

GAMETEST: UM PROTÓTIPO DE JOGO PARA APOIAR O ENSINO-APRENDIZAGEM DE TESTE DE SOFTWARE

GAMETEST: A GAME PROTOTYPE TO SUPPORT SOFTWARE TESTING TEACHING-LEARNINGS

Maria Adriana Ferreira da Silva¹

Alana Gabrielly Pereira Pinheiro²

Pedro Henrique Aires da Silva³

Alysson Filgueira Milanez⁴

RESUMO

O Teste de Software (TS) é um processo que consiste em tentar descobrir problemas em softwares antes que estes sejam entregues aos clientes. Desta forma, o teste é um processo indispensável no âmbito do desenvolvimento de software e influencia diretamente na sua qualidade. Apesar de sua importância, eventualmente os estudantes apresentam dificuldades no tocante ao aprendizado teórico e prático nas disciplinas de Teste de Software. Diante deste contexto, o presente trabalho tem como objetivo apresentar o desenvolvimento de um protótipo de um jogo educacional intitulado GameTest que tem a finalidade de auxiliar no processo de ensino-aprendizagem de Teste. O protótipo foi validado a partir de uma pesquisa realizada com 40 pessoas. A pesquisa foi baseada na ISO 9241 e buscou medir a usabilidade em termos de desempenho e satisfação do usuário. Como resultados, obteve-se respostas positivas quanto à importância deste trabalho para a área de Teste.

Palavras-chave: Jogo. Protótipo. Teste de Software

ABSTRACT

Software Testing (ST) is a process that consists of trying to discover problems in software before it is delivered to customers. Thus, it is an indispensable process within the scope of software development and directly influencing its quality. Despite its importance, eventually students have difficulties in terms of theoretical and practical learning in Software Testing disciplines. In this context, the present work aims to present the development of a prototype of an educational game called GameTest with the purpose of assisting in the teaching-learning process of software testing. The prototype was validated by means of a survey carried out with 40 people. The survey was based on ISO 9241 and sought to measure usability by means of performance and user satisfaction. As results, we obtained positive responses regarding the importance of the work for the test area.

Keywords: Game. Prototype. Software Testing

Disponibilidade: <https://conjecturas.org/index.php/edicoes/article/view/1104/840>

1 INTRODUÇÃO

¹ Discente - Graduada em Tecnologia da Informação pela Universidade Federal Rural do Semi-Árido.

² Discente - Graduada em Tecnologia da Informação pela Universidade Federal Rural do Semi-Árido.

³ Discente - Graduado em Tecnologia da Informação pela Universidade Federal Rural do Semi-Árido.

⁴ Docente - Doutor em Ciência da Computação pela Universidade Federal de Campina Grande.

Teste de software é um processo que visa encontrar problemas para aumentar a confiança de um produto antes que seja entregue ao cliente, a fim de checar se o comportamento está conforme o especificado (QUEIROZ; PINTO; SILVA, 2019). Seu uso busca assegurar maior qualidade no software que está sendo desenvolvido e reduzir custos (DELAMARO; JINO; MALDONADO, 2013). A qualidade de um software pode estar fortemente relacionada à existência de defeitos adquiridos durante sua fase de desenvolvimento, e uma das maneiras de auxiliar a encontrar tais defeitos é por meio do teste de software (PONTES, 2009; DELAMARO; JINO; MALDONADO, 2013).

No mercado, inicialmente, tais atividades eram realizadas por desenvolvedores; no entanto, não existiam ferramentas dedicadas apenas a essas atividades o que contribuía para que os testes fossem realizados quase no fim do desenvolvimento do software causando inúmeros problemas (BENITTI; ALBANO, 2012).

Assim, para que o mercado de desenvolvimento de software implementa melhores práticas nas atividades de teste de software, visando a sua importância na garantia de qualidade, é preciso que os cursos de Computação dêem maior ênfase à disciplina de Teste de Software (BENITTI; ALBANO, 2012). Entretanto, na maioria dos cursos de Computação, o conteúdo de Teste é abordado dentro da disciplina de Engenharia de Software sendo apresentado aos alunos no formato de aulas expositivas, o que contribui para falta de concentração e desinteresse por parte dos alunos o que pode influenciar na desistência da disciplina (QUEIROZ; PINTO; SILVA, 2019; WANGENHEIM; WANGENHEIM, 2012).

Para o cenário do ensino de Teste de Software, uma das maiores dificuldades está na necessidade da aplicação da parte prática, pois muitas vezes tem-se apenas o aprendizado teórico (COUTINHO; BEZERRA, 2018). Na literatura, diversas soluções vêm sendo propostas para melhorar o ensino de teste, como, por exemplo, o trabalho de Coutinho e Bezerra (2018) que apresenta uma avaliação de um jogo para o ensino de Teste levando em consideração o método de avaliação de softwares educativos e o ponto de vista de alunos que cursaram a disciplina de Engenharia de Software. No entanto, a maioria dessas soluções estão relacionadas a aspectos conceituais, o que colabora para uma deficiência em ferramentas e materiais de apoio para a parte prática da disciplina (FRANCISCO et al. 2018).

