

Uma metodologia remota gamificada para o ensino de testes unitários

Renata Faria Gomes
Universidade Federal do Ceará
Fortaleza, Brasil
renata.faria.gomes@hotmail.com.br

Valéria Lelli
Universidade Federal do Ceará
Fortaleza, Brasil
valerialelli@ufc.br

Resumo

Testes de software são fundamentais para assegurar a qualidade de um software. No contexto de teste de Software, existem os testes unitários. Apesar da importância, existe uma carência de profissionais qualificados, que pode estar intimamente ligada à falta de ensino prático. Para isso, o presente trabalho propõe a utilização de uma metodologia gamificada, que é constituída por três etapas interligadas com uma forte narrativa. A primeira etapa é um jogo que introduz de maneira sutil conceitos de testes unitários, de duplês de teste e a estrutura given-when-then. A segunda é uma videoaula que utiliza um código similar ao do jogo para explicar e exemplificar os conceitos abordados anteriormente. A terceira etapa é composta por um exercício de fixação e um desafio. A metodologia foi aplicada em uma disciplina de Verificação, Validação e Testes da Universidade Federal do Ceará. Ela foi avaliada através de dois questionários e um fórum com perguntas subjetivas. Os resultados obtidos foram positivos, a metodologia utilizada foi avaliada com notas acima de 8 em uma escala de 1 a 10. Os alunos gostaram da narrativa lúdica e os resultados indicaram que os alunos ficaram satisfeitos com o aprendizado, uma vez que observou-se que todos o grupos acertaram pelo menos 60% do exercício.

Keywords

Teste de Software; Ensino remoto; Gamificação.

1 Introdução

Testar um software é o ato de executar uma aplicação, simulando comportamentos de entrada e analisando se os resultados obtidos estão conforme o esperado [29]. No contexto de teste de Software, existem os testes unitários, que são aqueles que levam em consideração a lógica interna de uma unidade de um programa de computador [1].

Nessa direção, essa é uma atividade extremamente importante para o mercado de tecnologia, uma vez que garante o correto funcionamento de aplicações que estão presentes no cotidiano das pessoas e que poderiam causar grandes prejuízos caso apresentassem erros [4]. No entanto, apesar da importância dos testes, há uma escassez do profissional com esse conhecimento na indústria. Segundo Valle [29], essa carência de profissionais capacitados está intimamente ligada ao método tradicional de ensino, que tem como principal metodologia a utilização de aulas teóricas e falha na criação de ambientes práticos que motivem os estudantes.

Como uma potencial solução para esse problema, Fraser [10] afirma que a gamificação oferece elementos que facilitam o ensino. Dentre outros fatores, a gamificação possui narrativa, que é uma característica muito importante para a educação, pois exemplifica conteúdos do mundo real por meio de metáforas que ajudam na compreensão [20].

Nesse contexto, Elbaum et al.[9], criou um jogo chamado *Bug Hunt*, uma das fases do jogo era relacionada a testes de caixa branca. No entanto, o jogo não possuía um elemento chave de jogos: uma narrativa fantástica [20]. Já em Lelli et al. [18], é utilizada uma metodologia gamificada que faz uso de narrativas por meio da plataforma *ClassCraft* para o ensino de Engenharia de Software, no entanto, os conteúdos relacionados à Teste de Software são mais introdutórios e os testes unitários são abordados de forma superficial.

Sendo assim, o presente trabalho tem como objetivo utilizar elementos de gamificação, principalmente uma narrativa, para tornar o ensino de testes unitários mais acessível para estudantes de graduação. A metodologia proposta consistiu de três etapas: o desenvolvimento de um jogo *web* baseado em uma dinâmica de tentativa e erro; a produção de uma videoaula que aborda conceitos teórico-práticos de testes unitários; e a elaboração de um exercício de fixação com desafio para os alunos praticarem a escrita de testes unitários.

Devido à pandemia do COVID-19, a metodologia foi aplicada de forma remota em uma disciplina de Verificação, Validação e Testes (V&V) da Universidade Federal do Ceará e foi avaliada por meio de dois questionários e um fórum com perguntas subjetivas. Os resultados obtidos mostraram que os alunos gostaram do Jogo e da narrativa lúdica proporcionada pelo metodologia. Além disso, 4 de 9 equipes concluíram o exercício de fixação com êxito. Porém, apenas 2 equipes conseguiram resolver corretamente o desafio e, do total de 8 alunos que responderam o questionário final do exercício, 2 alunos mencionaram que acharam o exercício difícil de resolver.

Este artigo está estruturado da seguinte maneira: na Seção 2 são apresentados os conceitos que são abordados por este trabalho. Na Seção 3 são discutidos os trabalhos relacionados a jogos que se apoiam no ensino de testes e gamificação. A metodologia, a concepção do Jogo, a produção da videoaula e a elaboração do exercício de fixação são descritos nas seções 4, 5, 6 e 7, respectivamente. Os resultados obtidos com a aplicação da metodologia são apresentados na Seção 8 e, por fim, na Seção 9 são apresentados as considerações finais e os futuros direcionamentos para este trabalho.

2 Fundamentação Teórica

Esse trabalho apoia-se sobre dois principais tópicos: o de gamificação no âmbito educacional e o de testes unitários. Ambos conceitos serão apresentados a seguir.

2.1 Elementos da gamificação

Gamificação é quando elementos de jogos, como objetivos, regras, desafios e interatividade, são incorporados em atividades diferentes daqueles dos jogos [15][8]. Essa técnica é bastante utilizada

como uma potencial solução para tornar o ensino mais divertido e engajador [10].

Nesse cenário, existem fatores que podem aumentar o engajamento de jogos educacionais, como elementos emocionais, um *design* que construa uma atmosfera convincente e principalmente uma narrativa [14] [10]. Os elementos emocionais fornecem algum sentimento ao jogador, podendo ser derrota, vitória, responsabilidade entre outros, dessa forma o jogador se engaja para conquistar o objetivo [14]. Além desse fator, a estética (cores, desenhos, *feedback*, entre outros) auxilia nos elementos emocionais e torna o jogador mais imerso dentro do jogo. Adicionar um elemento de desafio, com picos de tensão e que requerem bastante atenção, também podem auxiliar na imersão dos jogadores em jogos [3].

Por último, a narrativa, é definida como um conjunto de regras que são vivenciadas durante o jogo, de modo que o jogador interage com o universo do jogo [26]. A narrativa é um elemento sensorial passivo, que ajuda o jogador a entender os objetivos e tomar decisões corretas que influenciarão seu resultado positivo [24]. Além disso, a narrativa de um jogo é um elemento-chave, ainda mais no âmbito educacional, pois exemplifica conteúdos do mundo real por meio de metáforas que ajudam na compreensão [20].

2.2 Teste Unitário

Testes de software são uma parte importante da Engenharia de Software, já que detectam se o programa de computador executa conforme o esperado e de acordo com os requisitos [21]. Nesse contexto, existem diversos tipos diferentes de testes, dentre eles, o teste unitário. Segundo a IEEE [1], testes unitários referem-se ao teste que leva em consideração a lógica interna de uma unidade de software. Essa unidade pode ser definida como um componente de software que não pode ser subdividido em outros componentes [1].

