

## SERIOUS GAME PARA APRENDIZAGEM DE FUNDAMENTOS DE TESTE DE SOFTWARE

Samara Cardoso (IC) e Pollyana Coelho da Silva Notargiacomo (Orientadora)

**Apoio: PIVIC Mackenzie**

### RESUMO

O mercado brasileiro de desenvolvimento de software tem alcançado projeção global, com um aumento notável de empresas e receitas substanciais. No entanto, esse crescimento é acompanhado pela falta de profissionais qualificados para atender à crescente demanda. Esse desequilíbrio entre a oferta de mão de obra e as necessidades do mercado é especialmente evidente no campo dos testes de software, no qual há uma significativa discrepância entre os currículos acadêmicos e as exigências da indústria. Além disso, o ensino da disciplina de testes nas universidades brasileiras de tecnologia enfrenta desafios notáveis, sobretudo devido à ausência de diretrizes claras, resultando em variações substanciais na inclusão e abordagem desses conteúdos. Nesse contexto, este artigo foca no desenvolvimento e validação de um *serious game* projetado para complementar o ensino dos fundamentos dos testes de software para estudantes de graduação e profissionais da área de tecnologia. A pesquisa, de abordagem qualitativa, foi conduzida com alunos e recém-formados da Faculdade de Ciência e Informática da Universidade Presbiteriana Mackenzie. Os resultados destacam que o *serious game* apresenta potencial para contribuir na aquisição de conhecimento relacionado ao tema proposto, o qual os participantes previamente não dominavam. Dessa forma, o estudo apresenta uma ferramenta de apoio que tem o intuito de contribuir para reduzir as deficiências no ensino dessa disciplina crucial, visando capacitar os profissionais de forma mais eficaz para enfrentar os desafios do cenário tecnológico.

**Palavras-chave:** jogos sérios, teste de software, engenharia de software

### ABSTRACT

The Brazilian software development market has reached global projection, with a notable increase in companies and substantial revenues. However, this growth is accompanied by a lack of qualified professionals to meet the growing demand. This imbalance between labor supply and market needs is especially evident in the domain of software testing, where there is a significant discrepancy between academic curricula and industry requirements. Furthermore, teaching the subject of testing at Brazilian universities of technology faces notable challenges, mainly due to the lack of clear guidelines, resulting in substantial variations in the

inclusion and approach of these contents. In this context, this article focuses on the development and validation of a serious game designed to complement the teaching of software testing fundamentals to undergraduate students and technology professionals. The research, with a qualitative approach, was carried out with students and recent graduates from the Faculty of Sciences and Informatics of Universidade Presbiteriana Mackenzie. The results highlight that the serious game has the potential to contribute to the acquisition of knowledge related to the proposed topic, which the participants previously did not master. In this way, the study contributes by presenting a support tool that aims to contribute to the reduction of deficiencies in the teaching of this crucial subject, aiming to train professionals more effectively to face the challenges of the technological scenario.

**Keywords:** serious games, software testing, software engineering

## 1. INTRODUÇÃO

O mercado brasileiro de desenvolvimento de software possui relevância mundial. A pesquisa realizada pela Associação Brasileira de Empresas de Software (ABES) em 2020 revelou que 24.135 empresas nacionais voltadas para o desenvolvimento, produção e distribuição de software obtiveram um faturamento superior a 54 milhões de dólares no ano anterior. Esses números posicionam o Brasil como o 9º maior mercado de desenvolvimento de software globalmente e o 1º na América Latina (ABES, 2021). Enquanto o mercado continua a crescer, surge uma lacuna de profissionais qualificados no Brasil. Segundo um relatório da Brasscom (GALLINDO; OLIVEIRA, 2021), entre 2021 e 2025, o setor de tecnologia deve criar 797 mil vagas, com remuneração três vezes superior à média salarial de outras áreas. Adicionalmente, o segundo semestre de 2022 registrou um número de desempregados no Brasil chegando a 10,1 milhões, de acordo com o IBGE. Assim, existem vagas, mas há falta de profissionais capacitados para preenchê-las.

Embora as instituições de tecnologia desempenhem um papel nevrálgico na formação dos futuros profissionais, há uma desconexão entre os objetivos da indústria e os currículos acadêmicos. Isso é particularmente evidente na formação de profissionais na área de testes de software, na qual a falta de diretrizes comuns resulta em variações nas decisões sobre obrigatoriedade, conteúdo e carga horária da disciplina (BESSA; PESSOA; LOBÃO, 2019).

Como complemento à formação acadêmica, as certificações têm adquirido popularidade ao abordar conteúdos teóricos e práticos necessários ao mercado. As certificações confirmam habilidades adequadas para determinadas ocupações e podem ser emitidas pela indústria ou por associações profissionais (CANTOR, 2002). No campo de testes de software, a certificação CTFL, emitida pelo ISTQB, é especialmente relevante.

Este artigo aborda o desenvolvimento e a avaliação de um *serious game* para apoiar o ensino de testes de software. O objetivo é proporcionar uma ferramenta complementar ao ensino formal de testes em cursos de tecnologia e auxiliar profissionais interessados em aprofundar seus conhecimentos na área. O conteúdo do jogo é fundamentado nas obras de Myers *et al.* (2012) e Pressman (2010), autores relevantes nas ementas da área de Engenharia de Software. O presente *serious game*, denominado de "Jogos Dourados", coloca o jogador no papel de um estagiário de qualidade em uma *startup* de jogos educativos, desafiando-o a aplicar conceitos essenciais de qualidade no desenvolvimento de software.

