que durante o uso do jogo não houve indisponibilidade devido ao uso da Internet ou qualquer tipo de intermitência que comprometesse o uso da plataforma, a comunicação entre os participantes e a avaliação da qualidade do jogo.

No início das duas rodadas de avaliação, o instrutor explicou aos jogadores a proposta do jogo e a motivação da pesquisa, disponibilizou o link para preenchimento do formulário e forneceu as devidas orientações quanto à execução da plataforma. Após a finalização da apresentação, os jogadores foram orientados a realizar o cadastro na plataforma e confirmar o seu cadastro através do link enviado para o e-mail do jogador. Em seguida, os jogadores efetuaram o *login* na plataforma e confirmaram a visualização das três arenas cadastradas previamente e descritas na Seção 4.1.

Dentro do tempo limite de jogo, os jogadores combinaram qual seria a primeira arena dado análise prévia do código-fonte de cada uma delas. O avanço para o próximo nível

era discutido entre os jogadores durante a execução do jogo. A sequência de arenas a ser jogada foi definida entre eles e coincidiu para ambos os grupos: foram as arenas de Código 4.1, 4.2 e 4.3.

Ao final do jogo os jogadores foram solicitados a responder o questionário contendo as perguntas de avaliação do jogo disponibilizado pelo MEEGA+ [1] com o objetivo de compreender a experiência de jogo e usabilidade proporcionada aos jogadores.

4.3 Resultados

Após a execução da avaliação, os dados foram organizados e analisados. O MEEGA+ disponibiliza uma planilha que auxilia na análise da qualidade do jogo. Para os fins do estudo, foram analisadas as dimensões **experiência do jogador** e **usabilidade**.

A dimensão “experiência do jogador” possui as subdimensões **confiança** (se a organização do conteúdo ajudou a sentir confiança no que iria aprender), **desafio** (se o jogo ofereceu desafio), **satisfação** (se recomendaria o uso do jogo para outras pessoas), **interação social** (se o jogo ofereceu momentos de cooperação), **diversão**, **atenção focada** (se houve envolvimento ao ponto de perder a noção do tempo), **relevância** (se o conteúdo do jogo é relevante para o aprendizado) e **percepção de aprendizagem** (se o jogo contribuiu para o aprendizado).

A dimensão “usabilidade” apresenta as subdimensões **estética**, **aprendizabilidade** (se o jogo é fácil de jogar), operabilidade (se o jogo é fácil de jogar e as regras são claras) e **acessibilidade**.

Os resultados serão apresentados para cada um dos dois grupos avaliados.

4.3.1 Resultados do primeiro grupo

Esse grupo foi composto por três participantes, entre eles um aluno de Mestrado da UNIRIO e dois profissionais da indústria de software. Nenhum dos participantes conhecia sobre testes de mutação antes de utilizar o jogo. As Figuras 4.2, 4.1 e 4.3 apresentam as informações demográficas dos participantes: 100% dos jogadores são homens, 67% dos jogadores estão na faixa etária de “29 a 39 anos” e 33% estão na faixa de “18 a 28 anos”. Sobre o costume de jogar jogos digitais, 33% dos participantes relataram jogar “Diariamente: jogo todos os dias”, 34% afirmaram “Raramente: jogo de tempos em tempos” e 33% disseram jogar “Mensalmente: jogo pelo menos uma vez por mês”.

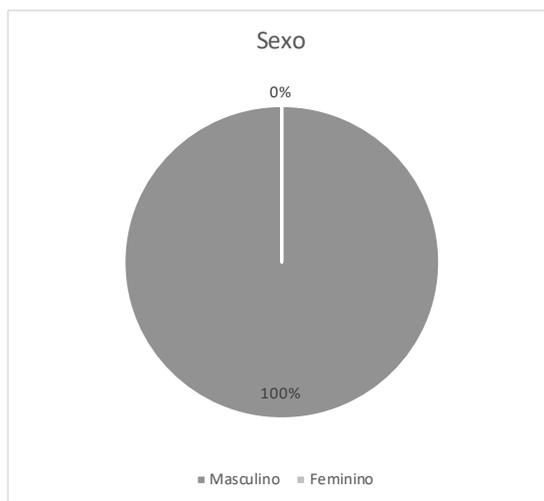


Figura 4.1: Distribuição de gênero entre os participantes do Grupo 1.

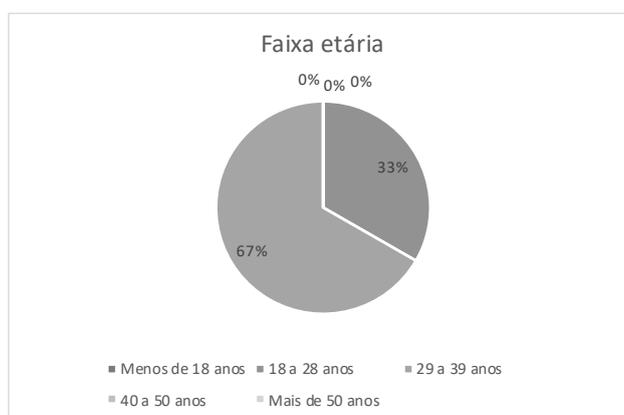


Figura 4.2: Distribuição da faixa etária dos participantes do Grupo 1.



Figura 4.3: Distribuição de frequência de uso de jogos dos participantes do Grupo 1.

A dimensão “experiência do jogador” foi bem avaliada pelos participantes. A Figura 4.4 apresenta a avaliação de experiência dos jogadores, na qual percebe-se um alto grau

de concordância com as afirmativas que orientam a discussão em todas as subdimensões da experiência do jogador. A mediana apresentada na última coluna está relacionada ao valor atribuído ao termo linguístico “Concordo totalmente” na escala Likert utilizada e corrobora o alto grau de concordância.

Na subdimensão **desafio**, um participante respondeu “indiferente” para a afirmativa “Este jogo é adequadamente desafiador para mim”. Os demais concordaram totalmente com esta afirmativa. A indiferença do primeiro jogador pode estar ligada ao fato de que, durante a execução do jogo, o mesmo relatou não saber desenvolver em *JavaScript* e também ao fato de um terço dos jogadores raramente jogar jogos digitais. Houve concordância total nas demais afirmativas desta subdimensão, o que leva a evidências de que o jogo foi considerado desafiador.

Quanto às subdimensões **satisfação** e **diversão**, os dois indicadores conseguiram 100% de concordância entre os participantes (dois concordaram totalmente e um concordou, no caso da subdimensão **satisfação**), indicando que o jogo proporcionou situações que trouxeram diversão ao realizá-lo. Dessa forma, podemos compreender que os jogadores desse grupo se sentiram desafiados ao prosseguir nas fases, satisfeitos com o conteúdo e acharam o jogo divertido.

Na subdimensão **interação social**, 100% dos jogadores concordaram totalmente ter interagido com outras pessoas durante o jogo e que o jogo gerou um ambiente competitivo, mesmo que o estudo experimental tenha sido aplicado por video-chamada. Isso nos traz indícios que mesmo em um ambiente virtual o jogo pode proporcionar momentos descontraídos com foco no aprendizado.

Quanto a subdimensão **confiança**, 100% dos jogadores concordaram totalmente que a organização do conteúdo ajudou a estar confiante sobre o aprendizado proposto. O mesmo ocorreu na subdimensão **atenção focada**, na qual 100% dos jogadores concordaram totalmente que a plataforma capturou a atenção ao ponto de perderem a noção do tempo e esquecer do ambiente ao redor enquanto jogava.