Os jogos educacionais podem auxiliar na parte prática da disciplina de forma divertida e motivadora, auxiliando no processo de ensino aprendizagem dos conteúdos fundamentais para o entendimento de testes de software (QUEIROZ; PINTO; SILVA, 2019). Jogos têm sido utilizados para ajudar no ensino de diversas áreas do conhecimento, quando utilizados como ferramentas educacionais, permitindo a experimentação de diversas situações, entre elas as ligadas ao ambiente profissional (FARIAS et al., 2012).

Neste contexto, este trabalho visa o desenvolvimento de um protótipo de jogo educacional para auxiliar na parte teórica e prática do ensino-aprendizagem de Teste de Software. O protótipo foi validado a partir de uma pesquisa realizada com 40 pessoas. A pesquisa foi baseada na ISO 9241 e buscou medir a usabilidade em termos de desempenho e satisfação do usuário. Como resultados, obteve-se respostas positivas quanto à importância do presente trabalho para a área de Teste. Além disso, ele oferece as funções necessárias para a realização de forma eficiente das tarefas relacionadas ao teste de software de modo que pode auxiliar e contribuir para a parte prática da disciplina.

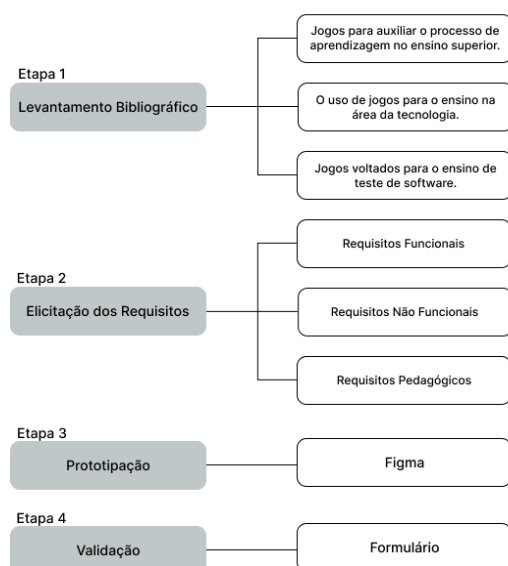
O restante do trabalho é organizado como segue. A seção 2 apresenta a metodologia utilizada. A seção 3 descreve a fundamentação teórica. Na seção 4, são abordados os trabalhos relacionados com a temática deste trabalho. A seção 5 descreve o desenvolvimento do protótipo proposto. A seção 6, expõe os resultados e discussões da avaliação do protótipo. Por fim, a seção 7 apresenta as considerações finais e prospectos para trabalhos futuros.

2 METODOLOGIA

A metodologia é um método estudado para buscar determinado conhecimento, a sua importância é complementada pela diversidade dos métodos que procuram atender às necessidades da pesquisa (ARAGÃO; NETA, 2017). Dessa forma, na Figura 1, é apresentada a metodologia de desenvolvimento do trabalho. As etapas são definidas a seguir.

- **Levantamento Bibliográfico:** Galvão (2010) afirma que realizar um levantamento bibliográfico é potencializar intelectualmente com o conhecimento coletivo, a fim de conhecer novos recursos necessários para a construção de um estudo específico. O levantamento bibliográfico contemplou pesquisas que englobam o uso de jogos voltados para o ensino e aprendizagem, não apenas de tecnologias, mas também, de testes de software.
- **Elicitação de Requisitos:** Os requisitos de um software são descrições dos recursos que o software pode ter, estes são compostos por requisitos funcionais e não funcionais. Os funcionais descrevem o comportamento de um software em relação a tarefas e serviços da perspectiva do seu usuário em específico, já os não funcionais são restrições de ordem geral aos requisitos funcionais (VAZQUEZ; SIMÕES, 2016; PRESSMAN; MAXIM, 2016). Por meio da análise de jogos similares e do conhecimento do contexto geral de jogos educacionais, os requisitos do jogo para aplicação no protótipo foram licitados.
- **Prototipação:** A etapa de prototipação auxilia os envolvidos a compreender o que está para ser construído de acordo com os requisitos (PRESSMAN; MAXIM, 2016). Na etapa de prototipação, a ferramenta Figma⁵ foi utilizada para gerar o protótipo e expor de forma gráfica como o jogo será desenvolvido.
- **Validação:** A última etapa realizada refere-se à validação do protótipo com um formulário aplicado ao público. Esse formulário buscou coletar dados sobre a usabilidade do protótipo e possíveis sugestões de melhorias ao mesmo.

Figura 1 – Metodologia



Fonte: AUTORES (2022)

3 FUNDAMENTAÇÃO TEÓRICA

⁵ Disponível em: <https://www.figma.com/>. Acesso em 29 mai. 2022.

Nos últimos 50 anos, os softwares estiveram em contínua evolução, saindo de uma ferramenta especializada em análise de informações para uma indústria em si (PRESSMAN; MAXIM, 2016). Contudo, existem problemas para desenvolver softwares de qualidade. Nesse sentido, a Engenharia de Software surgiu para organizar o software bem como oferecer um melhor gerenciamento de seus processos enquanto produto (REZENDE, 2006).