Com base nas questões levantadas, este artigo se estrutura da seguinte forma: a seção 2 explora os elementos teóricos que embasam a investigação; a seção 3 detalha os

procedimentos metodológicos do desenvolvimento e teste do *serious game* proposto; e a seção 4 apresenta as conclusões da pesquisa e próximos passos.

## 2. REFERENCIAL TEÓRICO

### 2.1 ENGENHARIA DE SOFTWARE

Softwares profissionais são construídos por equipes organizadas de maneira estruturada (com papéis definidos), que serão capazes de prestar suporte ao produto e realizar melhorias durante sua vida útil. A engenharia de software é uma disciplina que aplica os conceitos da engenharia na construção de programas de computador por meio do uso de abordagens sistemáticas, disciplinadas e quantificáveis ao longo de todo o processo de desenvolvimento (VALENTE, 2022).

Na engenharia de software, um "processo" não é uma prescrição que deve ser seguida com rigidez para construir programas. É necessário adaptar o processo às pessoas que fazem parte do time de desenvolvimento, de forma a garantir que o software seja entregue em tempo hábil e com qualidade para o cliente (PRESSMAN, 2010). No intuito de organizar as atividades necessárias para a construção de software, alguns autores dividem o processo em etapas mais ou menos flexíveis.

Para Sommerville (2011), o processo de software inclui quatro atividades fundamentais para a engenharia de software: Especificação de Software, Desenvolvimento de Software, Validação de Software e Evolução de Software. Na atividade de *Especificação*, os requisitos e as limitações são identificados e, dependendo da metodologia utilizada, é produzida uma documentação mais ou menos robusta. O *Desenvolvimento* é a atividade de codificação, em que ocorre a construção do programa, seguido pela atividade de *Validação*, na qual o software é testado para garantir que cumpre os requisitos levantados. Por fim, existe a evolução, sendo que o software é modificado de acordo com as novas necessidades identificadas junto ao cliente.

De maneira complementar, Valente (2022) aponta que os modelos sequenciais (como o "cascata") possuem uma ordem de etapas a serem seguidas: levantamento de requisitos, análise (ou projeto de alto nível), projeto detalhado, codificação e testes. Todo o processo de desenvolvimento é realizado de forma sequencial e encadeado, com critérios de entrada e saída claros. Devido às características do modelo, o produto a ser entregue no final do projeto atende aos requisitos definidos no processo de especificação, com restritas possibilidades de

alterações. Esses modelos são indicados para projetos críticos de segurança, que possuem requisitos definidos de maneira precisa e que dificilmente sofrerão mudanças ao longo do processo de desenvolvimento.

Entretanto, em um ambiente de negócios dinâmico e com mudanças constantes, o uso do modelo anterior não é adequado porque os requisitos estabelecidos no início do projeto podem não fazer mais sentido para o cliente quando o programa é entregue (SOMMERVILLE, 2011). Nos modelos ágeis, como o *Scrum* e o *Extreme Programming*, o desenvolvimento do software ocorre de forma iterativa e incremental, sendo que, ao final de cada ciclo de cerca de um mês (VALENTE, 2022), existe uma entrega de um programa funcional para o usuário.

## 2.2 TESTES DE SOFTWARE

De acordo com Myers *et al.* (2012, p. 16), "*Teste de software é um processo, ou uma série de processos, projetado para ter certeza que o código de computador faz o que foi projetado para fazer e, inversamente, que não faz nada involuntário*". É por meio do teste que a equipe responsável pelo projeto consegue certificar que o programa não apresenta problemas que impeçam seu funcionamento correto e que ele atende às necessidades do usuário.

A atividade de teste de software, entretanto, não é capaz de garantir que um programa não possui defeitos ("bugs"), apenas apontar a presença deles (MYERS; SANDLER; BADGETT, 2012; VALENTE, 2022). Independentemente da complexidade do programa, é praticamente impossível afirmar que o programa não possui defeitos, dado que o teste de todas as combinações de entradas válidas e inválidas, quando matematicamente possível, inviabilizaria economicamente o projeto por conta do tempo e recursos necessários. Os testes, portanto, devem ser capazes de identificar a maior quantidade possível de defeitos com os recursos disponíveis, e garantir que estes erros sejam corrigidos antes que o programa esteja disponível em ambiente de produção.

Dada a impossibilidade de realizar testes exaustivos, o sucesso do processo de teste está associado à qualidade do conjunto de casos de testes, e o uso de critérios de seleção é útil para determinar quais testes deverão ser realizados. Esses critérios estão divididos em três técnicas complementares de testes: funcional, estrutural e baseada em erros. As técnicas aplicam abordagens diferentes em função da origem dos dados e caminhos que o testador deve seguir para validar as informações processadas (BARBOSA *et al.*, 2000).

A técnica funcional (ou "caixa-preta") permite que o sistema seja validado sem a necessidade de que o testador possua conhecimento sobre sua estrutura interna (SOMMERVILLE, 2011).

Na técnica estrutural (ou "caixa-branca") o testador deriva os testes a partir da análise do funcionamento interno da aplicação. Os testes modelados com essa técnica verificam o funcionamento das estruturas sequenciais e das condições, e permitem avaliar partes reduzidas do código, sem que seja necessária a construção completa do sistema antes do teste.