O primeiro conceito utilizado neste trabalho e que tem relação à testes unitários é conceituado no trabalho de [16], no qual a escrita de testes pode ser facilitada utilizando uma estrutura de sentenças nomeadas de *given-when-then*, ou em português, dado-quando-então. Essa metodologia divide o teste em três seções:

- (1) *Dado* – Especifica o estado e pré-condições existentes antes do teste. Caso seja necessário a configuração prévia de algum objeto, esta é a seção adequada.
- (2) *Quando* – Está relacionada ao comportamento que deverá levar ao cenário posterior verificado pelo teste unitário.
- (3) *Então* – Descreve o resultado do comportamento, fazendo as afirmações e conferindo se o teste obteve o resultado esperado.

Por exemplo, ao testar uma função de cálculo de médias: **dado** um *array* com os valores 6 e 8, **quando** o método *calculoMedia* é chamado passando como parâmetro o *array* criado anteriormente, **então** o retorno esperado da função deve ser igual a 7.

A metodologia proposta neste trabalho também se baseia em conceitos de objetos simulados ou duplês de teste (*test double* em inglês), segundo [19], é uma maneira para auxiliar o desenvolvimento de testes complexos e que são difíceis de manter. Resumidamente, os objetos simulados são utilizados para prever domínios até então desconhecidos, simplificando e despoluindo a estrutura do teste [19]. Por exemplo, uma função que recupera dados de um banco de

dados pode ser testada utilizando uma função que mimetiza esse comportamento sem precisar consultar artefatos externos.

Apesar de visto em muitos trabalhos acadêmicos o uso do termo *mock*, este é adotado para resumir os tipos diferentes de duplês, como em [2] [5] [11], neste trabalho eles são apresentados individualmente como:

- **Dummy**. um objeto sem implementação, utilizado em métodos que necessitam de argumentos que não serão utilizados [22].
- **Spy**. Segundo [17], são objetos utilizados no lugar de dependências que observam os valores passados dentro de funções. Dessa forma, é possível comparar o que foi passado e o que se espera e ao verificar que os valores são iguais o teste é bem-sucedido. Em adição a isso, também é possível utilizar esses duplês para contabilizar a quantidade de vezes que uma função é chamada.
- **Stub**. [5] define *stubs* como sendo implementações falsas de código que retornam resultados esperados durante testes unitários.

Além disso, também foi utilizado o termo *SUT* (*system under test*), este trabalho usa a definição de [22]: no contexto de testes unitários para sistemas orientados a objetos, o *SUT* é qualquer classe que está sendo testada.

3 Trabalhos Relacionados

Elbaum et al.[9], desenvolveram um jogo chamado *Bug Hunt* em que os jogadores são desafiados a aplicar diferentes técnicas de teste para solucionar problemas. Criado para a plataforma *Web*, o jogo é dividido em 4 missões, sendo a terceira relacionada ao teste de caixa branca. Nessa missão, os jogadores conseguem visualizar o código-fonte e por meio dele inserir na plataforma os casos de teste. Quanto maior a cobertura de código, melhor será a pontuação do jogo. Apesar de alguns exercícios terem uma um pequeno contexto, o jogo em si não possui uma narrativa do início ao fim.

Beppe et al. [4], por sua vez, utiliza um jogo de cartas para explicar conceitos de teste de software, tais como teste de aceitação, teste de desempenho entre outros. A jogabilidade é composta por cartas com diferentes funções os jogadores podem executar ações ao utilizar cartas que estejam em sua mão, como carta testador, cartas testes entre outras. Apesar de algumas cartas possuírem algumas narrativas curtas, o jogo não possui uma narrativa constante durante toda sua duração. Além disso, as atividades são desenvolvidas para aulas presenciais, mas a jogabilidade demonstrou ser bastante interessante.

Lelli et al. [18] aborda uma metodologia de ensino gamificada utilizando missões por meio da plataforma digital *Classcraft*. A metodologia aborda narrativas em duas diferentes situações: ao inserir o aluno como um herói que precisa completar missões e obter conquistas e em alguns jogos que foram adaptados, onde existem cenários fictícios, como no jogo de gerência de riscos e no de gerência de projetos. Esse trabalho utiliza de maneira eficaz as narrativas, porém aborda os conteúdos relacionados a Teste de Software para uma disciplina de Engenharia de Software com foco em conceitos introdutórios de teste de software, e, portanto, não abordando a escrita de testes unitários.

Valle et al. [13] desenvolveram um jogo com conceitos de teste funcional, teste estrutural e baseado em defeitos. Apesar de possuir teste estrutural, não foram abordados testes unitários. O jogo foi feito para *Web* e existe um personagem que anda, pula e tenta derrotar inimigos por meio de desafios baseados em testes. No entanto, apesar de conter elementos que podem formar uma narrativa, ela não está presente durante o jogo. Porém, o fato do jogador possuir um personagem é visto como um elemento positivo para a imersão do jogo.

Tabela 1: Quadro comparativo dos trabalhos relacionados

| Trabalho | Elementos de Gamificação | | Testes unitários | Atividades Remotas |
|---------------|--------------------------|-----------|------------------|--------------------|
| | Jogo | Narrativa | | |
| [9] | x | | x | x |
| [4] | x | | | |
| [18] | x | x | | x |
| [13] | x | | | x |
| Este Trabalho | x | x | x | x |

O diferencial deste trabalho está no uso de elementos de gamificação associado a etapas teórico-práticas para o ensino de testes unitários. Nos trabalhos relacionados, o que mais se assemelha é o de Lelli et al. [18], pois apesar de não possuir uma missão dedicada à testes unitários, possui uma metodologia que envolve conteúdos teórico-práticos utilizando uma narrativa como guia. Além desse, o trabalho de Elbaum et al. [9] é bastante interessante, pois aborda de maneira gamificada testes unitários, porém não utiliza narrativas para guiar os jogadores. Por último, [4] e [13] oferecem uma jogabilidade interessante, com elementos mais desafiadores que os demais trabalhos, no entanto, não abordam testes unitários e não possuem narrativas para guiar o jogo.

4 Metodologia

A metodologia proposta se divide em três etapas que se interligam entre si por meio de uma narrativa única. Conforme a Figura 1, a metodologia de ensino proposta consistiu de três etapas: (i) Jogo; (ii) videoaula; e (iii) exercício.



Figura 1: Metodologia do trabalho

A primeira etapa consiste de um jogo introdutório, com o objetivo de introduzir a narrativa e auxiliar na introdução dos conceitos

de testes unitários que seriam posteriormente explicados durante a videoaula. A concepção do jogo foi conduzida em quatro passos principais, no primeiro foi selecionado um jogo para basear a jogabilidade, em seguida, a narrativa que guiou o jogo foi definida.

A jogabilidade do jogo foi incorporada através de um protótipo de baixa fidelidade (prototipagem em papel) e, então, após um refinamento foi desenvolvida uma versão do jogo para a plataforma *Web*.

A segunda etapa é uma videoaula, que utiliza os exemplos do jogo para exemplificar o uso de testes. Além disso, a videoaula utiliza um código exemplo, que foi desenvolvido de forma a simular o código do jogo, de maneira a evitar dúvidas sobre a lógica do código. Por exemplo, não era preciso entrar em detalhes em como funcionava a função de *atacar*, pois os alunos já sabiam que a lógica desse método deveria conferir se as armas estavam corretas e nas posições certas, pois tinham jogado o jogo que possuía essa funcionalidade.