Na subdimensão **relevância**, 100% dos jogadores concordaram totalmente que esqueceram o ambiente ao redor enquanto jogavam o jogo e que o conteúdo é relevante para os seus interesses. Todos os jogadores também concordaram que o conteúdo do jogo está relacionado a disciplina de testes. Como temos jogadores profissionais da indústria de software, esse indicador nos traz indícios que o jogo poderia ser abordado no ambiente corporativo para o ensino de testes de mutação.

Na subdimensão sobre **percepção de aprendizagem**, dois jogadores concordaram totalmente e um jogador concordou relatando que preferem aprender com este jogo do que de outra forma. Todos os jogadores relataram que o jogo contribuiu para o aprendizado, que o jogo foi eficiente para a aprendizagem em comparação com outras atividades da disciplina e que o jogo contribuiu para o aprendizado sobre testes de mutação.



Figura 4.4: Gráfico de experiência dos jogadores do Grupo 1.

A dimensão “usabilidade” também foi muito bem avaliada pelos jogadores. A Figura 4.5 apresenta a avaliação de usabilidade dos jogadores, na qual percebe-se um alto grau de concordância com as afirmativas que orientam a discussão nas subdimensões de usabilidade, embora não tão alto quanto visto na dimensão “experiência do jogador”.

Observa-se, em especial, um maior número de respostas indiferentes e uma resposta em discordância na subdimensão **operabilidade**.

Na subdimensão **estética**, 67% dos jogadores concordam totalmente que o o *design* do jogo é atraente, os textos, cores e fontes combinam e são consistentes, enquanto 33% relatam concordar parcialmente.

Quanto à **aprendizibilidade**, 100% dos jogadores concordaram totalmente que precisaram aprender poucas coisas para poder começar a jogar o jogo e também concordam totalmente que aprender a jogar o jogo não foi uma tarefa difícil.

Na subdimensão **operabilidade**, 67% dos jogadores concordaram totalmente que o jogo é fácil de jogar e 33% relataram indiferença na resposta. 67% dos jogadores concordaram totalmente que as regras são claras e compreensíveis, porém um jogador discordou totalmente desta afirmação.

Quanto a **acessibilidade**, 67% concordaram totalmente que as fontes utilizadas no jogo são legíveis e as cores utilizadas no jogo são compreensíveis enquanto um jogador relata concordar parcialmente com estas afirmativas.

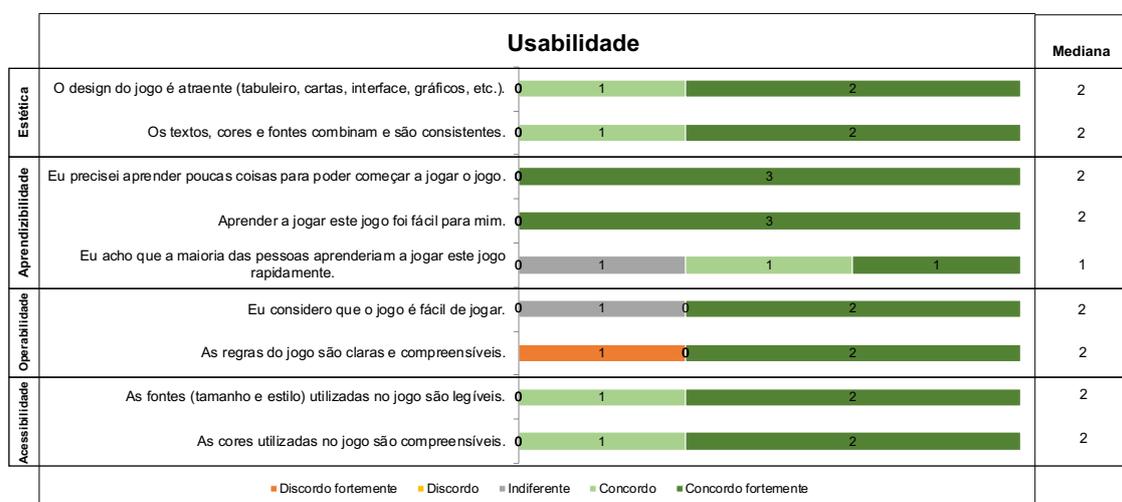


Figura 4.5: Gráfico de usabilidade dos jogadores do Grupo 1.

Quanto às questões abertas, na pergunta “O que você mais gostou do jogo?”, a maioria dos jogadores mencionou o *design* visual do jogo ser muito atrativo e o fato dos elementos de jogos incorporados trazerem uma boa experiência no seu uso. Abaixo, apresentamos alguns relatos dos jogadores:

“Eu gostei porque aprendi testes de uma maneira diferente.”

– Participante 1

“Esse ambiente competitivo me deixou na expectativa de saber se o mutante tinha morrido ou não.”

– Participante 2

“A dinâmica do jogo é bem interessante para entender o conceito.”

– Participante 3

Com relação aos pontos que poderiam ser melhorados no jogo, os jogadores citaram:

“O tamanho da janela de código poderia ser um pouco maior.”

– Participante 1

“Um tutorial interativo explicando como jogar.”

– Participante 2

“Disponibilizar exemplos opcionais para os usuários e dicas sobre testes.”

– Participante 3

É importante relatar que nenhuma falha na plataforma foi reportada durante o uso do jogo, o que significa dizer que a atenção em prover uma implementação de qualidade do jogo facilitou que o foco dos jogadores estivesse no objetivo do aprendizado, ao invés do relato de falhas. As melhorias citadas se referem a novas funcionalidades que poderiam ser implementadas na plataforma.

A sugestão do Participante 1 será resolvida implementando a possibilidade de maximização da janela de inserção do código-fonte, tanto na tela da arena no modo jogador quanto na tela de modo instrutor. A funcionalidade sugerida pelo Participante 2 tem como objetivo facilitar o entendimento sobre o uso da plataforma e suas funcionalidades em um primeiro contato. O Participante 3 sugere exemplos de implementação de testes e assim, uma seção ajuda será implementada na plataforma para atender a esta demanda.

4.3.2 Resultados do segundo grupo

O segundo grupo que analisou o jogo foi composto por dois alunos de Mestrado da UNIRIO e dois profissionais da indústria de software. Todos os participantes disseram não ter conhecimento prévio sobre testes de mutação.

As Figuras 4.6, 4.7 e 4.8 apresentam as informações demográficas dos participantes: 100% dos jogadores são homens, 50% dos jogadores estão na faixa etária de “29 a 39

anos” e 50% estão na faixa de “40 a 50 anos”. Sobre jogos digitais, 50% dos participantes relataram jogar “Raramente: jogo de tempos em tempos”, 25% afirmaram jogar “Mensalmente: jogo pelo menos uma vez por mês” e 25% disseram jogar “Semanalmente: jogo pelo menos uma vez por semana”.

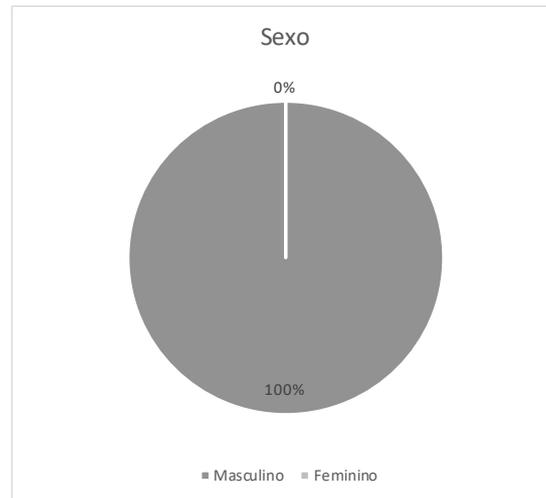


Figura 4.6: Distribuição de gênero entre os participantes do Grupo 2.