A Engenharia de Software é vista como uma metodologia de desenvolvimento e manutenção de sistemas modulares (REZENDE, 2006). Engloba processos, métodos e ferramentas que possibilitam a construção de sistemas digitais dentro do prazo e com qualidade (PRESSMAN; MAXIM, 2016). O teste de software é uma das atividades que procuram contribuir para a melhoria da qualidade dos sistemas desenvolvidos em geral (DELAMARO; JINO; MALDONADO, 2013). É a adoção de práticas de teste que garante que um determinado sistema esteja sendo desenvolvido de acordo com os requisitos especificados (SALOMÃO, 2016).

Visando a adoção de melhores práticas para auxiliar os alunos da área de Computação a praticar e estudar conceitos relacionados a Teste, surge a necessidade de realizar um levantamento bibliográfico acerca dos conteúdos: construção de jogos para auxiliar o processo de aprendizagem no ensino superior e o uso de jogos para ensino na área da tecnologia.

3.1 Jogos para auxiliar o processo de aprendizagem no ensino superior

Segundo Lozza e Rinaldi (2017), o uso dos jogos em sala de aula deve ser marcado por uma relação de aprendizagem, e deve ser utilizado como uma ferramenta de apoio ao aprendizado para que o aluno e o professor possam ter uma experiência prazerosa de construção do conhecimento.

Lozza e Rinaldi (2017) ainda destacam os cuidados necessários ao incluir os jogos no ambiente da sala de aula, pois estes precisam ser previamente testados, devem possuir uma síntese do conteúdo do jogo; também deverá haver regras no jogo, dispor de elaboração de atividades pedagógicas e, por fim, pontuação para gerar motivação.

3.2 Uso de jogos para ensino na área da tecnologia

Para Ribeiro, Timm e Zaro (2006), o lúdico é uma grande ferramenta no auxílio da educação, e os jogos digitais são capazes de permitir momentos ricos de aprendizagem. Ainda segundo os autores, ao combinar diversão e ambiente virtual, é possível criar ferramentas de narrativas educacionais extremamente poderosas e tais ferramentas são capazes de potencializar a capacidade de ensino-aprendizado.

3.3 Uso de jogos para ensino na área da tecnologia

Para Ribeiro, Timm e Zaro (2006), o lúdico é uma grande ferramenta no auxílio da educação, e os jogos digitais são capazes de permitir momentos ricos de aprendizagem. Ainda segundo os autores, ao combinar diversão e ambiente virtual, é possível criar ferramentas de narrativas educacionais extremamente poderosas e tais ferramentas são capazes de potencializar a capacidade de ensino-aprendizado.

4 TRABALHOS RELACIONADOS

Na literatura, alguns jogos vêm sendo desenvolvidos procurando melhorar o ensino e aprendizagem em cursos da área de Computação, com o objetivo de contribuir para a melhoria do ensino (MOREIRA; MONTEIRO, 2018). No entanto, são identificados poucos

jogos educativos que focam no ensino da parte prática da disciplina de teste de software (SILVA; MÜLLER; BERNARDI, 2011). A seguir, descreve-se alguns deles junto a uma análise e comparativo com a abordagem proposta neste trabalho.

O Islandtest (QUEIROZ; PINTO; SILVA, 2019) é um jogo sério para apoiar o ensino de Teste de Software. O seu enredo foi inspirado no seriado americano *Lost*, de J. J. Abrams. Os assuntos abordados foram: as fases de um processo de teste, tipos de teste, conceitos de defeito, erro, falha, depuração e casos de testes. Para a validação, foram aplicados questionários para avaliar o jogo com questões que abordavam desde a utilização e funcionamento do jogo até sugestões de melhorias. Os resultados do estudo possibilitaram identificar uma diferença na aprendizagem dos conteúdos antes e depois da utilização do jogo.

O gTest Learning (SILVA et al., 2016) é um jogo sério educacional para ensino de teste de software, inspirado na certificação Foundation do ISTQB (International Software Testing Qualifications Board). É um jogo para apenas um jogador, no qual o jogador deve derrotar inimigos através de pequenos questionários em fase de turno, adicionando uma técnica de diálogo usada em jogos de RPG. O nível de dificuldade é balanceado através dos inimigos, sendo quatro inimigos por estágio, três perguntas para os três primeiros; no caso do quarto inimigo ele é considerado o BOSS (o chefe, aquele que apresenta maior dificuldade) do estágio, havendo só uma pergunta para enfrentá-lo.

O iLearnTest (RIBEIRO, 2014) é uma proposta para complementar o ensino de teste de software e busca aumentar o interesse dos estudantes sobre o tema. Além disso, o iLearnTest tem como objetivo ajudar os estudantes na preparação para a realização do exame de certificação Base do ISTQB. A validação do jogo consistiu em propor uma experiência com alunos universitários, com o objetivo de analisar se os resultados obtidos pelos alunos que usaram o jogo na aprendizagem eram melhores que os resultados obtidos pelos alunos que não o usaram. Os resultados mostram que o iLearnTest apresentou um bom resultado e que poderá contribuir como um auxílio ao ensino tradicional.