Para o desenvolvimento da videoaula foi necessário pensar em uma forma de conectá-la com o jogo, para isso, foi decidido a utilização de um código similar ao do jogo. Por isso, o segundo passo para o desenvolvimento da videoaula foi a confecção do código a ser utilizado. Após a codificação do exemplo, a videoaula foi gravada e disponibilizada na plataforma digital Youtube¹.

Com base no jogo desenvolvido na etapa anterior, os alunos eram incentivados a realizar um exercício de fixação composto de duas atividades e um desafio. O exercício foi elaborado para ser feita em equipe, de modo a permitir a colaboração entre alunos, também foi adicionado pontuação extra para os primeiros grupos que resolvessem a questão desafio, adicionando um pouco de competitividade ao exercício, os alunos tiveram o prazo de uma semana para executar e entregar as atividades.

5 Concepção do jogo

A concepção do jogo foi feita em quatro passos principais: escolha de um jogo base a ser adaptado; definição da narrativa; prototipagem em papel; e o desenvolvimento do jogo para a plataforma Web. Essas etapas são apresentadas nas próximas subseções.

5.1 Escolha do jogo base

O jogo foi baseado em um jogo de tabuleiro nomeado *Mastermind* [28], que explora uma dinâmica de tentativa e erro: o jogador precisa acertar uma senha gerada aleatoriamente, para isso ele testa diferentes combinações até acertar a combinação da resposta. Essa dinâmica é parecida com um teste unitário: dado uma resposta, ao questionar para o jogador que sabe a senha, será dado o veredito.

No entanto, o jogo original não possuía uma narrativa ou pelo menos elementos fantasiosos, porém, esse é um dos elementos-chave de jogos, atuando intrinsecamente e fornecendo metáforas e analogias que serão utilizadas posteriormente [20]. Além disso, a narrativa é mais que um elemento que guia um pensamento, é uma maneira de orientar ações dos estudantes [24]. Por essa razão, foi adicionado ao jogo – e à metodologia de ensino – uma narrativa.

5.2 Definição da narrativa

Segundo [8], a narrativa para concepção de projetos educacionais gamificados devem possuir a existência de um ator; elementos de

¹youtube.com

escolha; interatividade e sequência de eventos; espaço; e tempo de interação.

Por essa razão, foi primeiro definido o espaço onde o jogo estaria contido, sendo definida que a temática em que a história estava inserida seria da Terra Média, um universo baseado na mitologia nórdica dentro da janela temporal da idade média [6].

Com o espaço temporal definido, foi definido que o ator (o papel que o aluno atuaria ao jogar) seria um guerreiro que está iniciando seu treinamento para derrotar um monstro e para isso precisa escolher qual arma deve utilizar, sendo essa mecânica o elemento de escolha da narrativa, proporcionando também interatividade.

Cada ataque é uma sequência de eventos, cada vez que o jogador ataca mais perto ele estará perto da vitória ou da derrota. Em adição a isso o jogador também pode fazer testes de um ataque, com a justificativa que o jogador ainda está aprendendo a formular um ataque. Para isso, também foi adicionado um boneco de treino, direcionando o ataque do monstro para este personagem. Dessa forma, o jogador entenderia os possíveis motivos para seu ataque não ter sido eficaz, podendo alterar as armas escolhidas e tentar novamente.

Por fim, o tempo de interação foi definido por limitação da quantidade máxima de ataques e testes, que serão aprofundadas na seção 5.3.

Com a narrativa definida, foi possível o desenvolvimento da jogabilidade. Para isso, foi utilizado um protótipo em papel, pois dessa forma os recursos gastos durante o desenvolvimento das mecânicas do jogo seriam mais baixos [12]. Além disso, como o jogo seria adaptado para a narrativa e para inserir conceitos de testes, seria inevitável mudanças na estrutura do jogo, o que consumiria muitos recursos caso o jogo já estivesse implementado.

5.3 Protótipo de papel

Nessa etapa, a jogabilidade foi desenvolvida e testada por meio de protótipos feitos em papel. Apesar de bastante similar a dinâmica do jogo original (*Mastermind* [28]), a jogabilidade precisava também considerar a narrativa criada e os conceitos de testes unitários. Para isso, foi necessário uma maior flexibilidade em relação a possíveis alterações nas mecânicas, o que foi possível com a prototipagem em papel.

A mecânica a ser adaptada era composta por uma sequência de quatro esferas de cores distintas que eram definidas aleatoriamente no início do jogo por outro jogador, o qual era responsável por descobrir a sequência, por meio de tentativa e erro. Para isso, ele escolhia quatro esferas de cores distintas, num total de seis cores diferentes, e as posicionava no tabuleiro. O jogador que inseriu a sequência inicial dizia, então qual foi o resultado da jogada: o número de cores corretas e o número de posições corretas. Então a rodada se repetia até que a quantidade de tentativas terminasse ou até que o jogador acertasse a sequência.

A primeira mudança a ser feita no jogo era o elemento a ser inserido, pois precisava se adaptar à narrativa e também porque a utilização de cor como elemento único de jogabilidade poderia causar problemas em casos de alunos daltônicos [25]. Dessa forma, o novo elemento a ser adotado na jogabilidade seriam armas, como faca, arco e flecha, dentre outras, como pode ser visto na Figura 2.

Essas armas seriam sequenciadas de forma a montar um ataque, que seria utilizado para derrotar um monstro.

Inicialmente, foi verificado se era possível estender a jogabilidade para um número maior de jogadores. Por isso foram feitos dois testes: um com um grupo de cinco pessoas e outro em dupla. Nesses casos, cada um dos jogadores tentava adivinhar a sequência de um jogador definido inicialmente como portador da senha. Porém, foi visto que o jogo ficava cansativo e pouco competitivo, também foi notado que em duplas, o jogador que tinha a senha participava pouco da dinâmica de testes. Dessa forma, como melhoraria foi definido que o jogo final seria jogado individualmente por cada aluno.

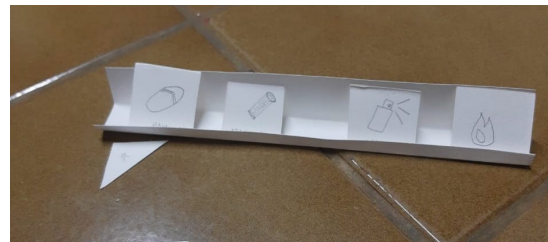


Figura 2: Protótipo em papel do Jogo

Durante os testes, foi observado uma dificuldade em lembrar os últimos ataques, de forma a comparar o impacto da troca de uma sequência para outra. Após inserir essa funcionalidade, os jogadores se sentiram mais seguros em modificar e testar seus ataques, evitando a repetição de jogadas passadas.

Por último, também foi verificado como o jogador poderia ganhar ou perder o jogo, para atender esse requisito foi introduzido o conceito de pontos de vida: cada erro fazia o jogador perder um ponto. Porém, foi notado que os jogadores podiam testar infinitas vezes seus ataques, já que não perderiam pontos, tornando o jogo muito fácil. Por essa razão, foram criados pontos de teste, limitando o número de testes possíveis. Após algumas rodadas, foi visto que a quantidade ideal eram três pontos de vida e cinco pontos de testes, desde que após perder um ponto de vida os jogadores poderiam recuperar seus pontos de teste iniciais.