Por fim, a técnica baseada em erros utiliza o conhecimento sobre os enganos mais comuns cometidos no processo de desenvolvimento para avaliar o programa e criar estratégias para localizá-los (BARBOSA *et al.*, 2000).

O tópico final a ser abordado nesta seção diz respeito à distinção entre o emprego de técnicas de testes estáticos e dinâmicos. Os testes estáticos são conduzidos por meio da análise da documentação disponível da funcionalidade, como fluxogramas, histórias de usuário e códigos-fonte. Essas análises viabilizam a identificação e correção de problemas na documentação antes mesmo da fase de codificação, resultando na redução de custos associados à correção de falhas em estágios subsequentes.

Já para a aplicação das técnicas dinâmicas de teste, é necessário que haja uma aplicação executável. Esses testes são capazes de desvendar defeitos mais complexos, que os testes estáticos geralmente não conseguem identificar. Por essa razão, a utilização complementar de ambas as técnicas é altamente recomendada (PRESSMAN, 2010).

### 2.2.1 Ensino de Teste de Software

Apesar da importância da atividade de teste no processo de desenvolvimento, o ensino do conteúdo nas universidades brasileiras de tecnologia carece de diretrizes específicas, principalmente no que diz respeito ao conteúdo prático. Benitti e Albano (2012) apontam para uma falta de clareza nos currículos de referência nacionais sobre como a disciplina de testes deve ser ensinada. Ao analisar entidades brasileiras, os autores constataram que todas elas reconhecem a necessidade de inclusão do assunto de testes em sala de aula, mas não trazem normativas concretas sobre a forma que o conteúdo deve ser lecionado, deixando a decisão a cargo das instituições de Ensino. Como exemplo disso se pode citar que, no ano de 2010, entre 28 planos de ensino de nove diferentes universidades

com cursos em tecnologia avaliadas, apenas 11 possuíam uma disciplina exclusivamente voltada ao ensino de Teste de Software (BENITTI; ALBANO, 2012).

Inclusive, esses dados são corroborados por Paschoal e Souza (2018), que conduziram uma pesquisa qualitativa com 52 professores atuantes na área de tecnologia, a fim de mapear o panorama do ensino de teste de software no Brasil. Dentre as conclusões do estudo, se verificou que apenas 46% dos professores lecionavam os conceitos de teste em uma matéria exclusiva para isso e que, entre elas, existia variância considerável na nomenclatura e na carga horária atribuída. A mesma pesquisa também constatou que o conteúdo ensinado nas universidades, seja nas disciplinas exclusivas ou naquelas que incluem o tópico de testes não seguem um padrão, variando conforme a escolha pedagógica do professor. Apesar disso, o ensino dos conceitos teóricos básicos, como níveis e técnicas de teste está presente em todos os programas.

O problema apresentado, no entanto, não se limita apenas ao cenário brasileiro. Ao comparar o currículo de teste de software nacional com universidades fora do país (VALLE; BARBOSA; MALDONADO, 2015), observou-se que em ambos, o conteúdo sobre testes é comumente lecionado como um tópico de Engenharia de Software ou Fundamentos de Programação, sem a integração com outras disciplinas e com um tempo reduzido dedicado. Essa decisão pedagógica resulta na formação de profissionais com um conhecimento insuficiente sobre testes e, conseqüentemente, sem as qualificações técnicas necessárias para adentrarem a indústria de tecnologia na área de qualidade.

A carência de diretrizes para o ensino de testes traz implicações na formação dos profissionais de qualidade e na absorção deles pela indústria de tecnologia. Garousi *et al.* (2020) avaliaram como a indústria enxerga o papel da academia na produção de soluções que se vinculem às necessidades do mercado de trabalho. Esta investigação mapeou que, entre os maiores desafios enfrentados pelos entrevistados no âmbito do processo de teste de software, o “treinamento dos profissionais” representou 10% das respostas obtidas, a maior porcentagem entre as opções disponíveis.

Como uma resposta à reduzida correspondência entre as expectativas da indústria e o ensino das universidades de tecnologia em geral, as certificações surgiram nos anos 90 e têm se popularizado desde então como um complemento à educação formal para validar o aperfeiçoamento de competências específicas adquiridas dentro da academia (RANDALL; ZIRKLE, 2005). Nesse cenário, as certificações em Teste de Software desempenham um papel importante para os profissionais de qualidade porque atestam um conhecimento fundamental

da atividade e as habilidades necessárias para desenvolvê-la. Dentre as certificações disponíveis na área, a CTFL é a mais obtida e valorizada pelos profissionais (BESSA; PESSOA; LOBÃO, 2019). A CTFL é uma certificação de nível fundamental na área de testes, provida pela certificadora ISTQB, que possui representantes regionais em diversos países, inclusive no Brasil. (BOARD, 2023).

### 2.3 *SERIOUS GAMES*

Desde os primórdios, seres humanos e animais jogam. O jogo é uma prática voluntária, que se contrapõe à seriedade e que permite ao jogador distanciar-se da vida “real” para participar de uma atividade de faz-de-conta (HUIZINGA, 2008). O primeiro jogo digital desenvolvido foi o “Tennis for Two”, em 1958, no qual o jogador controlava uma bolinha de tênis por meio de um osciloscópio (BATISTA; QUINTÃO; LIMA, 2008). Em 1972, surgiu o primeiro videogame doméstico, o “Odyssey 100”, que vendeu mais de 100 mil cópias nos Estados Unidos no período de quatro meses (NESTERIUK, 2018).