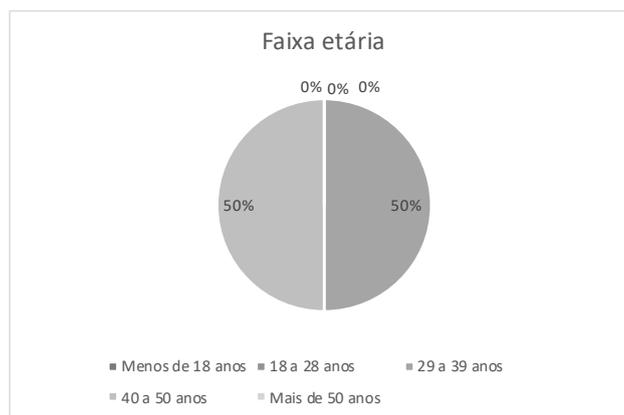


Figura 4.7: Distribuição de faixa etária dos participantes do Grupo 2.

Sobre a dimensão “experiência do jogador”, a figura 4.9 apresenta o gráfico de experiência dos jogadores do Grupo 2. Observa-se claramente uma maior diversidade de opiniões do que aquelas colhidas dos jogadores do Grupo 1.

na subdimensão **confiança** 100% dos participantes concordaram que a organização do conteúdo ajudou a estar confiante de que iriam aprender com o jogo.

Quanto ao **desafio**, 75% dos jogadores concordaram totalmente que o jogo é desafiador, 50% concordaram totalmente que o jogo oferece novos desafios e 75% concordaram que o jogo não se torna monótono em suas tarefas (repetitivas ou com tarefas chatas). Nesse grupo, diferente do Grupo 1 descrito na Seção 4.3.1, os jogadores não relataram

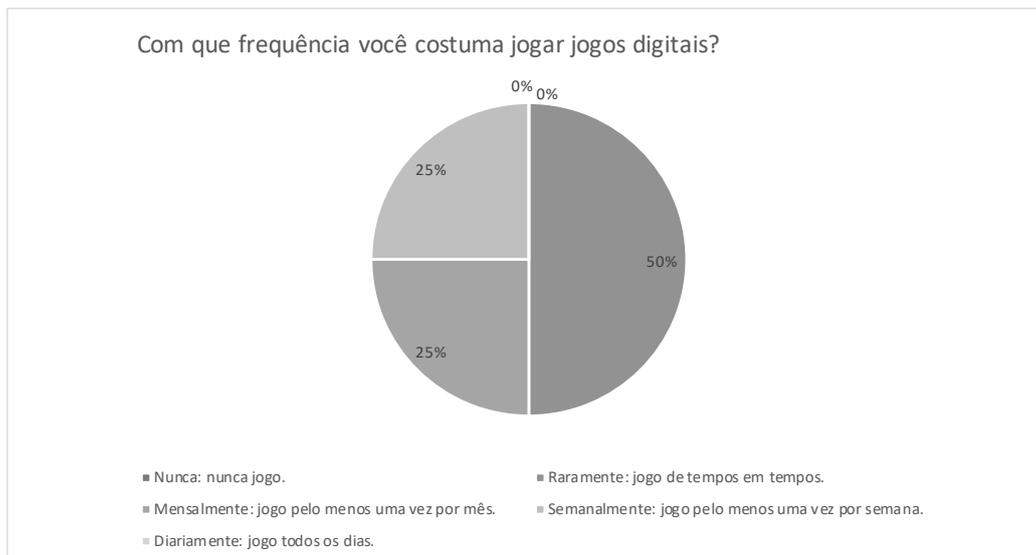


Figura 4.8: Frequência de uso de jogos digitais entre os participantes do Grupo 2.

dificuldades ao escrever códigos em *JavaScript*. Portanto, nos traz indícios de que o jogo exige ao jogador um conhecimento prévio na linguagem *JavaScript* para a escrita dos casos de teste. Os resultados dessa dimensão também nos permitem concluir que o jogo apresentou desafios e ritmo adequados, sem se tornar monótono em suas tarefas.

Quanto a **satisfação**, 75% dos jogadores concordaram totalmente relatando que ao completar uma tarefa sentiram-se realizados com o avanço, 75% concordaram (parcial ou totalmente) que conseguiram avançar no jogo devido ao esforço pessoal, 75% concordaram totalmente ao dizer que se sentiram satisfeitos com o que aprenderam no jogo e 75% dos jogadores concordaram totalmente que recomendariam o jogo para outros colegas. Durante o jogo, os alunos questionaram sobre a possibilidade de criar mais arenas para continuar o experimento além das arenas criadas. Porém não seguimos adiante com a criação de novas arenas devido ao tempo planejado para o uso da plataforma se aproximar do limite. Isso nos permite concluir que os alunos ficaram satisfeitos com o uso do jogo para o aprendizado.

Na subdimensão de **interação social**, 50% dos jogadores nem discordam e nem concordam que puderam interagir com outras pessoas durante o jogo, 50% dos jogadores concordaram totalmente que o jogo promove um ambiente de competitividade e se sentiram bem interagindo com outras pessoas durante o jogo. Essa dimensão traz indícios que o ambiente *multiplayer* e os mecanismos de jogos utilizados no Mutant Arena, proporcionaram um ambiente adequado para a interação social entre os jogadores.

Na subdimensão **diversão**, 75% dos jogadores concordaram totalmente que se divertiram com o jogo e 75% dos jogadores concordaram (parcial ou totalmente) que durante o

jogo alguma situação aconteceu que os fez sorrir. Com isso, temos evidências de que o jogo foi considerado um mecanismo de aprendizado divertido e indica que o jogo prestou o papel de proporcionar diversão enquanto interagia com os demais participantes mesmo em um ambiente de video-chamada.

Quanto a **atenção focada**, 50% concordaram totalmente e os outros 50% concordaram que houve algo interessante no início do jogo que capturou a atenção deles. 75% concordaram totalmente que estiveram tão envolvidos no jogo que perderam a noção do tempo e concordaram totalmente que esqueceram o ambiente ao redor enquanto jogavam o jogo. Sobre a afirmativa ‘Houve algo interessante no início do jogo que capturou a minha atenção’ ter sido bem avaliada, um dos jogadores relatou que foi interessante antes de iniciar o jogo, entender quais eram os códigos das arenas. Esse entendimento inicial ajudou os jogadores a escolherem qual seria a primeira arena a iniciar o jogo, além de trazer o foco dos jogadores na plataforma.

Na subdimensão **relevância**, 75% concordaram totalmente que o conteúdo é relevante para os seus interesses. 75% concordaram que o conteúdo do jogo está relacionado a disciplina de testes e que poderiam ser também abordados na indústria de software. 50% dos jogadores concordaram totalmente relatando que o jogo é um método adequado para a disciplina e 75% concordaram relatando que é claro como o conteúdo do jogo está relacionado com a disciplina. Desse modo, esta dimensão nos traz indícios que o jogo apresenta benefícios como uma ferramenta de ensino para a disciplina.