O Jogo das 7 Falhas (DINIZ; DAZZI, 2011) é um jogo sério que tem o intuito de auxiliar estudiosos de teste de software a descobrir sete falhas existentes nas funcionalidades testadas, tendo como objeto realizar testes de caixa preta. O jogo apresenta dois níveis, o fácil e o médio, ambos tendo o mesmo funcionamento e o mesmo tipo de fases, sendo diferente apenas a complexidade das fases. Para jogar, são apresentados problemas ao jogador, em que é necessário que sejam apontadas as sete falhas presentes na fase. Para validar o jogo, foi realizada uma análise quantitativa com 21 alunos do curso de Ciência da Computação da UNIVALI, em que nesta análise foram obtidos feedbacks positivos sobre a ferramenta, tendo uma boa aceitação dos usuários, o que mostra ser um jogo de interesse para o público alvo.

O iTest Learning (FARIAS et al., 2012) apresenta um jogo sério que busca auxiliar alunos da Computação no planejamento de testes de software. Assim, estimulando o aluno a compreender a fase inicial dos testes, partindo de simulações de problemas para possíveis problemas em sistemas. O jogo funciona da seguinte forma: o jogador escolhe um dos três níveis de dificuldade, fácil, médio ou difícil, e inicia o jogo. É necessário avançar sobre as seis fases do jogo. A primeira é definir os itens de teste, a segunda fase é voltada para definir os tipos de testes a serem realizados. A terceira fase define os níveis de teste. Na quarta etapa, são definidos os critérios de aceitação. A quinta fase define as ferramentas a serem utilizadas. Na sexta e última fase, são definidos os artefatos. Ao final do jogo é obtido um feedback com todo o planejamento dos testes a serem realizados nos sistemas.

4.1 Análise

O Quadro 1 apresenta um comparativo entre os trabalhos encontrados na literatura versus o GameTest.

Quadro 1 - Comparação entre jogos encontrados na literatura

	Suporte de jogadores	Baseado na certificação <i>Foundation</i> do ISTQB	Intuitivo	Bom retorno de aprendizagem
Islandtest	<i>single-player</i>	Não	Sim	Sim
gTest Learning	<i>single-player</i>	Sim	Pouco	Sim
iLearnTest	<i>single-player</i>	Sim	Sim	Sim
Jogo das 7 Falhas	<i>single-player</i>	Não	Pouco	Sim
iTest Learning	<i>single-player</i>	Não	Sim	Sim
GameTest	<i>single-player</i>	Não	Sim	Sim

Fonte: AUTORES (2022)

Todos os jogos encontrados na literatura apresentam uma contribuição interessante para auxiliar na parte prática do ensino de teste. Quanto mais ferramentas para o ensino de teste e quanto mais estas apresentarem uma boa experiência de uso, melhores serão os resultados obtidos. Mediante a isso, em uma perspectiva semelhante a dos estudos citados, este trabalho visa desenvolver um protótipo de um jogo educativo voltado para apoio e suporte ao ensino-aprendizagem de teste de software segundo uma abordagem lúdica.

5 DESENVOLVIMENTO DO PROTÓTIPO

Nesta seção, apresenta-se a definição e o desenvolvimento do protótipo de um jogo educacional voltado para o ensino-aprendizagem de teste de software.

5.1 Definição do protótipo

Para a criação do protótipo, utilizou-se a ferramenta Figma, um editor online de prototipagem para projetos de *design*. Antes de dar início ao desenvolvimento do protótipo, foi necessário especificar os usuários/*stakeholders* do mesmo. Assim, após algumas pesquisas, definiu-se que seria qualquer pessoa interessada em aprender sobre teste de software, com ênfase em alunos da área de Computação.

Para isso, definiu-se requisitos funcionais, não-funcionais e requisitos pedagógicos. Os requisitos funcionais apresentam o comportamento que o software deve ter em termos das tarefas ou serviços do usuário, os não-funcionais descrevem as limitações gerais dos requisitos funcionais, complementando a especificação do software e possibilitando um melhor funcionamento do mesmo (SOMMERVILLE, 2011; VAZQUEZ; SIMÕES, 2016). Já os requisitos pedagógicos, são fundamentados em teorias e/ou métodos de aprendizagem, buscando auxiliar o desenvolvimento de softwares educacionais (HENRIQUE, 2016).

A seguir, são apresentados os requisitos funcionais, não-funcionais e os requisitos pedagógicos propostos para o protótipo do GameTest.

5.2 Requisitos Funcionais

Os requisitos funcionais e suas descrições podem ser encontrados no Quadro 2.

Quadro 2 - Requisitos Funcionais

Título	Descrição
RF01 - Realizar Cadastro	O usuário poderá cadastrar-se no sistema informando um apelido, e-mail e senha.
RF02 - Realizar <i>Login</i>	O usuário poderá realizar <i>login</i> no sistema.

RF03 - Recuperar Senha	O usuário poderá recuperar sua senha sempre que esquecer ou desejar mudá-la sem entrar no sistema.
RF04 - Editar Perfil	O usuário poderá editar suas informações de perfil, tais como: apelido, e-mail e senha.
RF05 - Realizar <i>Logout</i>	O usuário poderá realizar <i>logout</i> no sistema.
RF06 - Apresentar a história	Apresentar a história
RF07 - Jogar Partida	O usuário poderá resolver os desafios sobre testes de software e receber uma pontuação em formato de estrelas de acordo com o tempo para a resolução de cada desafio proposto.
RF08 - Salvar desempenho	O usuário poderá salvar seu desempenho no jogo.
RF09 - Visualizar Desempenho	O usuário poderá visualizar seu desempenho para cada nível do jogo.