O desenvolvimento dos *serious games* difere em alguns aspectos do desenvolvimento dos jogos digitais comerciais (MICHAEL; CHEN, 2005). O primeiro ponto de contraste está no design dos jogos: enquanto os jogos comerciais buscam aproveitar os hardwares de última geração disponíveis no mercado, os *serious games* precisam ser compatíveis com equipamentos que variam em hardware, processamento e funcionalidades disponíveis. Outro aspecto importante é que a disponibilidade de recursos (orçamentários, de tempo e de pessoas) para o desenvolvimento de *serious games* é consideravelmente inferior do que aquele disponibilizado por produtoras de jogos comerciais de sucesso. Essa limitação faz com que os responsáveis pelo desenvolvimento dos *serious games* concentrem seus esforços apenas nos elementos essenciais para o jogo, com reduzidas possibilidades de construção de universos e personagens mais complexos. Por fim, os *serious games* precisam ser integrados com o modelo de aprendizagem que apoiam e, por essa razão, requerem funcionalidades adicionais aos jogos comerciais, como o “modo observador” (para gravar o uso do programa pelo público-alvo) e versões diferentes, dependendo do papel do utilizador, como professores e alunos.

### 2.4 TRABALHO RELACIONADOS

*Serious Games* têm sido utilizados como uma ferramenta de apoio para a obtenção de habilidades em diferentes áreas, como negócios e tecnologia. Nos últimos anos, *serious games* foram desenvolvidos como ferramenta de apoio para o Ensino de Teste de Software.

"IslandTest" (QUEIROZ; PINTO; SILVA, 2019) é um *serious game* do gênero quebra-cabeça para apoiar o ensino-aprendizagem da disciplina de teste de software e suprir a lacuna das práticas de teste na graduação. Baseado na franquia de televisão "Lost", o jogador assume o papel de um programador que, após um acidente de avião, se vê preso em uma ilha misteriosa com seu notebook. Para conseguir escapar, ele precisa cumprir desafios relacionados aos tipos de teste (unidade, integração, validação e sistema) e processos de plano de teste que liberarão o computador para o personagem pedir ajuda.

Outro exemplo é o "iLearnTest" (RIBEIRO; PAIVA, 2015), *serious game* que auxilia no aprendizado dos conceitos de teste de software presentes na certificação ISTQB de nível fundamental (no Brasil, a "CTFL"). O software consiste em um compilado de *mini games* independentes entre si, que trazem desafios relacionados aos capítulos do *Syllabus*, o guia da certificação. O jogador não precisa seguir uma ordem para jogar, podendo escolher o tema que deseja no momento da seleção do jogo.

Já "O Jogo das Sete Falhas" (DINIZ; DAZZI, 2011) coloca o jogador no papel de um testador dentro de um sistema fictício que precisa ser testado. Sem qualquer conhecimento do código, ele deve encontrar bugs (ou as "7 falhas") na funcionalidade e classificá-los de acordo com os critérios de "valor limite" ou "partição de equivalência". Quando consegue resolver o desafio, o jogador avança para um nível superior (em grau de dificuldade), em que os erros são mais complexos de achar e de classificar.

Um quarto exemplo é o "gTestLearning" (SILVA et al., 2016), desenvolvido para ensinar conceitos básicos de teste de Software presentes na certificação CTFL. É inspirado na saga Mario World, com elementos de quiz e de RPG. O jogador deve responder às perguntas referentes ao tema de teste de software em turnos para derrotar os inimigos e ganhar pontos.

Finalmente, mas não menos importante, o "GameTest" (SILVA et al., 2022) é um protótipo de um *serious game* baseado na série de livros e filmes "Harry Potter". O artigo apresenta a proposta de desenvolvimento de jogo educacional no estilo "quiz" para auxiliar no processo de ensino-aprendizagem de teste de software, em particular os conceitos pertinentes aos testes unitários e testes funcionais.

### 3. METODOLOGIA

Dado o cenário apresentado, foi desenvolvido o serious game "Jogos Dourados" como ferramenta de apoio para o processo de ensino-aprendizagem de teste de software. Por meio do uso de elementos do gênero *visual novel*, "Jogos Dourados" possibilita que graduandos de cursos de tecnologia ou profissionais recém-formados na área possam simular a vivência em uma empresa de desenvolvimento de jogos enquanto agentes responsáveis pela garantia de qualidade. "Visual Novel" é um gênero de jogo digital focado na narrativa, sendo que o jogador deve ser capaz de impactar a progressão da história por meio da interação com os personagens e da tomada de decisões (CAMINGUE; CARSTENSDOTTIR; MELCER, 2021).

O jogo foi desenvolvido na estrutura WEB, com o uso de HTML5, CSS e Javascript. A escolha da linguagem em questão deu-se por sua flexibilidade e suporte para desenvolvimento de jogos digitais que podem ser experienciados diretamente no navegador, sem a necessidade de transferência dos arquivos para o computador ou instalação na máquina do usuário. Além disso, a linguagem ocupa a sétima posição (entre as linguagens mais utilizadas no ano de 2022) no *ranking* IEEE (IEEE SPECTRUM, 2022), indicando que trata-se de uma tecnologia atual, com integração com os recursos mais recentes e com suporte da comunidade.