Quanto a subdimensão sobre **percepção de aprendizagem**, 50% concordaram totalmente que preferem aprender com o jogo do que de outra forma, que houve contribuição para a aprendizagem e que o jogo foi eficiente para a aprendizagem. 50% dos jogadores concordaram totalmente e 50% concordaram que o jogo contribuiu para o aprendizado sobre testes de mutação.

Com respeito à dimensão “usabilidade”, o jogo obteve resultado positivo, ainda que com sinais mais fracos do que os identificados para o Grupo 1. A Figura 4.10 apresenta o gráfico de usabilidade dos jogadores.

Na subdimensão **estética**, 50% dos jogadores concordaram que o *design* do jogo é atraente e os outros 50% concordaram totalmente. Três dos quatro jogadores concordaram totalmente que as fontes e cores combinam e são consistentes enquanto um jogador relatou concordar.

Quanto a **aprendizibilidade**, um jogador discordou sobre precisar aprender poucas coisas para poder começar a jogar devido a raramente jogar jogos digitais. Por outro

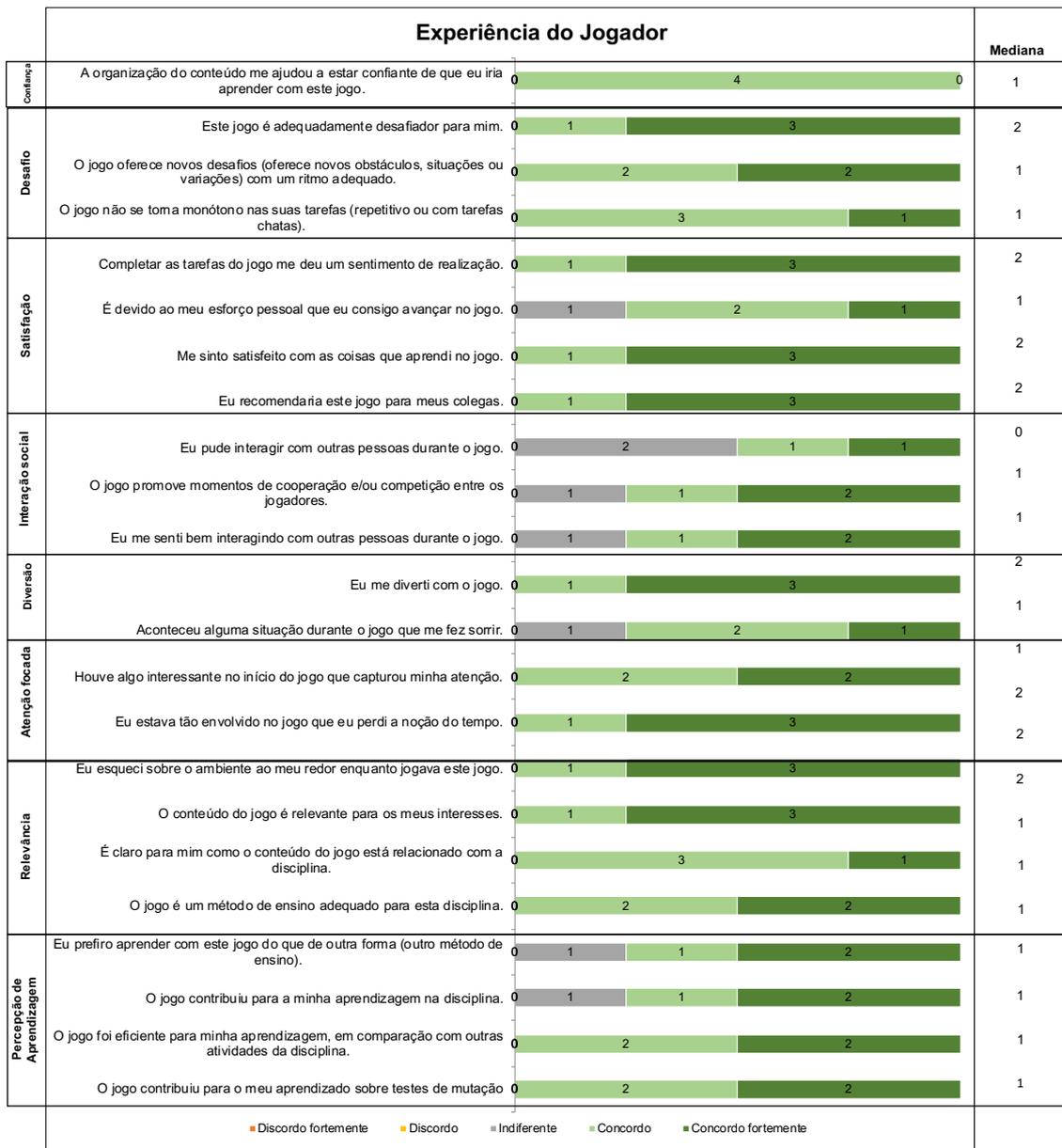


Figura 4.9: Gráfico de experiência dos jogadores do Grupo 2.

lado, dois jogadores concordaram e um jogador concordou totalmente com esta afirmativa, o que nos leva a crer que o conhecimento de *JavaScript* no mínimo facilita o uso do jogo, podendo a falta deste conhecimento atuar como barreira para os jogadores. Metade dos jogadores concordaram totalmente que não tiveram dificuldades em aprender o jogo, havendo concordância parcial para os demais.

Na subdimensão **operabilidade**, todos os jogadores concordaram que o jogo é fácil de jogar e 75% concordaram (parcial ou totalmente) que as regras são claras e compreensíveis. Um jogador demonstrou indiferença quanto à clareza das regras.

Quanto a **acessibilidade**, 75% concordaram totalmente que as fontes utilizadas no jogo são legíveis e as cores utilizadas são compreensíveis.

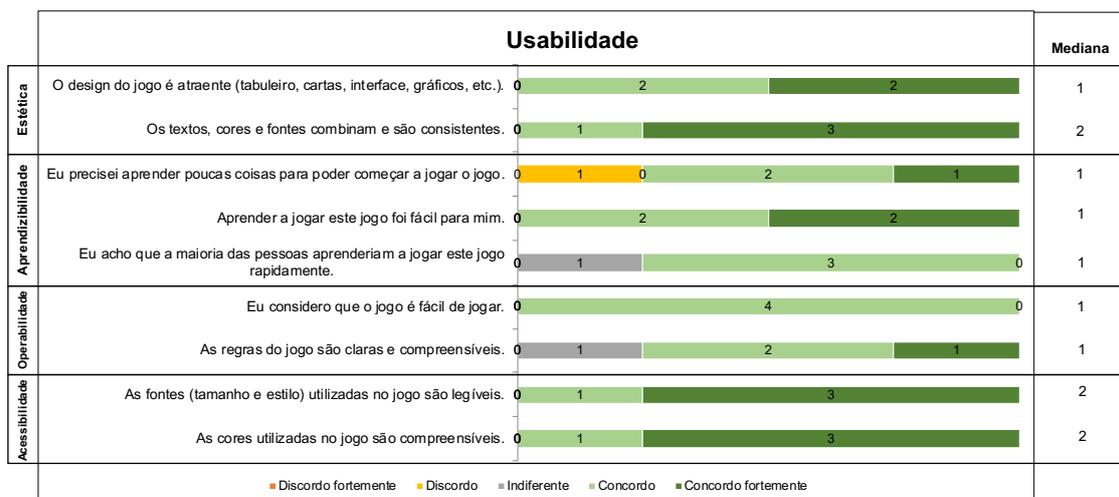


Figura 4.10: Gráfico de usabilidade dos jogadores do Grupo 2.