5.4 Desenvolvimento do jogo para a plataforma Web

Com a narrativa e a jogabilidades definidas, foi possível iniciar o processo de desenvolvimento do jogo final. Com a pandemia do COVID-19, o requisito de tornar o jogo disponível remotamente se tornou imprescindível. Para isso, era necessário o desenvolvimento do jogo para a plataforma *Web*.

Para se adequar a narrativa, o *design* foi baseado na idade média. Para isso foram adicionadas elementos como armas, um boneco de treinamento e um *goblin*. Além disso, também foram utilizados tons de marrom e amarelo, como forma de remeter às cores de um escudo medieval.

Para o *design* do jogo foi utilizado a plataforma *Figma* e exportados em formato *svg*, de modo a facilitar o redimensionamento em diferentes tamanhos de tela, caso algum aluno jogasse utilizando um dispositivo móvel. A Figura 3 ilustra a tela inicial do jogo.

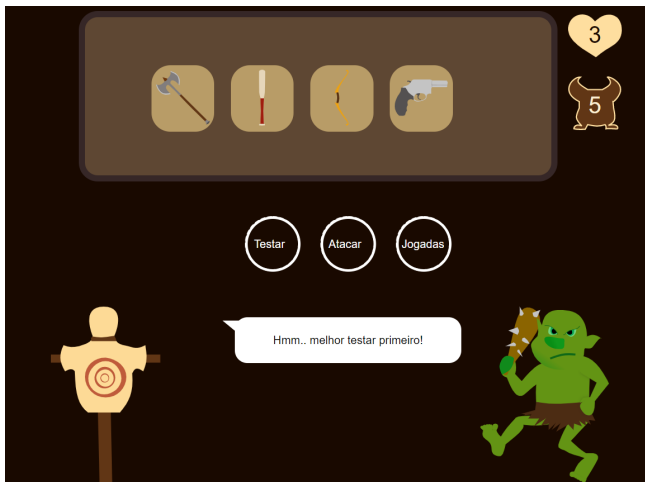


Figura 3: Visão da tela inicial do jogo

O jogo, por sua vez, foi desenvolvido utilizando a linguagem de programação *Javascript*, com auxílio do *framework JQuery*² e foi disponibilizado publicamente³.

A versão final do jogo pode ser observada nas Figuras 3 e 4. Ao abrir o jogo, o aluno visualizará a narrativa e um tutorial de como jogar. Após clicar em "estou pronto", a tela de tutorial desaparece e a área de jogo é mostrada, esta pode ser subdividida em quatro áreas principais. Na parte de cima estão presentes duas delas: o tabuleiro com armas e o painel de pontos de vida e de testes. O tabuleiro com armas possuem células clicáveis, onde o jogador pode tocar em cada uma para alterar a arma que deseja. Ao posicionar o ponteiro do mouse sobre cada célula, ela aumenta de tamanho, bem como o ponteiro do mouse é alterado para o modo clicável, indicando uma possibilidade de interação e, ao pressioná-la, ela pisca para mostrar que uma ação ocorreu.

No painel de pontos, o usuário consegue ver quantos pontos de vida e de testes ainda estão disponíveis para seres utilizados. Ao errar um ataque ou um teste, o fundo da imagem em que o número de pontos está inserido pisca ao mesmo tempo que o número decresce, de maneira a chamar a atenção do jogador. Quando o número de testes chega a zero, o fundo da imagem fica totalmente escurecido.

Logo abaixo do tabuleiro de armas estão os botões de ação, o usuário possui três opções de ação: *Testar*, *Atacar* e ver o histórico de *Jogadas*. Assim como nas células de ataque, os botões também sofrem alterações ao posicionar o ponteiro do mouse em cima de cada botão, indicando a possibilidade de interações através de cliques ou toques. Quando o botão de teste é acionado, o botão escurece e a seguinte frase se torna visível para o jogador: "Altere as armas para continuar", essa funcionalidade foi implementada para evitar ataques iguais, fazendo com que o jogador perca pontos de testes de maneira desnecessária, ao trocar alguma das armas, o botão volta ao seu estado normal. Após uma ou mais tentativas de testes, o jogador pode pressionar o botão *Jogadas*, e o jogo abrirá um painel com todas as últimas jogadas executadas.

²<https://jquery.com>

³<https://xreec.github.io/reset/>

Abaixo dos botões, estão inseridos dois personagens do jogo: o monstro e o boneco de treino. Entre os dois, existe um balão de falas que se altera para sinalizar as respostas de cada um dos personagens. Ao pressionar o botão *Testar*, o balão de falas aponta para o boneco de treino e indica resultado do ataque do jogador. Caso o botão pressionado for o de *Atacar*, então o balão aponta para o monstro, que informa se o ataque foi incorreto e o número de vidas que o jogador ainda possui, caso seja positivo, um *pop-up* aparece indicando que o usuário ganhou o jogo.

No jogo existe também um opção "modo desenvolvedor" que pode ser acessada através de um botão na parte de baixo da interface, ele foi posicionado dessa forma para funcionar como um *Easter egg*⁴. O modo desenvolvedor modifica alguns elementos do jogo, tornando alguns dos conceitos abstratos em conceitos de código e de testes unitários, como o monstro e o boneco de testes que passam a ser, respectivamente, dependência e espião de dependência, em uma referência aos dublês de teste. Essa mudança pode ser vista na Figura 4. O conceito de dublês de testes nesse contexto, servem como uma metáfora, pois o aluno pode escrever um teste unitário utilizando uma dependência real ou usando um dublê de testes, enquanto que no jogo ele pode fazer seu ataque contra um boneco de testes (um dublê) ou contra o monstro (dependência de uma classe).

6 Produção da videoaula

Após o desenvolvimento do jogo, foi possível elaborar a videoaula para explicar os conceitos de testes unitário. Essa preparação foi feita em três passos: relação jogo-aula; desenvolvimento do código; e gravação.

No primeiro passo, foi pensando em como integrar a narrativa dentro da videoaula. Para isso, todos os exemplos, da teoria à prática, foram feitos com base na narrativa. Por exemplo, na parte teórica, o jogo foi apresentado como sistema, um exemplo de teste foi uma função fictícia, por exemplo, *testarAtaque*.

Além disso, a videoaula abordou conteúdos práticos, mostrando para os alunos um passo a passo de como escrever testes unitários. Para essa prática, foi escrito previamente um código em Python simulando a lógica implementada no jogo⁵, de maneira a fazer a ligação com os exemplos vistos dentro do jogo.

Durante a aula, os testes unitários foram escritos usando a metodologia *given-when-then* [16] que estrutura e organiza os testes e conecta com o que o jogo apresentou, por exemplo: "Dado (*given*) que duas armas estão certas, *quando (when)* é chamada a função de *testar, então (then)* o jogador não deverá ganhar o jogo".

Na videoaula foi também introduzido o conceito de dublês de testes, que auxiliavam testes de classes que possuíam outras classes como dependência. Como exemplo, para testar uma resposta era fundamental já conhecer a resposta correta, o que não era possível utilizando a classe original, pois a mesma gerava uma no início do jogo, de modo aleatório. Por isso, foi criado um dublê que continha uma resposta certa, esta era passada no lugar da classe original, tornando o teste viável. Esses dois conceitos foram previamente apresentados no jogo. Fazendo uma analogia ao jogo, a metodologia *given-when-then* [16] foi representada como o teste antes do ataque.