Devido à versatilidade da linguagem, nos últimos anos foram desenvolvidos diversos *frameworks* para a construção de jogos em Javascript como o MelonJS, ImpactJS, PlayCanvas e Phaser. Dentre os *frameworks* disponíveis, optou-se pelo uso do Phaser em sua terceira versão. O Phaser é uma biblioteca gratuita e de código aberto que simplifica a criação de jogos 2D e 3D para navegadores WEB, permitindo o acesso de mais usuários sem a necessidade de aparelhos de última geração. Além disso, o Phaser possui uma documentação robusta e atualizações são publicadas com recorrência, mais um diferencial em relação aos demais *frameworks* disponíveis.

Para a publicação do jogo, foi criado um site, hospedado por meio da plataforma GitHub Pages. Essa abordagem se baseia na utilização eficiente da ferramenta de hospedagem, que permite a publicação de páginas estáticas diretamente de repositórios públicos no GitHub, sem qualquer custo. Isso proporciona uma solução eficaz tanto para a manutenção do código quanto para a disponibilização do jogo online.

No jogo, o jogador possui o papel de um estudante de tecnologia que assume a posição de estagiário de qualidade em uma *startup*. A empresa, aberta por alunos da faculdade deste, desenvolve jogos eletrônicos para escolas particulares utilizarem em sala de aula. Na equipe do jogador há duas pessoas: Pedro, responsável pelos aspectos do negócio, e Julia, encarregada do desenvolvimento da aplicação dos jogos. Além disso, uma terceira personagem, Otavio, desempenha o papel de um analista de qualidade que está saindo dessa empresa e que pretende treinar o jogador para que ele consiga desempenhar a sua função (Figura 1).

Figura 1. Tela de interação com a personagem Otavio



Fonte: Elaborada pela autora.

"Jogos Dourados" oferece aos jogadores a oportunidade de vivenciar a jornada de um profissional de qualidade ao longo de um ciclo simulado de três semanas, tempo comumente utilizado em equipes de desenvolvimento que atuam no modelo ágil (LI, 2010). Durante esse período, os jogadores podem explorar e assimilar conceitos fundamentais relacionados à função de um analista de qualidade e aos princípios do teste de software.

Ao longo de cada uma das semanas, um personagem assume o papel principal de guia, alinhado com o objetivo de aprendizagem definido para aquele período. O conceito em foco é então apresentado por meio de diálogos interativos e vídeos de aproximadamente 15 segundos. Após a apresentação do conceito, os jogadores são desafiados com mini games para aplicar o que aprenderam, conforme apresentado na Tabela 1:

Tabela 1. Estrutura dos conceitos apresentados no jogo

	Conceitos apresentados	Atividade de fixação	Personagem principal
<b>Semana 0</b>	Importância do papel do analista de qualidade, atividades desempenhadas pelo analista de qualidade	Jogo de verdadeiro ou falso, perguntas e respostas	Otávio
<b>Semana 1</b>	Testes estáticos, análise de requisitos funcionais, viés de confirmação	Perguntas e respostas, jogo de verdadeiro ou falso	Pedro
<b>Semana 2</b>	Testes dinâmicos, reporte de defeitos na aplicação	Simulação de um teste dinâmico na aplicação "jogo da memória", perguntas e respostas	Julia

Fonte: Elaborada pela autora.

A ordem do conteúdo, bem como o personagem que é definido como o principal guia de cada semana, relaciona-se diretamente com as atividades desempenhadas por um analista de qualidade durante sua atuação na entrega de uma funcionalidade. Primeiramente, o jogador conhece o projeto e adquire informações sobre a sua atuação dentro dele como estagiário em qualidade (Figura 2). Em seguida, os requisitos funcionais são apresentados e devem ser avaliados por meio da aplicação de técnicas de teste estáticas, representadas no jogo como uma "análise de requisitos funcionais" (Figura 3). Finalmente, o jogador tem a possibilidade de ver a aplicação em funcionamento e realizar um teste dinâmico, reportando, ao final, os defeitos que ele foi capaz de identificar.

Figura 2. Tela de Desafio



Fonte: Elaborada pela autora.

Ao final de cada desafio realizado, o jogador interage com o personagem guia, que apresenta uma reação determinada pelo seu desempenho na atividade anterior. Por exemplo, quando o jogador é capaz de cumprir o desafio sem cometer erros, o comportamento da

personagem guia reflete sentimentos como orgulho e felicidade. Se, por outro lado, o jogador não for capaz de passar pelo desafio sem cometer enganos, o personagem guia interagirá dizendo a solução esperada.

Figura 3. Simulação de uma análise de requisitos funcionais



Fonte: Elaborada pela autora.

Ao final do jogo, a soma dos pontos obtidos é avaliada para determinar o final que será apresentado ao jogador. Caso os pontos obtidos sejam maiores do que 90% do máximo possível, o jogador é parabenizado e recebe a proposta de ser efetivado, mesmo com experiência reduzida de trabalho na empresa. Se a soma estiver entre 60% e 89%, o jogador tem seu trabalho reconhecido por meio de uma carta escrita pelos colegas, afirmando que a qualidade das entregas tem melhorado e que estão felizes por tê-lo ali. Por outro lado, se o desempenho final estiver abaixo de 60%, o antigo analista de qualidade da equipe retorna e o jogador é convidado a aprender mais sobre a função para que possa ter uma oportunidade futuramente ali.