Para análise das questões abertas, na pergunta "O que você mais gostou do jogo?" os jogadores destacaram os elementos visuais e o ambiente competitivo como bons elementos ao aprendizado.

“O desafio de alcançar o objetivo e a competitividade com os demais participantes se destacam no jogo”

– Participante 1

“Plataforma muito fácil de usar.”

– Participante 2

“Achei muito boa a abordagem do ranking no aprendizado.”

– Participante 3

“A interface (visual) é muito boa.”

– Participante 4

Como pontos de melhoria, os jogadores do Grupo 2 trouxeram itens relevantes de serem implementados na plataforma:

“A funcionalidade de criar uma arena e mandá-la como desafio para outro participante.”

– Participante 1

“Regras mais claras: o número de envios diminui a pontuação do jogador.”

– Participante 2

“Não saber quando eu programava errado.”

– Participante 3

“Erros de sintaxe poderiam ser avisados.”

– Participante 4

Foi interessante perceber que a maioria dos pontos a melhorar estão relacionadas a novas funcionalidades desejadas para a plataforma. Cada participante sugeriu novas implementações em diferentes componentes do jogo. O Participante 1 mencionou uma nova funcionalidade que se distancia da proposta inicial sobre o uso do jogo em sala de aula ao qual o propósito de todos participarem da mesma arena tem como objetivo proporcionar um ambiente competitivo. O Participante 2 sugeriu informar aos jogadores qual é a regra de funcionamento do *ranking* na tela de arena e, assim, tornar claro o seu mecanismo. Os Participantes 3 e 4 mencionaram a falta de *feedback* da plataforma quando o jogador envia um código com erros de sintaxe. Uma aba contendo os erros identificados pelo servidor da plataforma seria uma forma de prover esse *feedback* aos jogadores.

4.4 Discussão dos Resultados

Por meio da análise dos resultados, foi possível perceber que as dimensões “usabilidade” e “experiência do jogador” foram bem avaliadas por ambos os grupos, isto é, os mecanismos e *design* do jogo proporcionaram uma experiência de aprendizado em um formato de texto, cores e fontes atraentes. A Figura 4.11 apresenta a dimensão de experiência do jogador e a Figura 4.12 apresenta a dimensão usabilidade, ambas representando uma visão consolidada para os participantes dos dois grupos.

Quanto a dimensão “experiência do jogador”, de maneira geral os jogadores concordaram que o mecanismo de jogos proporcionou uma experiência agradável de aprendizado. Mesmo em um ambiente virtual, diante da situação de distanciamento social provocada pela COVID-19 [28], o jogo foi capaz de proporcionar o aprendizado sobre testes de mutação, capturando a atenção dos jogadores, promovendo cooperação e competição entre eles e refletindo a relevância da atividade de testes de software.

Os jogadores relataram que o Mutant Arena é um jogo que traz um sentimento de realização ao completar os desafios de cada arena e isso se torna possível devido ao es-

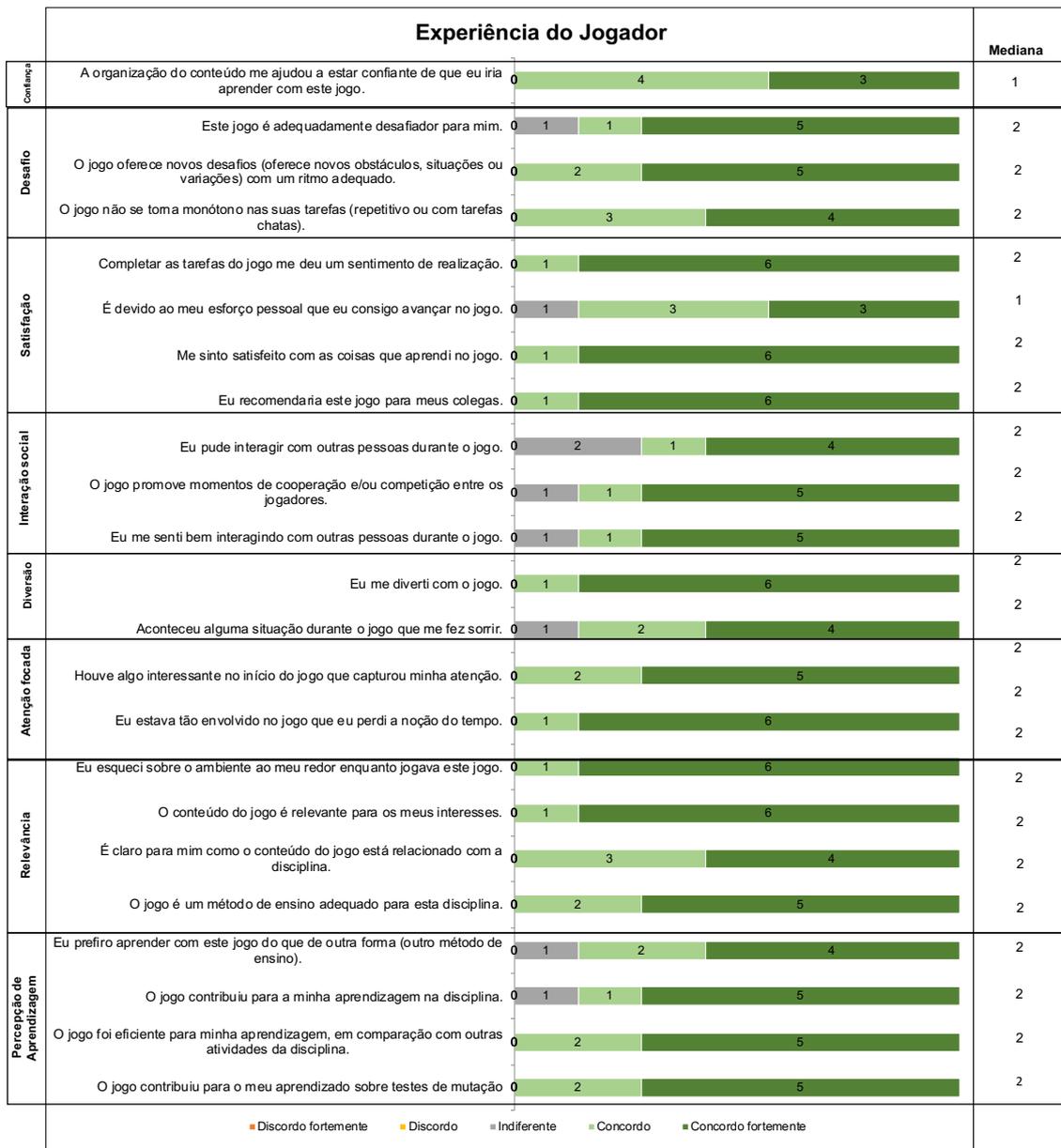


Figura 4.11: Gráfico de usabilidade dos jogadores de todos os grupos.

forço do jogador. A maioria dos jogadores relatou confiar no objetivo de aprendizado da plataforma, confiança que é traduzida na recomendação de uso do jogo para seus colegas. Como nos dois grupos tivemos jogadores com atuação profissional na indústria de software, podemos concluir que o jogo também poderia ser utilizado no meio corporativo como uma ferramenta de aprendizado.

A proposta do jogo é proporcionar o aprendizado de maneira divertida. Embora os dois grupos tenham concordado que se divertiram, o primeiro grupo avaliou este quesito de forma mais enfática, mesmo o grupo tendo na sua formação um jogador que raramente joga jogos digitais. Durante a execução, os jogadores perderam a noção do tempo e pediram para que novas arenas fossem cadastradas para continuar o jogo.