⁴Segundo [27] *Easter egg* é uma função escondida de um jogo, que gera interesse dos estudantes a quererem descobrir "o que vai acontecer se eu apertar nesse botão?".

⁵<https://github.com/xReee/TestesUnitariosTutorial>

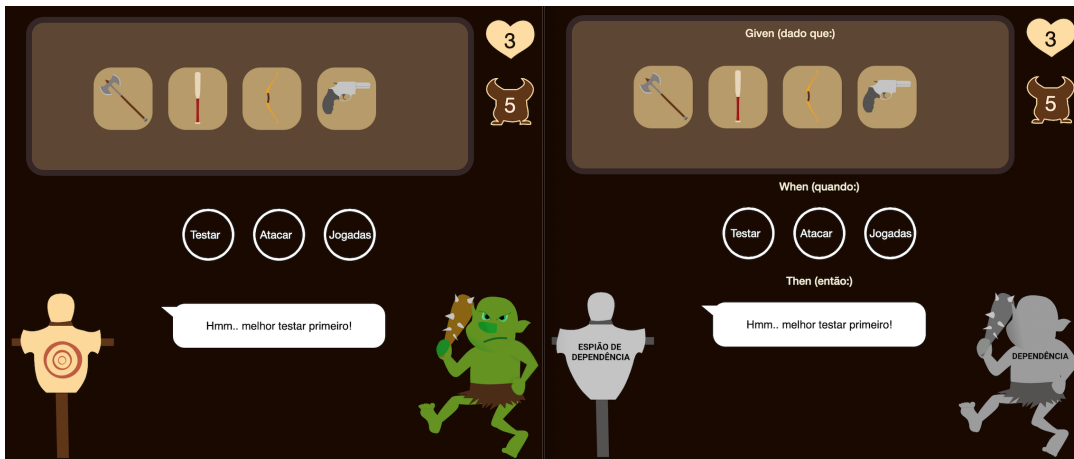


Figura 4: À esquerda o jogo no modo normal, à direita o jogo no modo desenvolvedor.

Já o conceito de dublê foi representado pelo boneco, que era a espião de dependências e o monstro que era a própria dependência. Ambos ficavam ainda mais claros ao ativar o modo desenvolvedor que explicitava cada etapa do sistema e indicava quem eram os atores dos testes.

A videoaula foi apresentada na disciplina de Verificação, Validação e Testes, ministrada para os cursos da área Computação da Universidade Federal do Ceará, por meio de uma *Webconferência* realizada dentro da plataforma Solar⁶. Durante a aula, os alunos podiam esclarecer as dúvidas relativas à videoaula. Esta foi gravada pelo Solar, mas também foi disponibilizada para os alunos posteriormente⁷. A Figura 8 ilustra a interação com os alunos durante a videoaula.

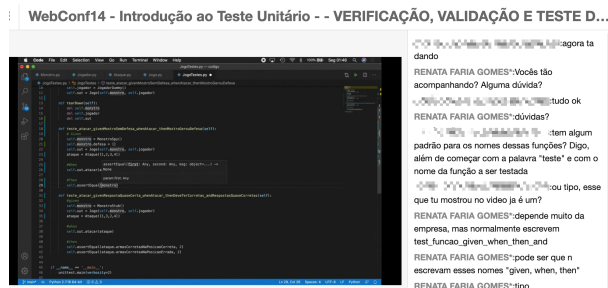


Figura 5: Interação com os alunos na videoaula

7 Elaboração dos exercícios

Para fixar os conteúdos vistos na aula sobre Testes Unitários, na qual foram utilizados o jogo e a videoaula, foi proposto ao final um exercício de fixação composto de duas atividades e um desafio. Na primeira atividade, os alunos precisavam escrever cada um dos três dublês de teste apresentados na aula para as classes *Ataque* do código apresentada na aula.

Na segunda, os alunos precisavam utilizar as respostas da primeira atividade para escrever pelo menos um teste de cada uma

das três funções existentes da classe *Ataque*: *acertouAtaque*, *acertouArma* e *conferirSeGanhou*.

Também foi proposto um desafio com bonificação extra. Dentro do código existiam algumas lógicas incorretas, e para cumprir o desafio os alunos precisavam entregar um teste que falhasse, mostrando que existiria um erro na função que foi testada. Por exemplo, ao errar o ataque no monstro o jogador deveria perder um ponto de vida, porém, no jogo o método *atacar* não estava diminuindo a variável *pontosDeVida*.

O exercício de fixação foi disponibilizado para os alunos em um repositório online⁸ e foi fornecido um tutorial explicando as atividades. Para a resolução, os alunos se organizaram em 9 equipes, a ideia era engajá-los nas atividades e proporcionar uma discussão entre eles promovendo o aprendizado sobre o conteúdo abordado.

8 Resultados e Análise

A metodologia foi aplicada para introduzir o conteúdo de testes unitários na disciplina de V&V. No plano da disciplina, o conteúdo de testes unitários foi abordado após o conteúdo de testes funcionais. Dessa forma, os alunos já haviam trabalhado previamente conceitos introdutórios de teste de software e os tópicos relacionados aos testes funcionais, mas ainda não tinham trabalhado o conteúdo de testes unitários. A aula sobre Testes Unitários ocorreu de forma síncrona através da plataforma Solar. Inicialmente, os alunos foram convidados para participar do jogo e desafiados a finalizá-lo. Para essa atividade foi estabelecido um prazo de 15 minutos. Cada aluno que conseguisse finalizar o jogo, deveria enviar uma foto da tela de finalização para o grupo da disciplina no Telegram. No total, 17 participantes enviaram uma foto da tela de vitória.

Após o jogo, os alunos assistiram a videoaula sobre Testes Unitários que foi previamente gravada. Em seguida, os alunos foram convidados a avaliar o jogo e a videoaula por meio de um questionário⁹ elaborado com base no MEEGA+[23], com respostas utilizando a escala Likert de 5 pontos[7]. Ao final da aula foi apresentado o exercício de fixação no qual os alunos tiveram o prazo de uma semana para concluí-lo. Vale ressaltar que os alunos foram divididos

⁶<https://solar.virtual.ufc.br/>

⁷<https://youtube.com/watch?v=R6aWbck5KRG>

⁸<https://github.com/xReee/TestesUnitariosTutorial>

⁹<https://forms.gle/7DYQQRVXvSudXnqT6>

em 9 grupos para resolver o exercício proposto. Ao entregarem o exercício, os alunos foram convidados a responder um segundo questionário¹⁰. O objetivo deste questionário era que os alunos pudessem avaliar a metodologia aplicada (jogo + videoaula) com base nas dificuldades enfrentadas durante a resolução do exercício de fixação (e.g., escrita dos testes unitários). Além disso, foi possível avaliar a metodologia aplicada por meio das correções dos exercícios e de um fórum online no Solar em que os alunos foram convidados a deixar suas opiniões sobre o jogo.

A seguir são apresentados os resultados dessas avaliações.

8.1 Avaliação do jogo

A metodologia proposta foi aplicada para os alunos na disciplina de Verificação, Validação e Testes de Software da Universidade Federal do Ceará.

Ao total, 15 alunos responderam o formulário, um do curso de Sistema e Mídias digitais, dois de Engenharia da Computação e os demais de Ciência da Computação. Todos os participantes são alunos do sétimo semestre ou superior, a maioria deles tinham idades entre 22 e 25 anos. Dos respondentes, apenas um não costumava jogar jogos digitais.