Uma vez realizada a implementação do jogo apresentado, foi selecionada uma amostra de usuários reais para a realização de uma prova de conceito. Os usuários foram captados por meio de formulários divulgados em grupos on-line da Faculdade de Computação e Informática do Mackenzie (FCI). Foram selecionados alunos e ex-alunos de cursos de tecnologia, com o critério de exclusão de que não poderiam ter atuado em algum momento como analistas de qualidade (ou exercido funções similares), uma vez que o conteúdo apresentado no jogo é considerado introdutório ao exercício da função. Dessa forma, foram selecionados cinco usuários, conforme a abordagem discutida por Bevan *et al.* (2003), apresentados na Tabela 2.

Tabela 2. Perfil dos participantes da pesquisa

	Idade	Sexo	Grau de instrução	Cargo atual
Indivíduo A	24	Feminino	Superior completo	Analista de <i>Business Intelligence</i>
Indivíduo B	26	Feminino	Superior completo	Analista de desenvolvimento <i>fullstack</i>
Indivíduo C	25	Feminino	Superior completo	Analista de desenvolvimento <i>back-end</i>
Indivíduo D	24	Masculino	Superior completo	Analista de desenvolvimento <i>back-end</i>
Indivíduo E	27	Masculino	Superior incompleto	Estagiário em desenvolvimento <i>back-end</i>

Fonte: Elaborada pela autora.

Uma vez selecionados os usuários, os testes foram realizados em três etapas distintas. Na primeira etapa, os indivíduos responderam a um questionário com perguntas abertas e fechadas a respeito dos temas apresentados no jogo. Na segunda etapa, os usuários foram convidados a entrar em uma sala de conferência virtual individualmente e compartilhar suas telas enquanto jogavam o jogo, a fim de compartilhar impressões ou dúvidas. Finalmente, após terem jogado o jogo, os usuários responderam a um questionário de avaliação, com as mesmas perguntas presentes no primeiro questionário, acrescido de duas perguntas abertas a respeito da avaliação da passagem de conhecimento presente na aplicação. Uma última questão pedia que os usuários atribuíssem uma nota sobre a experiência com o jogo no geral.

No que diz respeito à primeira etapa, os resultados para a questão fechada “Você sabe a diferença entre o uso de técnicas estáticas e dinâmicas para testar uma aplicação” mostraram que três dos cinco dos respondentes não sabiam do que se tratava ou não eram capazes de diferenciar técnicas de teste estáticas e dinâmicas. Por outro lado, quando questionados sobre o conhecimento a respeito do tema “viés de confirmação” (por meio da pergunta “Você sabe o que é viés de confirmação?”), quatro dos cinco respondentes afirmaram saber do que se tratava.

Em relação às perguntas abertas, os participantes foram solicitados a definir a atividade “teste de software”, mencionar três atividades desempenhadas por um analista de qualidade e discutir a importância desse profissional no processo de desenvolvimento de aplicações. No entanto, foi observado que houve uma interpretação divergente da primeira pergunta. Alguns participantes entenderam erroneamente que a pergunta buscava saber sua opinião pessoal em relação à atividade, se eles tinham ou não apreço por ela. Esse resultado sugere que a clareza das questões é fundamental para evitar interpretações variadas e garantir a coleta precisa dos dados desejados. Por essa razão, essa pergunta foi retirada do questionário seguinte e desconsiderada para os fins desse trabalho.

Em relação à segunda pergunta aberta, três indivíduos conseguiram elencar corretamente as atividades, enquanto dois foram incapazes de fazê-lo, limitando suas respostas a apenas duas ou nenhuma função. Por exemplo, o "Indivíduo A" afirmou que a função dos analistas de qualidade é "Testar, apontar erros", e o "Indivíduo E" afirmou apenas não saber. Por outro lado, as respostas da última pergunta indicaram que a maioria dos participantes já possuía uma compreensão sólida da importância do papel de um analista de qualidade, mesmo antes de fazerem uso da aplicação.

Após a conclusão do questionário, os participantes foram convidados a participar de uma sessão de chamada de vídeo, durante a qual compartilharam suas telas e interagiram com o sistema em análise. Foi nesse momento que tiveram seu primeiro acesso ao jogo. Durante as sessões de interação, os participantes receberam instruções para jogar o "Jogos Dourados" e foram encorajados a fazer perguntas sempre que necessário. Notavelmente, os cinco jogadores (totalidade dos participantes) conseguiram navegar pelas telas do jogo com sucesso, sem necessitar de auxílio direto do interlocutor presente na chamada. As perguntas levantadas pelos participantes, predominantemente, focaram-se na tecnologia subjacente à construção da aplicação e nas possibilidades de expandir o jogo para abranger outros temas.

#### **4. RESULTADO E DISCUSSÃO**

Após o término das sessões descritas anteriormente, os participantes da prova de conceito responderam a um questionário de avaliação de conhecimento sobre o tema e a aplicação testada. Os resultados das perguntas fechadas podem ser observados a seguir:

- Para a questão "Você sabe a diferença entre o uso de técnicas estáticas e dinâmicas para testar uma aplicação?" a totalidade das pessoas que participaram deste teste inicial de validação da proposta (cinco pessoas) indicaram a resposta "Sei e me sinto confortável para explicar para outras pessoas."
- Em relação à pergunta "Você sabe o que é viés de confirmação?", três pessoas indicaram a resposta "Sei e me sinto confortável para explicar para outras pessoas", sendo que, uma delas selecionou "Sei mas não conseguiria explicar para outras pessoas" e a outra escolheu "Já ouvi falar mas não sei bem o que é".