Fonte: AUTORES (2022)

5.3 Requisitos Não-Funcionais

Os requisitos não-funcionais (RNF) e suas descrições podem ser vistos nos quadro 3.

Quadro 3 - Requisitos Não-Funcionais

Título	Descrição
RNF01 - Usabilidade - Estética de interface com o usuário.	O sistema deve ser atrativo para o usuário, de modo que ofereça elementos que facilitem a sua interação (ícones, cores, botões e etc.).
RNF02 - Usabilidade – Operabilidade	O sistema deve conter uma interface interativa, ou seja, de fácil uso para o usuário. Apresentando cores interessantes e ícones intuitivos.
RNF03 - Confidencialidade	O sistema deve permitir que o usuário salve seu progresso.
RNF04 - Integridade	O sistema deverá assegurar que as informações não sejam alteradas e acessadas por terceiros.

Fonte: AUTORES (2022)

5.4 Requisitos Pedagógicos

Os requisitos pedagógicos (RP) com suas descrições estão disponíveis no Quadro 4.

Quadro 4 - Requisitos Pedagógicos.

Título	Descrição
RP01 - Estrutura cognitiva	Os conteúdos abordados devem ser planejados de forma hierárquica e sequencial.
RP02 - Curto Prazo	O sistema deve propor problemas com um prazo curto de resolução.
RP03 - Níveis cognitivos	Os conteúdos abordados devem ser divididos em níveis de conhecimento, iniciando sempre do menor nível de complexidade.
RP04 - Autonomia	O sistema deve permitir que o usuário realize escolhas durante a interação.

RP05 - Exercício e prática	Haverá exercícios de habilidades ou conteúdo que o aluno já possui um conhecimento prévio, porém não domina inteiramente.
RP06 - Tutorial	Deve ser apresentado um breve resumo de como utilizar as funcionalidades, tais como conteúdo, animações, gerenciamento de controles para facilitar o processo de uso.
RP07 - Jogo educativo	Concede uma fonte de recreação com o intuito de proporcionar determinado modelo de aprendizagem.
RP08 - Informativo	Apresenta informações em formas visuais, tais como: gráficos, tabelas ou textos.

Fonte: AUTORES (2022)

5.5 GameTest: um protótipo de jogo para suporte ao ensino-aprendizagem de teste de software

Uma vez definidos as partes interessadas e os requisitos (funcionais, não-funcionais e pedagógicos), partiu-se para a criação do protótipo, desde a criação do enredo, especificação dos elementos, até a construção das telas que representam as funcionalidades propostas.

O protótipo construído recebeu o nome de GameTest por se tratar de uma proposta de jogo voltado para o ensino de Teste de Software. A versão executável do protótipo encontra-se disponível online, permitindo a visualização e simulação do protótipo. Nas seções seguintes, serão apresentados o enredo e a discussão das telas do protótipo.

5.6 Enredo

O protótipo do jogo proposto foi inspirado nos filmes e livros de Harry Potter (ROWLING, 2000). Na trama, João é um garoto de 10 anos de uma cidade do interior do Brasil. No dia do seu aniversário, ele recebe uma carta pelo correio. Ao abri-la, percebe que se trata de um ilustre convite para participar da maior escola de software do mundo, a SoftMun. João estava perplexo, não imaginava que seria agraciado com tal honra. A SoftMun, é a escola de desenvolvimento de software mais cobiçada do mundo, todos desejavam ir para ela e ao que parece, João estava em negação, pois não se achava merecedor de tal conquista.

Dois meses depois, chegou o seu primeiro dia de aula. Ele estava muito ansioso; afinal, mesmo após um tempo, ainda tinha receio de não ser capaz de obter bons resultados na SoftMun. No entanto, devido a alguns contratemplos, ele acabou perdendo sua mochila. Nela estava seu computador de última geração, com os melhores recursos disponíveis.

Foi então que teve a ideia de ir à sala de segurança da escola para checar as câmeras, mas havia um problema, João teria que resolver alguns desafios de teste de software para conseguir o acesso e ele não tinha tantos conhecimentos para isso. Foi então que ele resolveu utilizar o GameTest para ajudá-lo nessa jornada pela procura de seu computador de forma a contribuir para seu entendimento e aprendizagem dos conceitos de teste.

Apresentado o enredo, o jogo também busca abordar conceitos introdutórios de teste de software, teste de unidade e testes funcionais. A proposta inicial do GameTest é um jogo do tipo single-player, que possibilita a participação de um único jogador por vez.

5.6 Prototipação das telas

Os componentes do GameTest foram criados diretamente na ferramenta Figma. Os ícones utilizados foram retirados do *Iconify*, um plugin instalado na própria Figma. Na Figura

2, são apresentadas as telas iniciais de *Login*, Cadastro de Usuário e a Tela Inicial que dá acesso ao jogo. Ao iniciar o jogo, o usuário tem acesso a tela de *login* (Figura 2(a)), podendo escolher entre logar na sua conta, redefinir senha e cadastrar uma nova conta. A Figura 2(b) diz respeito ao cadastro de uma nova conta. Ao realizar *login*, será liberado o acesso à tela inicial do jogo (Figura 2(c)) e a opção de conhecer a história do jogo será evidenciada; abordando os requisitos funcionais RF01 e RF02.