Além disso, 1 não tinha experiência prévia em testes unitários, 11 dos participantes tinham experiência intermediária, 2 tinham experiência, 1 tinha bastante experiência. Nenhum dos alunos tinham jogado algum jogo relacionado ao ensino de testes anteriormente.

Dessa forma, foi possível medir aspectos da usabilidade que subdivide-se em estética e aprendizagem, e aspectos da experiência que subdivide-se em desafio, satisfação, diversão, atenção focada, relevância e percepção de aprendizagem. Cada pergunta era respondida com uma escala Likert de 5 pontos, variando de 1 (discordo totalmente) até 5 (concordo totalmente).

Os resultados do formulário foram divididos em duas principais categorias: *Usabilidade* e *Experiência*.

Na categoria de *Usabilidade*, o subtópico de estética possuía duas perguntas. Na primeira, 11 respostas concordavam e 4 concordavam totalmente que o *design* do jogo era intuitivo e atraente. Na segunda, 6 concordavam e 9 concordavam totalmente que os textos, as cores e as fontes combinavam. Além disso, no subtópico aprendizagem (em relação a jogabilidade), 7 concordaram totalmente, 7 concordaram e 1 discordou totalmente que o jogo foi fácil de se aprender a jogar.

Na categoria *Experiência*, os subtópicos desafio, satisfação, diversão e atenção focada foram avaliados. Em relação ao desafio, 6 concordaram totalmente, 6 concordaram, 2 mantiveram-se neutros, 1 discordou totalmente que o jogo é adequadamente desafiador. Em relação à se os alunos recomendariam a metodologia para amigos (satisfação), 8 concordavam totalmente, 6 concordavam e 1 discordou.

Na questão de diversão, 4 concordaram totalmente, 9 concordaram e 2 mantiveram-se neutros em relação a diversão do jogo. Em atenção focada, 5 dos alunos concordaram totalmente e 10 concordaram que o jogo prendeu a atenção.

A seção de relevância tinham cinco perguntas. A pergunta se o conteúdo do jogo é relevante 10 dos alunos concordaram e 5 concordaram totalmente. Já a seção de percepção de aprendizagem tinham quatro perguntas. A pergunta de se os alunos preferem

aprender com essa metodologia do que com outro método obteve 7 respostas concordando totalmente, 6 concordando e 2 neutras. O resultado das outras perguntas pode ser verificada por meio da Figura 6.

Nas questões subjetivas, os alunos afirmam que o que mais gostaram foi a ludicidade da dinâmica, afirmando que ela contribuiu para o entendimento dos exemplos e para o engajamento durante a aula. Por outro lado, alguns alunos comentaram que a videoaula gravada apresentada durante a aula foi muito longa e não gostaram dela ter sido gravada. Além disso, alguns alunos preferiam que os conceitos de testes dentro do jogo estivesse mais explicitados.

8.2 Avaliação do exercício de fixação

O formulário de avaliação de exercício foi respondido por 8 alunos, desses 1 indicou zero como seu nível de conhecimento de testes unitários, 3 marcaram como 1, 3 responderam com 2, 1 respondeu 3 e nenhum marcou o valor máximo. Além disso, 5 alunos afirmaram que antes da aula não conseguiriam fazer os exercícios, em contraste com 2 que afirmaram que conseguiriam, 1 aluno manteve-se neutros.

Nesse contexto, 7 alunos concordaram que o objetivo da metodologia e das atividades foram claramente definidos, apenas 1 manteve-se neutros. Além disso, todos os alunos concordaram que que os critérios de avaliação foram bem estabelecidos.

Em relação ao uso de conhecimentos adquiridos durante a aplicação da metodologia, 6 alunos afirmaram que as atividades facilitaram a resolução dos exercícios e todos concordaram que utilizaram conhecimentos adquiridos durante a aula para resolução das atividades propostas.

Por fim, todos os alunos concordaram que as atividades propostas foram condizentes com o conteúdo proposto. No entanto, 2 alunos discordaram que o desafio estava numa atividade moderada, enquanto que 1 manteve-se neutro e 5 concordaram com a afirmação.

O resultado detalhado deste questionário é ilustrado na Figura 7. Além disso, os alunos avaliaram em uma escala de 1 a 10 a metodologia aplicada, todas as notas foram acima de 8 conforme ilustrado na Figura 8, revelando que os alunos ficaram satisfeitos com as atividades.

8.3 Avaliação pelo fórum

Os alunos também interagiram utilizando um fórum da plataforma *Solar*. Para incentivar a participação dos alunos, foram disponibilizadas duas perguntas de partida:

- O que vocês acharam do Jogo? Ele facilitou o entendimento dos conceitos introdutórios de teste unitário?
- Cite uma dificuldade que você enfrentou na escrita dos testes unitários.

Em suma, os alunos responderam na primeira pergunta que gostaram do jogo, que facilitou o entendimento de conceitos, porém, em alguns comentários, observou-se que esse entendimento só ocorreu por meio do "modo desenvolvedor", no qual os conceitos são mais explícitos. Os alunos também comentaram que o jogo auxiliou na entrega das atividades propostas. Alguns exemplos das respostas foram: "*Gostei do jogo. A ferramenta de desenvolvedor principalmente, realmente ajudou a compreender os conceitos do teste unitário.*" e

¹⁰<https://forms.gle/PuFPDpvbysTMhAsw8>

| USABILIDADE | | Discordo totalmente | Discordo | Neutro | Concordo | Concordo totalmente |
|--------------|--|---------------------|----------|--------|----------|---------------------|
| Estética | O design do jogo é intuitivo e atraente. | 0 | 0 | 0 | 11 | 4 |
| | Os textos, cores e fontes combinam e são consistentes. | 0 | 0 | 0 | 6 | 9 |
| Aprendizagem | Aprender a jogar este jogo foi fácil para mim. | 1 | 0 | 0 | 7 | 7 |

| Experiência | | Discordo totalmente | Discordo | Neutro | Concordo | Concordo totalmente |
|---------------------------|---|---------------------|----------|--------|----------|---------------------|
| Desafio | Este jogo é adequadamente desafiador, as tarefas não são muito fáceis nem muito difíceis. | 1 | 0 | 2 | 6 | 6 |
| Satisfação | Eu recomendaria esta metodologia (jogo + videoaula) para meus colegas que querem aprender testes unitários. | 0 | 1 | 0 | 6 | 8 |
| Diversão | Eu me diverti com o jogo. | 0 | 0 | 2 | 9 | 4 |
| Atenção Focada | O jogo prendeu minha atenção. | 0 | 0 | 0 | 10 | 5 |
| Relevância | É claro para mim como o conteúdo do jogo está relacionado com o conteúdo introdutório de teste unitário abordado na videoaula | 0 | 1 | 0 | 9 | 5 |
| | O jogo facilitou a introduzir conceitos que foram posteriormente explicados na videoaula | 0 | 0 | 0 | 9 | 6 |
| | O conteúdo do jogo é relevante ao aprendizado. | 0 | 0 | 0 | 10 | 5 |
| | Após o jogo e a videoaula me senti motivado a aprimorar meus conhecimentos em testes unitários. | 1 | 0 | 2 | 8 | 4 |
| | A metodologia (videoaula + jogo) é adequada para uma disciplina introdutória de testes unitários. | 0 | 0 | 0 | 5 | 10 |
| Percepção de Aprendizagem | Após o jogo e a videoaula, eu me sinto capaz de escrever testes unitários | 0 | 1 | 5 | 5 | 4 |
| | Quanto você acha que este jogo contribuiu para seu aprendizado em teste unitário? | 0 | 0 | 3 | 10 | 2 |
| | O quão eficiente foi a metodologia (videoaula + jogo) para sua aprendizagem, comparando-o com outras atividades da disciplinas (e.g., fóruns, trabalhos)? | 0 | 0 | 1 | 6 | 8 |
| | Eu prefiro aprender com esta metodologia (videoaula + jogo) do que de outra forma (outro método de ensino). | 0 | 0 | 2 | 6 | 7 |