Os resultados evidenciam, portanto, que os participantes deste primeiro teste adquiriram conhecimento e se sentiram mais confiantes para abordar os assuntos tratados no jogo.

Em relação às perguntas abertas, nas quais os participantes foram solicitados a citar três atividades desempenhadas por um analista de qualidade, quatro entre cinco participantes conseguiram responder corretamente à questão. O participante "A" conseguiu citar duas atividades, "reportar erros" e "testar a aplicação", o que mostra uma melhora na percepção comparado com a resposta anterior, onde apenas apontou apenas "testar" como uma atividade desse profissional

Por outro lado, as respostas à pergunta "Qual a importância de um analista de qualidade no processo de desenvolvimento de software?" mostram uma notável falta de variação em comparação com o questionário anteriormente administrado, na qual todos os participantes demonstraram conhecer a importância da função. Essa uniformidade nas respostas pode ser atribuída à clareza previamente estabelecida, que não sofreu alterações devido ao conteúdo do jogo, uma vez que este confirmou o conhecimento já existente entre os respondentes.

As avaliações sobre o jogo foram positivas. Os participantes destacaram que o jogo foi eficaz em fornecer explicações abrangentes sobre o teste de software e elogiaram a abordagem educacional adotada. A interação com os personagens e a participação em casos de teste transformados em minijogos também foram identificados como fatores motivadores. Não foram identificados aspectos considerados de menor interesse, sugerindo que a experiência geral do jogo foi adequadamente recebida.

Por fim, em relação à avaliação geral da experiência com o *serious game*, quatro dos cinco participantes atribuíram a nota máxima (5) e um participante atribuiu a nota 4 devido à presença de alguns *bugs* na aplicação: alguns objetos não eram excluídos da tela no momento correto, e apesar de estarem em segundo plano, continuavam clicáveis. Quando acidentalmente algum participante clicava na região onde estava esse objeto, havia um redirecionamento do usuário para uma tela de diálogo que já havia sido apresentada anteriormente, impossibilitando um avanço no jogo. Outro momento de falha ocorria quando o jogador não aguardava o carregamento completo dos personagens, clicando nas caixas de diálogo antecipadamente. Após alguns cliques, o jogo apresentava falha e era necessário reiniciá-lo. Apesar disso, média das notas ficou em 4.8, muito próxima da nota máxima.

As conclusões a respeito do jogo, assim como os trabalhos futuros propostos, são apresentados na seção a seguir.

## 5. CONSIDERAÇÕES FINAIS

Neste estudo foi delineado o desenvolvimento e a avaliação de um *serious game* voltado para o ensino dos fundamentos de teste de software a estudantes e profissionais da área de tecnologia da informação. Por meio da análise comparativa do conhecimento dos participantes antes e após a interação com o sistema, foi possível inferir que a aplicação atingiu seu objetivo de ampliar a compreensão desses conceitos e fomentar o interesse na adoção de princípios de qualidade de software.

Contudo, é necessário ressaltar três limitações do estudo que devem ser consideradas na análise dos resultados apresentados. A primeira delas é a representatividade da amostra: a seleção realizada apenas entre alunos formados e recém-formados em cursos de tecnologia pela Universidade Presbiteriana Mackenzie pode conter vieses devido à sua homogeneidade. Dado que o currículo acadêmico adotado pela Universidade possui diferenças pontuais entre os alunos selecionados, é esperado que o conhecimento no momento da aplicação do questionário seja próximo entre os participantes, assim como os resultados obtidos após o uso da aplicação.

Ainda em relação à amostra utilizada, a segunda limitação diz respeito à quantidade de participantes da pesquisa. Apesar do número de participantes ter sido considerado suficiente no momento da realização dos testes, a partir da abordagem adotada (BEVAN et al., 2003), avaliações posteriores do trabalho sugeriram que o uso de uma amostra quantitativamente mais significativa de usuários, assim como uma extensão no tempo de contato do usuário com o jogo e o uso de grupos de controle, poderia minimizar os possíveis impactos que a seleção utilizada possa ter tido no resultado.

Finalmente, o estudo apresentado não permite avaliar a aquisição de conhecimento prático na disciplina de teste de software, limitando-se apenas a complementar a formação teórica no tema. Dessa forma, não é possível avaliar a aplicação dos conhecimentos obtidos e o impacto deles no dia a dia dos participantes após o uso do *serious game*.

Posto isso, como proposta de trabalho futuro, propõe-se a seleção/modelagem de uma amostragem com maior grau de heterogeneidade de usuários para os testes, assim como uma amostra de usuários representativa. A adoção de uma metodologia que possibilite um contato do usuário com a aplicação por um período maior de tempo vai possibilitar a obtenção de resultados mais fidedignos. Em relação à aplicação, a expansão do conteúdo teórico apresentado, assim como a introdução de atividades práticas é sugerida de forma a ampliar o

escopo do jogo e aproximar os usuários de cenários que são vivenciados dentro das empresas de tecnologia.

Adicionalmente, se buscará, ainda, a introdução de um sistema de conquistas no jogo, com o objetivo de estimular a rejogabilidade e o engajamento contínuo dos usuários. Essas conquistas podem estar relacionadas à aquisição de conhecimentos específicos, ao desenvolvimento das relações com os personagens e à evolução do jogador dentro do jogo. Ao oferecer recompensas tangíveis por meio de conquistas, acredita-se que é possível fomentar um ciclo de aprendizado mais dinâmico, ao mesmo tempo em que os participantes são motivados a explorar diferentes trajetórias dentro do jogo.