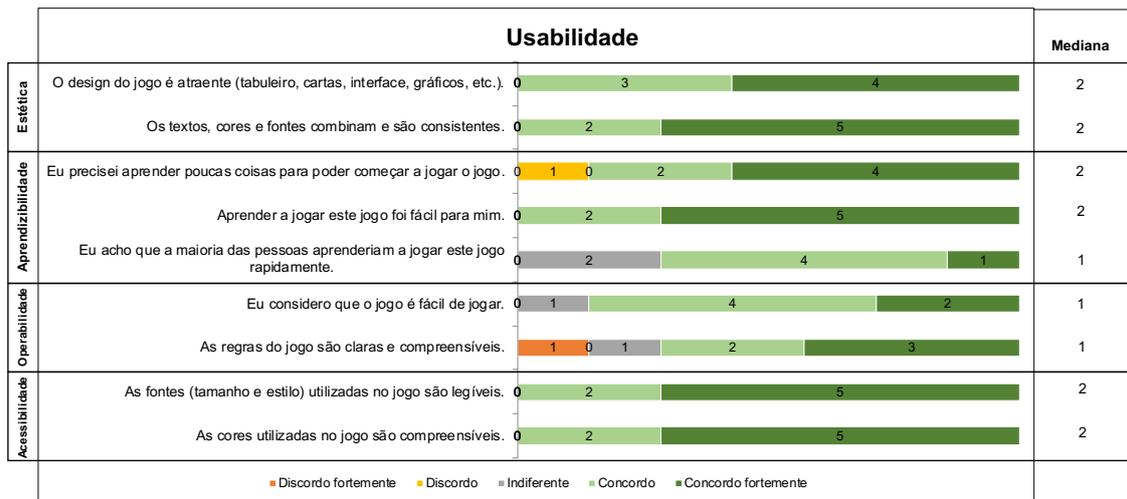


Figura 4.12: Gráfico de experiência dos jogadores de todos os grupos.

Os jogadores que nunca tinham utilizado testes de mutação terminaram o experimento destacando que o uso deste tipo de teste traria benefícios aos produtos de software que desenvolvem em suas pesquisas ou companhias. Isso traz indícios de que o jogo se mostra eficiente quanto aos objetivos de aprendizagem, mesmo para jogadores que não estejam habituados a conceitos mais avançados de testes.

Quanto a dimensão **usabilidade**, a maioria dos jogadores relatou precisar aprender poucas coisas para começar a jogar. Isso demonstra que a plataforma poderia ser utilizada em uma sala de aula com facilidade, no intuito de apresentar o tema de testes de mutação a alunos com pouca experiência em programação.

Um dos jogadores relatou que as regras do jogo não são claras e compreensíveis. De acordo com uma das questões abertas de pontos a melhorar relatadas na Seção 4.2 do Grupo 2, foi possível entender que esta resposta foi provocada pela ausência de regras sobre o funcionamento do *ranking*. Este elemento de jogo é muito utilizado em jogos digitais *multiplayer* [29, 30] e consideramos que não seria necessário explicar o seu funcionamento. Portanto, também podemos entender que a motivação para esta discordância se deve a que alguns jogadores raramente jogam jogos digitais.

4.5 Considerações Finais

Este capítulo teve por objetivo apresentar a etapa de avaliação da plataforma Mutant Arena, composta por planejamento, execução, análise e discussão dos resultados. Por meio da análise, foi possível perceber que, na percepção dos jogadores, o jogo é eficiente quanto aos seus objetivos de aprendizagem.

Por meio da avaliação, também foi possível perceber que os elementos de jogo da plataforma proporcionaram um ambiente dinâmico, competitivo e alinhado aos desafios em um ambiente sem monotonia no jogar. O jogo também se mostrou eficiente com jogadores que não estão habituados com o aprendizado através de uma plataforma gamificada.

Os pontos de melhoria sugeridos pelos participantes serão levados em consideração para a posterior aprimoramento do jogo. Os principais pontos são: informações sobre a regra de funcionamento do *ranking* na tela de jogo, *feedback* na tela para informar quando o jogador envia um código com erros de sintaxe, possibilidade de maximização da área de inserção do código-fonte (tanto no modo jogador quanto no modo instrutor), tutorial de uso da plataforma no modo jogador e uma tela com exemplos de casos de teste.

5. Conclusões

O uso de testes no ciclo de desenvolvimento de software têm contribuído fortemente para o aumento da qualidade no software. Porém, criar casos de teste não é uma tarefa fácil. Novas tendências tem surgido na literatura acadêmica onde o uso de elementos de jogos tem tornado a tarefa da elaboração de testes mais envolvente [4, 17, 26, 27].

O constante surgimento de novos conceitos e os avanços no processo de desenvolvimento de software têm tornado o ensino da disciplina de testes um desafio. Dessa forma, a área de Engenharia de Software tem buscado alternativas para o ensino em suas subáreas, e entre essas alternativas, cita-se o uso de jogos educacionais.

Sendo assim, esta dissertação desenvolveu um jogo para auxiliar o entendimento de conceitos de testes, bem como auxiliar os jogadores a fixá-los.

O Mutant Arena tem como objetivo ensinar conceitos relacionados a testes para alunos e desenvolvedores de software. A plataforma apresenta uma maneira informal e divertida no processo de aprendizado com o uso de elementos de jogos. A ideia geral é consolidar os conceitos sobre testes e reforçar o aprendizado, fornecendo uma ferramenta de apoio ao ensino.

Para entender a percepção dos alunos ao jogar o jogo, realizamos uma etapa de avaliação do jogo utilizando o Método de Avaliação de Jogos - MEEGA+. A avaliação ocorreu em dois grupos diferentes e, por meio da análise dos resultados, foi possível observar que, na percepção dos jogadores, o jogo é eficaz quanto aos seus objetivos de aprendizagem. Mesmo executando o jogo utilizando video-chamada devido a situação do mundo em relação à pandemia pelo vírus SARS-COV-2 [28], foi possível perceber que os jogadores tiveram uma boa interação no jogo em um ambiente competitivo.

O fato do jogo apresentar mecânicas simples e objetivos claros fez com que a maioria dos jogadores percebessem o propósito de aprendizado. De maneira geral, tanto a

experiência do jogador como a usabilidade foram dimensões muito bem avaliadas pelos jogadores. Assim, podemos concluir que os elementos de jogos e as mecânicas, proporcionaram um ambiente eficaz para o aprendizado, e a usabilidade mostrou que o design do jogo é atraente, as regras são claras e fácil de se aprender. Além disso, os alunos que não costumam jogar, acharam que o jogo é um método de ensino adequado e divertido.

5.1 Contribuição

Como principal contribuição deste trabalho, apresentamos um jogo (Mutant Arena) que tem como o objetivo ensinar conceitos relacionados ao desenvolvimento de testes para alunos e profissionais da indústria de software, bem como sua avaliação utilizando a Metodologia de Avaliação de Jogos Educacionais - MEEGA+.

O código-fonte do *Mutant Arena* está disponível em <https://github.com/gohackfelipe/mutantarena> de maneira *OpenSource*, assim toda comunidade acadêmica ou indústria de software pode utilizá-lo.

5.2 Limitações

Este trabalho apresenta limitações quanto a avaliação do jogo, o grupo participante consistiu em apenas sete (7) jogadores, um número muito pequeno para uma validação consistente.