Figura 2 - Tela de Login



Fonte: AUTORES (2022)

Na Figura 3, são apresentadas as telas de recuperação de senha e o modal que possibilita a edição dos dados do perfil, abordando os requisitos funcionais RF03 e RF04. A Figura 3(a) refere-se à tela que apresenta uma mensagem para que o usuário possa recuperar sua senha. A Figura 3(b) apresenta o modal para edição dos dados cadastrais.

Figura 3 - Recuperação de Senha



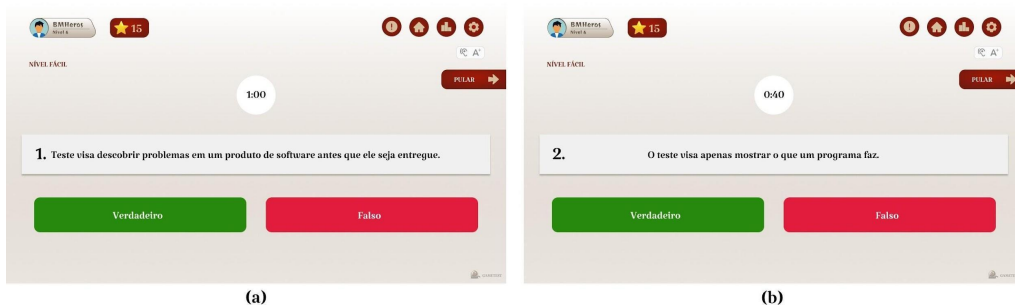
Fonte: AUTORES (2022)

No requisito funcional RF06 tem-se contextualização sobre como é iniciado todo o enredo e também quem é o personagem principal do jogo, com as opções de avançar e retornar à tela anterior, essa primeira parte da história dá liberdade a quem está lendo para explorar cada parte desse contexto.

Para dar início ao jogo é solicitado ao usuário qual tipo de assunto ele deseja abordar: teste funcional, unitário ou ambos. E, por fim, o usuário deve selecionar qual nível de dificuldade deseja jogar: fácil, médio ou difícil. Após escolher o nível do jogo é possível iniciar a partida, abordando o requisito funcional RF07.

Na Figura 4, são apresentadas perguntas do nível fácil do jogo. No nível fácil, foram definidas questões de verdadeiro ou falso, e vale salientar que também é possível observar que as questões possuem tempo limite para sua resolução.

Figura 4 - Algumas questões do Nível Fácil

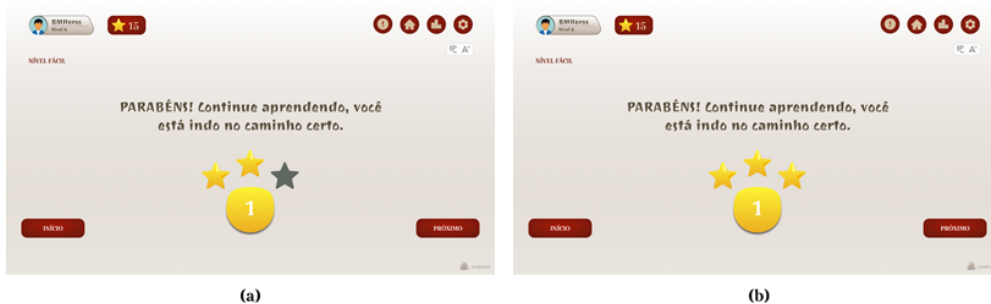


Fonte: AUTORES (2022)

Também é possível visualizar a opção de pular para a próxima questão bem como as funcionalidades dispostas na barra superior, como: menu de usuário, visualização de desempenho e ícone de reportar erro.

Após cada fase percorrida do jogo, é exibido um *feedback* ao usuário, mostrando o seu desempenho. Na Figura 5, é possível ver um exemplo de como será o *feedback* ao usuário. Será ilustrada a quantidade de estrelas ganhas em cada fase e a possibilidade de ir para a próxima questão ou retornar ao início do jogo; abordando o requisito funcional RF09.

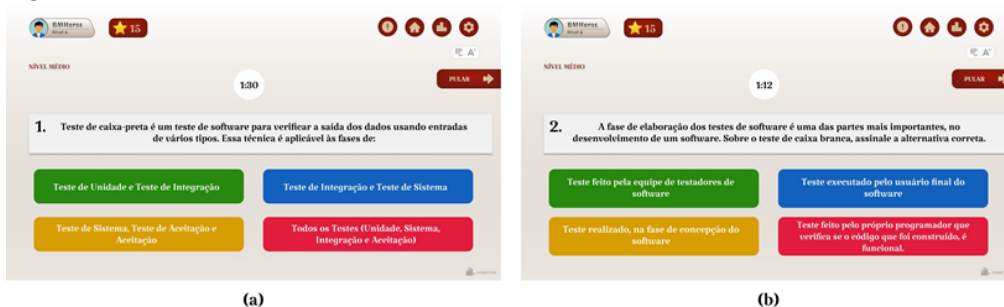
Figura 5 - Telas de *feedback* para o usuário ao finalizar uma partida no Nível Fácil



Fonte: AUTORES (2022)

Na Figura 6, são apresentadas as questões definidas para o nível médio: foram definidas questões de múltipla escolha nas quais é possível observar a opção de tempo limite definido para cada questão, a opção de pular para a próxima questão bem como as funcionalidades dispostas na barra superior: menu de usuário, visualização de desempenho, ícone de reportar erro e opções de acessibilidade para audição e aumento da fonte.