Em resumo, a investigação mostra que a utilização de um *serious game* no contexto de ensino de teste de software apresenta potencialidade para expandir o domínio dos conceitos fundamentais e fomentar a adoção de abordagens de qualidade. A crescente adesão a essas abordagens inovadoras pode desempenhar um papel relevante na formação de profissionais aptos a enfrentar os desafios, em constante evolução, no campo da tecnologia da informação, mesmo aqueles que nunca desempenharam diretamente a posição apresentada no jogo.

## 6. REFERÊNCIAS

- BARBOSA, E. F. et al. *Introdução ao Teste de Software*. João Pessoa, Brasil: [s.n.],2000.
- BATISTA, M. L. S.; QUINTÃO, P. L.; LIMA, S. M. B. Um estudo sobre a influência dos jogos eletrônicos sobre os usuários. *Revista Eletrônica da Faculdade Metodista*, n. 4, 2008.
- BENITTI, F. B. V.; ALBANO, E. L. Teste de software: o que é e como é ensinado? In: *CSBC 2012 - XXXII Congresso da Sociedade Brasileira de Computação*. Curitiba, Brasil, 2012. p. 1–10.
- BESSA, T.; PESSOA, M.; LOBÃO, L. Análise das atividades de uma equipe ágil em uma sprint para integrar qualidade no desenvolvimento de software. In: *Anais do VII Encontro Regional de Engenharia (ENCOENG 2019)*. Manaus, 2019.
- BEVAN, N. et al. The "magic number 5" is it enough for web testing?. In: *CHI'03 extended abstracts on Human factors in computing systems*. 2003. pp. 698-699.
- BOARD, B. S. T. Q. CTFL 4.0. 2023. <https://bstqb.org.br/b9/ctfl>.
- CAMINGUE, J.; CARSTENSDOTTIR, E.; MELCER, E. F. What is a visual novel? In: *Proc. ACM Hum.-Comput. Interact.*, v. 5, n. CHI PLAY, 2021.
- CANTOR, J. A. Skills certifications and workforce development: Partnering with industry and ourselves. *Leadership Abstracts*, v. 15, n. 1, p. 1–6, 2002.

DINIZ, L. L.; DAZZI, R. L. S. *Jogo das Sete Falhas: Um jogo educacional para apoio ao ensino do teste caixa preta*. Florianópolis, Brasil: [s.n.], 2011.

GALLINDO, S. P.; OLIVEIRA, M. *Demanda de Talentos em TIC e estratégia TCEM*. São Paulo, dez. 2021.

GAROUSI, V. et al. Software-testing education: A systematic literature mapping. *Journal of Systems and Software*, v. 165, p. 1-49, 2020. HUIZINGA, J. *Homo Ludens: o Jogo como Elemento na Cultura (1938)*. São Paulo: Perspectiva, 2008.

LI, J., "Agile Software Development," 2010.

MICHAEL, D. R.; CHEN, S. L. *Serious Games: Games That Educate, Train, and Inform*. Muska Lipman/Premier-Trade, 2005. MYERS, G. G.; SANDLER, C.; BADGETT, T. *The Art of Software Testing*. 3. ed. New Jersey: Wiley, 2012.

NESTERIUK, S. Games e educação: possibilidades e desafios. *Revista Páginas Abertas (Paulus)*, ano 38, n. 55. P. 30-31, 2018.

PASCHOAL, L. N.; SOUZA, S. de Rocio Senger de. A survey on software testing education in brazil. In: *SBQS: Proceedings of the 17th Brazilian Symposium on Software Quality*. Curitiba, Brasil, 2018. p. 334–343.

PRESSMAN, R. S. *Software Engineering: A practitioner's approach*. 7. ed. New York: McGrawHill, 2010.

QUEIROZ, R. J. de; PINTO, F. de S.; SILVA, P. C. da. *IslandTest: jogo educativo para apoiar o processo de ensino-aprendizagem de testes de software*. Salvador, Brasil, 2019.

RANDALL, M. H.; ZIRKLE, C. J. Information technology student-based certification in formal education settings: who benefits and what is needed. *Journal of Information Technology Education: Research*, v. 4, n. 1, p. 287–306, 2005. RIBEIRO, T. P. B.; PAIVA, A. C. R. *iLearnTest: Educational game for learning software testing*. Aveiro, Portugal, 2015.

SILVA, C. G. et al. *gTest Learning: Um Jogo para Ensino Básico de Teste de Software*. Natal, Brasil, 2016.

SILVA, M. A. F. da et al. Gametest: Um protótipo de jogo para apoiar o ensino-aprendizagem de teste de software. *Conjecturas: especial novos estudos*, v. 22, n. 6, p. 679–706, 2022.

SOMMERVILLE, I. *Engenharia de Software*. 9. ed. São Paulo: Pearson, 2011.

VALENTE, M. T. *Engenharia de Software Moderna: Princípios e práticas para desenvolvimento de software com produtividade*. Belo Horizonte: Independente, 2022.

VALLE, P. H.; BARBOSA, E.; MALDONADO, J. *Um mapeamento sistemático sobre ensino de teste de software*. 2015.