Devido a pandemia global provocada pelo vírus SARS-COV-2 [28], foi necessário seguir as recomendações do Ministério da Saúde do Brasil quanto ao distanciamento social, e isso evitou que a execução do experimento fosse realizada em um laboratório local com um maior número de jogadores.

Mesmo os jogadores relatando que a plataforma proporcionou uma boa experiência ao usuário com uma ótima usabilidade, o fato de utilizarmos uma ferramentas de videochamada pode ter afetado essas dimensões.

5.3 Trabalhos Futuros

Como trabalhos futuros, pretende-se evoluir o jogo com base nos resultados obtidos na avaliação apresentada no Capítulo 4. Além disso, pretende-se desenvolver as novas

mecânicas sugeridas pelos jogadores (relatadas na Seção 4.3.1 e 4.3.2) na avaliação do jogo a fim de reforçar ainda mais o aprendizado no jogo:

- A ausência na plataforma sobre as regras do jogo será resolvida com a implementação de uma seção ajuda onde o jogador poderá compreender o funcionamento das principais mecânicas do jogo; ranking, placar e relatório de análise dos mutantes, bem como exemplos de código que auxiliem o jogador a dar os seus primeiros passos na criação dos casos de testes.
- A implementação de um mecanismo de *log* com o propósito de informar ao jogador quando existe um erro no código-fonte detectado pelo servidor que executa os testes de mutação.
- Possibilidade de maximização do painel de inserção do código-fonte. Este painel está disposto na tela de arena no perfil jogador e configuração da arena no perfil instrutor.
- Desenvolvimento de um mecanismo de *feedback* em forma de *tutorial* para auxiliar no primeiro contato do jogador com a plataforma.

Esta nova versão do jogo também deverá ser avaliada e pretende-se avaliar o uso do jogo com mais jogadores, além da utilização de outras formas de avaliação (não apenas o MEEGA+) para reforçar a contribuição do jogo para o processo de ensino-aprendizagem de testes na Engenharia de Software.

Referências Bibliográficas

- [1] PETRI, G. et al. MEEGA+: Um Modelo para a Avaliação de Jogos Educacionais para o ensino de Computação. *Revista Brasileira de Informática na Educação*, v. 27, n. 03, p. 52, 2020. ISSN 1414-5685. Disponível em: <<https://www.br-ie.org/pub/index.php/rbie/article/view/8421>>.
- [2] RICE, J. The Gamification of Learning and Instruction: Game-Based Methods and Strategies for Training and Education. *International Journal of Gaming and Computer-Mediated Simulations*, 2012. ISSN 1942-3888.
- [3] ALHAMMAD, M. M.; MORENO, A. M. Gamification in software engineering education: A systematic mapping. *Journal of Systems and Software*, 2018. ISSN 01641212.
- [4] FRASER, G. et al. Gamifying a software testing course with code defenders. In: *SIGCSE 2019 - Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. [s.n.], 2019. p. 571–577. ISBN 9781450358903. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85064411003doi=10.1145>>
- [5] DUBOIS, D. J.; TAMBURRELLI, G. Understanding gamification mechanisms for software development. In: *2013 9th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, ESEC/FSE 2013 - Proceedings*. [S.l.: s.n.], 2013. ISBN 9781450322379.
- [6] GROUP, S. C.; HP. *World Quality Report 2015-16. Tech. Report. Cap Gemini S.A., Sogeti and Hewlett-Packard*. [S.l.], 2016. Disponível em: <<https://www.capgemini.com/thought-leadership/world-quality-report-2015-16>>.
- [7] JOHNSON, L. et al. *Horizon Report - 2016 Higher Education Edition*. [S.l.: s.n.], 2016. ISBN 9780989733557.

- [8] FARD, A. M.; MESBAH, A. JavaScript: The (Un)Covered Parts. In: *Proceedings - 10th IEEE International Conference on Software Testing, Verification and Validation, ICST 2017*. [S.l.: s.n.], 2017.
- [9] PEDREIRA, O. et al. Gamification in software engineering - A systematic mapping. In: *Information and Software Technology*. [S.l.: s.n.], 2015. ISSN 09505849.
- [10] DETERDING, S. et al. From game design elements to gamefulness: Defining "gamification". In: *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments, MindTrek 2011*. [S.l.: s.n.], 2011. ISBN 9781450308168.
- [11] GARCÍA, F. et al. A framework for gamification in software engineering. *Journal of Systems and Software*, v. 132, 2017. ISSN 01641212.
- [12] PASSOS, E. B. et al. Turning real-world software development into a game. In: *Brazilian Symposium on Games and Digital Entertainment, SBGAMES*. [S.l.: s.n.], 2011. ISSN 21596654.
- [13] NAH, F. F. H. et al. Gamification of education: A review of literature. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. [S.l.: s.n.], 2014. ISBN 9783319072920. ISSN 16113349.
- [14] WERBACH, K.; HUNTER, D. The Gamification Toolkit: Game Elements. *For the Win: How Game Thinking Can Revolutionize Your Business*, 2012.
- [15] ROJAS, J. M. et al. Code Defenders: Crowdsourcing Effective Tests and Subtle Mutants with a Mutation Testing Game. In: *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*. [S.l.: s.n.], 2017. p. 677–688.
- [16] AMMANN, P.; OFFUTT, J. *Introduction to Software Testing*. [S.l.: s.n.], 2008.
- [17] ROJAS, J. M.; FRASER, G. Code Defenders: A Mutation Testing Game. In: *Proceedings - 2016 IEEE International Conference on Software Testing, Verification and Validation Workshops, ICSTW 2016*. [S.l.: s.n.], 2016. ISBN 9781509018260.
- [18] CLEGG, B. S.; ROJAS, J. M.; FRASER, G. Teaching Software Testing Concepts Using a Mutation Testing Game. In: *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET)*. [S.l.: s.n.], 2017. p. 33–36.

- [19] ANDERSON, P.; NASH, T.; MCCAULEY, R. Facilitating programming success in data science courses through gamified scaffolding and Learn2Mine. In: *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE*. [S.l.: s.n.], 2015. ISBN 9781450334402. ISSN 1942647X.
- [20] SANTOS, H. M. D. et al. Cleangame: Gamifying the identification of code smells. In: *ACM International Conference Proceeding Series*. [S.l.: s.n.], 2019.
- [21] MOREIRA, G. G.; MARQUES, A. B. dos S. Evaluating the students' experience with the Scrum Card Game: An experience report in a Software Engineering course. In: *ACM International Conference Proceeding Series*. [S.l.: s.n.], 2018.
- [22] BEPPE, T. A. et al. GreaTest: A card game to motivate the software testing learning. In: *ACM International Conference Proceeding Series*. [S.l.: s.n.], 2018.
- [23] MOURA, V.; SANTOS, G. Procsoft: A board game to teach software processes based on ISO/IEC 29110 standard. In: *ACM International Conference Proceeding Series*. [S.l.: s.n.], 2018.
- [24] PETRI, G.; WANGENHEIM, C. G. V.; BORGATTO, A. F. A large-scale evaluation of a model for the evaluation of games for teaching software engineering. In: *Proceedings - 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering and Education Track, ICSE-SEET 2017*. [S.l.: s.n.], 2017.
- [25] SAVI, R. et al. Proposta de um Modelo de Avaliação de Jogos Educacionais. *RE-NOTE*, v. 8, n. 3, 2010. ISSN 1679-1916.
- [26] FRASER, G.; GAMBI, A.; ROJAS, J. M. A preliminary report on gamifying a software testing course with the code defenders testing game. In: *ACM International Conference Proceeding Series*. [s.n.], 2018. p. 50–54. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85049865039doi=10.1145>
- [27] ROJAS, J. M.; FRASER, G. Teaching Mutation Testing using Gamification. *European Conference of Software Engineering Education 2016 (ECSEE)*, 2016.
- [28] PASCARELLA, G. et al. *COVID-19 diagnosis and management: a comprehensive review*. 2020.
- [29] DICHEVA, D. et al. Gamification in education: A systematic mapping study. *Educational Technology and Society*, v. 18, n. 3, p. 75–88, 2015. ISSN 14364522.
- [30] RODRIGUES, L. F.; OLIVEIRA, A.; RODRIGUES, H. Main gamification concepts: A systematic mapping study. *Heliyon*, v. 5, n. 7, 2019. ISSN 24058440.