Figura 6 - Questões do Nível Médio

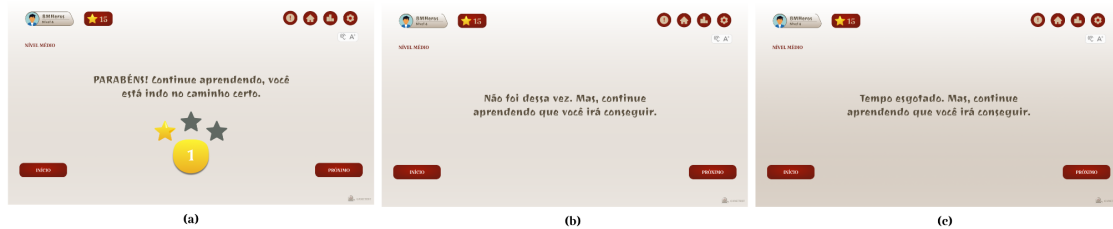


Fonte: AUTORES (2022)

Como *feedback* para o usuário após a escolha de sua resposta no nível médio, definiu-se telas para informar a quantidade de estrelas que o usuário irá ganhar; caso tenha resolvido a questão em mais de 50% até 90% do tempo estimado, contabilizará uma estrela

(Figura 7(a)). Caso a resposta da questão esteja errada, o sistema informa ao usuário (Figura 7(b)) e também há uma mensagem quando o tempo para a questão se esgota (Figura 7(c)).

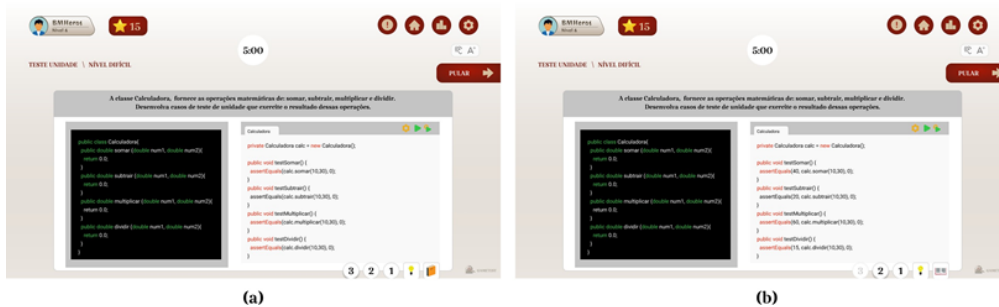
Figura 7 - Telas de *feedback* para o usuário ao finalizar uma partida



Fonte: AUTORES (2022)

Na Figura 8, apresenta-se o nível difícil, em que se tem um exemplo de um problema em código e a simulação de uma IDE (*Integrated Development Environment*) para o usuário desenvolver casos de testes de modo a exercitar o resultado das operações.

Figura 8 - Questões do Nível Difícil



Fonte: AUTORES (2022)

Neste nível, é ofertada a opção de dois tipos de dicas: uma de visualizar referências de livros para que o usuário consiga pesquisar, estudar sobre o assunto e voltar para resolver o problema; e a outra opção de dicas rápidas, que seria uma frase explicativa do conceito a ser utilizado para a resolução daquele problema.

Além disso, também são ofertadas três tentativas para que o usuário tente solucionar o problema. Uma vez utilizadas as dicas em uma partida, estas não poderão mais ser utilizadas nas demais partidas (Figura 8(b)), sendo representada a sua indisponibilidade por meio de ícones para simbolizar que as dicas já foram utilizadas e cores para as tentativas consumidas.

Na Figura 9(a), além das funcionalidades apresentadas nos níveis anteriores, tem-se também a possibilidade de o usuário tentar novamente resolver algum problema que inicialmente não conseguiu, indo para a sua segunda tentativa. E na Figura 9(b), é apresentado um modal, para o caso dele desejar voltar ao início e/ou salvar o seu desempenho.

Figura 9 - Telas de *feedback* para o usuário ao finalizar uma partida no Nível Difícil



(a)

(b)

Fonte: AUTORES (2022)

Na Figura 10, são apresentados os modais referentes à visualização do desempenho do usuário, que pode ser visto por nível do jogo. A opção de dicas e tentativas está disponível apenas no Nível Difícil (Figura 10(b)).

Figura 10 - Visualização do Desempenho



(a)

(b)

(c)

Fonte: AUTORES (2022)

6 RESULTADOS E DISCUSSÕES

Nesta seção é apresentada a metodologia utilizada para avaliar o protótipo (Seção 6.1) e a análise dos resultados (Seção 6.2) a partir da aplicação desses métodos.