Figura 6: Resumo das respostas do questionário do jogo

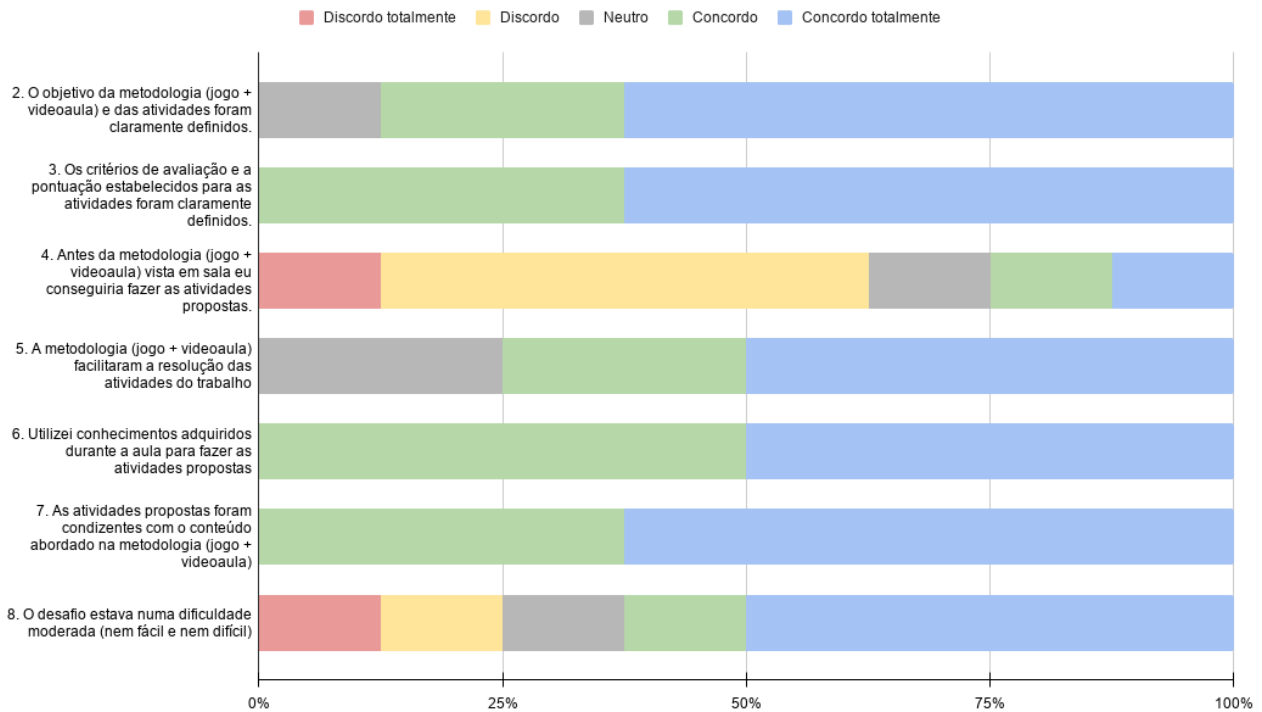


Figura 7: Relatório das respostas do questionário do exercício

12. Em uma escala de 1 a 10, como você avalia a metodologia e as atividades propostas no trabalho sobre Teste Unitário?

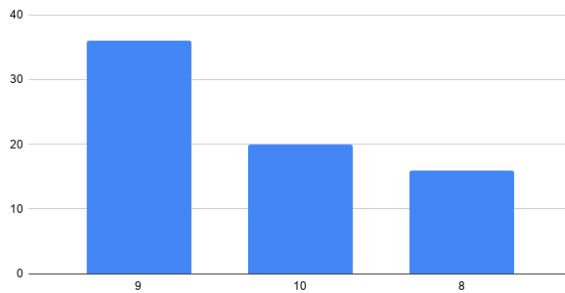


Figura 8: Avaliação final da metodologia

"Achei muito interessante!! Ele ajudou bastante a raciocinar sobre o trabalho e a elaborar os casos de teste."

Em relação à segunda pergunta, os alunos relataram dificuldades relacionadas à linguagem escolhida para a atividade (*Python*), às diferenças de cada dublê, aos possíveis casos de testes e ao verificar se o teste estava funcionando corretamente. Alguns exemplos das respostas foram: *"Como o conceito ficou claro, tive mais dificuldades com a linguagem Python do que propriamente com a estrutura dos testes unitário."*; *"Senti dificuldade de entender o papel e as características de cada dublê."*; e *"Dependendo do caso, senti uma certa dificuldade*

em entender de fato o que precisa ser testado, principalmente nos testes de Ataque."

8.4 Correção do exercício de fixação

Por meio da correções dos exercícios de fixação, foi possível entender quais assuntos os alunos entenderam melhor e quais deles foram mais desafiadores. Os exercícios eram parte obrigatória da metodologia e seriam pontuados como parte da nota da disciplina. Porém, como o desafio era opcional, foi possível medir o engajamento das atividades de fixação por meio dele.

As notas do exercício variavam de 0 a 1. Já o desafio valia pontos extras, caso o aluno conseguisse acertar o desafio, ele ganharia 0.3 pontos extras.

Do total de 9 grupos que realizaram os exercícios, 4 obtiveram a nota máxima, sendo que destes, dois também concluíram corretamente o desafio. Além desses, 3 grupos obtiveram taxa de acerto entre 80% e 90% no exercício, e apenas 2 grupos obtiveram taxas entre 60% e 70%. Sendo assim, observou-se que todos o grupos acertaram pelo menos 60% do exercício. Em relação ao desafio, 7 grupos tentaram fazer, porém, apenas 2 conseguiram responder corretamente.

Pela resolução do exercício, observou-se que os alunos conseguiram assimilar corretamente o conceito de *given-when-then*, pois todos os grupos escreveram seus testes utilizando essa estrutura. Além disso, os grupos conseguiram acertar a implementação dos dublês *spy* e *dummy*.

Das dificuldades percebidas durante a correção, foi visto que os alunos não compreenderam bem a função do duplê *stub*. Os erros mais comuns foi a implementação de um *stub* para cada caso de teste e *stubs* chamando funções do *SUT* no lugar de enviar um dado esperado. Além disso, observou-se que a maioria dos grupos optou pelo uso de *stubs*, porém os que utilizaram *spy* para a realização da atividade, conseguiram concluir com êxito os testes.

Em relação à implementação de duplês, três grupos não implementaram *spy* e um não implementou *dummy*, podendo isso ser um problema de definição dos objetivos da atividade.