A. Formulário de Avaliação de Jogo

As páginas a seguir contém o formulário preenchido pelos participantes da avaliação experimental da plataforma Mutant Arena.

O link para acesso ao formulário está disponível em <https://forms.gle/RShNsLpcSCGHSURy9> e as respostas do formulário se encontram no arquivo *respostas.pdf* no repositório do projeto <https://github.com/gohackfelipe/mutantarena>.

Mutant Arena: Um software gamificando o ensino de teste

Este questionário está sendo conduzido pelo aluno de mestrado Felipe Ribeiro (UNIRIO).

Este survey tem como objetivo entender a qualidade do uso do jogo Mutant Arena, que busca apresentar conceitos sobre testes de maneira gamificada.

***Obrigatório**

Termo de Consentimento Livre e Esclarecido

Ao responder a esta pesquisa, você concorda em permitir que o pesquisador obtenha, use e divulgue as informações anônimas fornecidas conforme descrito abaixo.

CONDIÇÕES E ESTIPULAÇÕES

1. Eu entendo que todas as informações são confidenciais. Eu não serei pessoalmente identificado e concordo em concluir o survey para fins de pesquisa e que os dados derivados dessa pesquisa anônima podem ser publicados em periódicos, conferências e publicações em blogs.
2. Entendo que minha participação neste estudo de pesquisa é totalmente voluntária e que recusar participar não envolverá penalidade ou perda de benefícios. Se eu escolher, posso retirar minha participação a qualquer momento. Eu também entendo que, se eu optar por participar, posso me recusar a responder qualquer pergunta que eu não esteja confortável em responder.
3. Entendo que posso entrar em contato com o pesquisador se tiver alguma dúvida sobre a pesquisa. Estou ciente de que meu consentimento não me beneficiará diretamente. Também estou ciente de que o autor manterá os dados coletados em perpetuidade e poderá utilizar dados para trabalhos acadêmicos futuros.
4. Ao clicar no botão abaixo, eu livremente forneço consentimento e reconheço meus direitos como participante voluntário da pesquisa, conforme descrito acima, e forneço consentimento ao pesquisador para usar minhas informações na condução de pesquisas sobre as áreas mencionadas acima.

RESPONSÁVEIS

Pesquisador:
Felipe Ribeiro Santos <felipe.santos@uniriotec.br>
PPGI - Programa de Pós-Graduação em Informática

Professor:
Prof. Márcio de Oliveira Barros <marcio.barros@uniriotec.br>
Departamento de Informática Aplicada - UNIRIO

1. Eu concordo em participar da avaliação conduzida por Felipe Ribeiro Santos sob orientação do Prof. Márcio de Oliveira Barros

Marcar apenas uma oval.

Sim

Não *Pular para a seção 5 (Obrigado)*

Informações
Demográficas

Por favor, solicitamos que você responda as perguntas abaixo com as informações demográficas. Todos os dados são coletados anonimamente e somente serão utilizados no contexto desta pesquisa.

2. Instituição / Empresa *

3. Curso / Área de atuação *

4. Você possui conhecimento prévio sobre testes de mutação? *

Marcar apenas uma oval.

Sim

Não

5. Faixa Etária

Marcar apenas uma oval.

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

6. Sexo

Marcar apenas uma oval.

- Masculino
- Feminino
- Outros

7. Com que frequência você costuma jogar jogos digitais?

Marcar apenas uma oval por linha.

	Nunca: nunca jogo	Raramente: jogo de tempos em tempos	Mensalmente: jogo pelo menos uma vez por mês	Semanalmente: jogo pelo menos uma vez por semana	Diariamente: jogo todos os dias
Com que frequência você costuma jogar jogos digitais?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Com que frequência você costuma jogar jogos não-digitais (de carta, tabuleiros, etc)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Link e Descrição do Jogo

O Mutant Arena é um jogo educativo com o objetivo de consolidar os conhecimentos sobre testes de mutação.

O link para acesso ao jogo é <https://mutantarena.feliperibeiro.dev/>

Ao abrir o link, utilize o navegador Google Chrome. O jogo será carregado em seu navegador e esse processo poderá demorar um pouco.

Após a utilização da plataforma, você deverá continuar respondendo o questionário.

Basta clicar em próximo para responder as próximas perguntas.

Questionário para avaliação da Qualidade do Jogo.

Nesta etapa, gostaríamos que você avaliasse o jogo respondendo ao questionário.

8. Usabilidade

Marque uma opção conforme sua avaliação

Marcar apenas uma oval por linha.

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

9. Usabilidade

Marque uma opção conforme sua avaliação

Marcar apenas uma oval por linha.

	Discordo Totalmente	Discordo	Nem Discordo, nem concordo	Concordo	Concordo Totalmente
A organização do conteúdo me ajudou a estar confiante de que eu iria aprender com este jogo.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Este jogo é adequadamente desafiador para mim.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O jogo oferece novos desafios (oferece novos obstáculos, situações ou variações) com um ritmo adequado.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O jogo não se torna monótono nas suas tarefas (repetitivo ou com tarefas chatas).	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Completar as tarefas do jogo me deu um sentimento de realização.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
É devido ao meu esforço pessoal que eu consigo avançar no jogo.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Me sinto satisfeito com as coisas que aprendi no jogo.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Eu recomendaria este jogo para meus	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

colegas.

Eu pude interagir com outras pessoas durante o jogo.

O jogo promove momentos de cooperação e/ou competição entre os jogadores.

Eu me senti bem interagindo com outras pessoas durante o jogo.

Eu me diverti com o jogo.

Aconteceu alguma situação durante o jogo (elementos do jogo, competição, etc.) que me fez sorrir

Houve algo interessante no início do jogo que capturou minha atenção.

Eu estava tão envolvido no jogo que eu perdi a noção do tempo.

Eu esqueci sobre o ambiente ao meu redor enquanto jogava este jogo.

O conteúdo do jogo é relevante para os meus interesses.

É claro para mim como o conteúdo do jogo está relacionado com a disciplina.

O jogo é um método de ensino adequado para esta disciplina.

Eu prefiro aprender com este jogo do que de outra forma (outro método de ensino).

O jogo contribuiu para a minha aprendizagem na disciplina.

O jogo foi eficiente para minha aprendizagem, em comparação com outras atividades da disciplina.

10. O que você mais gostou no jogo? *

11. O que poderia ser melhorado no jogo? *

12. Gostaria de fazer mais algum comentário? *

Obrigado

Muito obrigado pela sua contribuição!

Este conteúdo não foi criado nem aprovado pelo Google.

Google Formulários