9 Conclusão e Trabalhos futuros

Testar um software é extremamente importante para o mercado de tecnologia, uma vez que garante o correto funcionamento de aplicações que estão presentes no cotidiano das pessoas e que poderiam causar grandes prejuízos caso apresentassem erros. Uma das causas desse problema pode estar relacionada ao ensino muito teórico desse conhecimento.

Dessa forma, a gamificação se apresenta como uma possível solução para esse problema, pois possui fatores que podem engajar estudantes de graduação. No entanto, os trabalhos identificados na literatura durante esta pesquisa não abordam diretamente a escrita de testes unitários ou não utilizam narrativas, um fator-chave no contexto de gamificação do ensino para engajar os alunos no processo de aprendizagem.

Nesse cenário, esse artigo apresentou uma metodologia composta por três principais etapas que se interligam entre si por meio de uma narrativa. Uma avaliação foi conduzida com os alunos de uma disciplina de Verificação, Validação e Teste de Software. Os resultados indicaram que os alunos ficaram satisfeitos com o aprendizado, uma vez que observou-se que todos o grupos acertaram pelo menos 60% do exercício. Portanto, pode-se afirmar que a metodologia proposta ajudou os alunos no engajamento das atividades bem como promoveu o aprendizado do ensino de testes unitários uma vez que os alunos conseguiram concluir as atividades propostas com êxito.

Como trabalhos futuros, pretende-se melhorar o modo desenvolvedor e trabalhá-lo como o modo principal. Além disso, planeja-se levar o jogo para a plataforma analógica, visando futuras aulas presenciais, para isso, é necessário estudar e buscar uma jogabilidade que permita que todos os usuários participem dos testes.

Referências

- [1] Ieee standard glossary of software engineering terminology. *IEEE Std 610.12-1990* (1990), 1–84.
- [2] ARCURI, A., FRASER, G., AND JUST, R. Private api access and functional mocking in automated unit test generation. In *2017 IEEE International Conference on Software Testing, Verification and Validation (ICST)* (2017), pp. 126–137.
- [3] BASTOS, A. S., GOMES, R. F., DOS SANTOS, C. C., AND MAIA, J. G. R. Synesthesia: A study on immersive features of electronic games. *SBC Journal on Interactive Systems* 9, 2 (2018), 38–51.
- [4] BEPPE, T. A., DE ARAÚJO, Í. L., ARAGÃO, B. S., DE SOUSA SANTOS, I., XIMENES, D., AND ANDRADE, R. M. C. Greatest: a card game to motivate the software testing learning. In *Proceedings of the XXXII Brazilian Symposium on Software Engineering* (2018), pp. 298–307.
- [5] COELHO, R., KULESZA, U., VON STAA, A., AND LUCENA, C. Unit testing in multi-agent systems using mock agents and aspects. In *Proceedings of the 2006 international workshop on Software engineering for large-scale multi-agent systems* (2006), pp. 83–90.
- [6] DA SILVA NUNES, R. Leituras do poder na terceira era da terra-média: um ensaio baseado em o senhor dos anéis. *Espaço e Cultura*, 41 (2017), 148–166.
- [7] DEVELLIS, R. F. *Scale development: Theory and applications*, vol. 26. Sage publications, 2016.
- [8] DICKEY, M. D. Murder on grimm isle: The impact of game narrative design in an educational game-based learning environment. *British journal of educational technology* 42, 3 (2011), 456–469.
- [9] ELBAUM, S., PERSON, S., DOKULIL, J., AND JORDE, M. Bug hunt: Making early software testing lessons engaging and affordable. In *29th International Conference on Software Engineering (ICSE'07)* (2007), IEEE, pp. 688–697.
- [10] FRASER, G. Gamification of software testing. In *2017 IEEE/ACM 12th International Workshop on Automation of Software Testing (AST)* (2017), IEEE, pp. 2–7.
- [11] FREEMAN, S., MACKINNON, T., PRYCE, N., AND WALNES, J. Mock roles, not objects. In *Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications* (2004), pp. 236–246.
- [12] FULLERTON, T., SWAIN, C., AND HOFFMAN, S. *Game design workshop: Designing, prototyping, & playtesting games*. CRC Press, 2004.
- [13] HENRIQUE DIAS VALLE, P., TODA, A. M., BARBOSA, E. F., AND MALDONADO, J. C. Educational games: A contribution to software testing education. In *2017 IEEE Frontiers in Education Conference (FIE)* (2017), pp. 1–8.
- [14] IKE, T. C., HOE, T. W., KIM, J. L. E., AND Y'NG, N. Y. Exploring user experience from an emotional context when designing immersive games for education. *Journal of ICT in Education* 8, 1 (2021), 10–25.
- [15] KIRYAKOVA, G., ANGELOVA, N., AND YORDANOVA, L. Gamification in education. Proceedings of 9th International Balkan Education and Science Conference.
- [16] KORHONEN, J. Automated model generation using graphwalker based on given-when-then specifications, 2020.
- [17] KROGEN, E. Bond: A spy-based testing and mocking library. Tech. rep., Technical Report. Electrical Engineering and Computer Sciences, University of . . . , 2016.
- [18] LELLI, V., ANDRADE, R. M., FREITAS, L. M., SILVA, R. A., FILHO, F. G. S., GOMES, R. F., AND DE OLIVEIRA SEVERO, J. S. Gamification in remote teaching of se courses: Experience report. In *Proceedings of the 34th Brazilian Symposium on Software Engineering* (2020), pp. 844–853.
- [19] MACKINNON, T., FREEMAN, S., AND CRAIG, P. Endo-testing: unit testing with mock objects. *Extreme programming examined* (2000), 287–301.
- [20] MALONE, T. W. Toward a theory of intrinsically motivating instruction. *Cognitive science* 5, 4 (1981), 333–369.
- [21] MARQUES, A., RAMALHO, F., AND ANDRADE, W. L. Comparing model-based testing with traditional testing strategies: An empirical study. In *2014 IEEE Seventh International Conference on Software Testing, Verification and Validation Workshops* (2014), IEEE, pp. 264–273.
- [22] MESZAROS, G. *xUnit test patterns: Refactoring test code*. Pearson Education, 2007.
- [23] PETRI, G., VON WANGENHEIM, C. G., AND BORGATTO, A. F. Meega+: an evolution of a model for the evaluation of educational games. *INCoD/GQS* 3 (2016), 1–40.
- [24] POLKINGHORNE, D. E. *Narrative knowing and the human sciences*. Suny Press, 1988.
- [25] RAMOS, L. R. B. Proposta de modo de acessibilidade para o jogo nihilumbra: tornando o jogo mais acessível para jogadores daltônicos.
- [26] SALEN, K., TEKINBAŞ, K. S., AND ZIMMERMAN, E. *Rules of play: Game design fundamentals*. MIT press, 2004.
- [27] STEIN, K. A., AND BARTON, M. H. The “easter egg” syllabus: Using hidden content to engage online and blended classroom learners. *Communication Teacher* 33, 4 (2019), 249–255.
- [28] STUCKMAN, J., AND ZHANG, G.-Q. Mastermind is np-complete. *arXiv preprint cs/0512049* (2005).
- [29] VALLE, P., BARBOSA, E. F., AND MALDONADO, J. Um mapeamento sistemático sobre ensino de teste de software. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)* (2015), vol. 26, p. 71.