6.1 Metodologia de Avaliação

A metodologia de avaliação foi elaborada a partir das especificações apresentadas na ISO 9241 que descreve os benefícios de medir usabilidade em termos de desempenho e satisfação do usuário (ABNT et al., 2002). Foram utilizadas as recomendações constantes na parte 10 da norma ISO 9241 (Princípio de Diálogos), que consiste em sete princípios voltados para o grau de satisfação dos usuários: (I) adequação à tarefa, (II) auto-descrição, (III) controlabilidade, (IV) conformidade, (V) tolerância a erros, (VI) suporte a individualização e (VII) adequação ao aprendizado (SILVA et al., 2015).

Também foi aplicada a teoria fundamentada nos dados, que consiste em construir uma teoria baseada nos dados, através da análise qualitativa destes e que poderá acrescentar ou trazer novos conhecimentos à área de estudo (CASSIANI; CALIRI; PELÁ, 1996).

A avaliação do protótipo foi realizada por meio de um questionário online⁶, disponibilizado no período do dia 1º ao dia 24/09/2021 e contou com a participação de 40 usuários interessados em teste de software. Primeiro, o questionário apresenta a ideia geral do

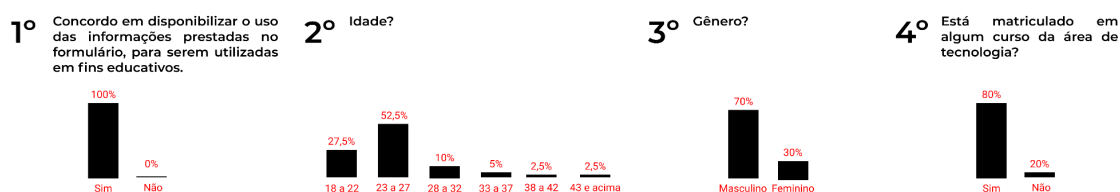
⁶ <https://forms.gle/a7yQyo7z2vvKZ9UUA>

protótipo, em seguida, os usuários acessam o *link* do protótipo e, depois de testarem o jogo, retornam para o preenchimento do questionário que consiste em questões de identificação do perfil dos usuários, seguida de perguntas voltadas para a aplicação dos princípios e, por fim, uma última questão para dúvidas, sugestões de melhorias e críticas.

6.2 Análise dos Resultados

As respostas dos usuários coletadas na avaliação do protótipo do GameTest foram consolidadas e analisadas a partir da ISO 9241 e pela teoria fundamentada nos dados. Ao todo foram coletadas 40 respostas, em que 100% dos participantes do questionário concordaram com a disponibilização dos dados para fins acadêmicos. Houve uma maior predominância de usuários do sexo masculino (70%) e apenas 30% do sexo feminino (Figura 11).

Figura 11 - Caracterização dos usuários



Fonte: AUTORES (2022)

Como mencionado na Seção 5.1, o GameTest foi desenvolvido com o foco em qualquer pessoa que tenha interesse em aprender sobre teste de software, com ênfase em alunos da área de computação. Assim, para recrutar participantes para a pesquisa, utilizou-se esse critério, então o questionário foi enviado a coordenações de curso da área de Computação da região Nordeste; além disso, foi divulgado pelas redes sociais dos autores do trabalho.

Não foi necessária a aprovação da pesquisa pelo comitê de ética, pois não foram coletados quaisquer dados sensíveis que possibilitem a identificação dos participantes; no entanto, para dar maior liberdade aos usuários, antes de iniciar o questionário, a pesquisa foi apresentada e ao final pergunta-se se o usuário concorda em disponibilizar os seus dados para serem utilizadas no trabalho, em caso positivo, as demais perguntas são apresentadas, caso contrário o formulário é finalizado antes de iniciar a apresentação das demais perguntas.

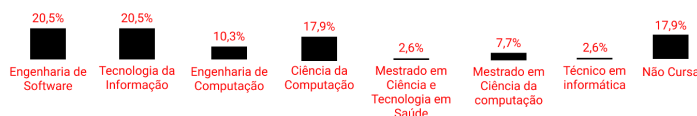
A idade dos participantes variou de 18 a 43 anos, com uma maior predominância na faixa etária de 23 a 27 anos (52,5%). Além disso, quando questionados se estavam matriculados em cursos de computação, 80% confirmaram que sim (Figura 11).

Na Figura 12, apresentam-se os cursos que os usuários estavam realizando no momento da pesquisa. Dos 40, apenas 32 estavam matriculados em cursos de Computação: Engenharia de Software, Tecnologia da Informação, Engenharia de Computação, Ciência da Computação, Mestrado em Ciência e Tecnologia em Saúde, Mestrado em Ciência da Computação e Técnico em Informática.

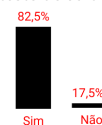
Destes, os cursos de Engenharia de Software, Tecnologia da Informação e Ciência da Computação obtiveram a maior predominância e 17,9% não estavam matriculados em nenhum curso de Computação. Quando questionados se já cursaram a disciplina de teste de software, 82,5% informaram que sim (Figura 12).

Figura 12 - Cursos dos usuários na área de tecnologia.

5º Se a sua resposta foi "SIM", qual o seu curso?



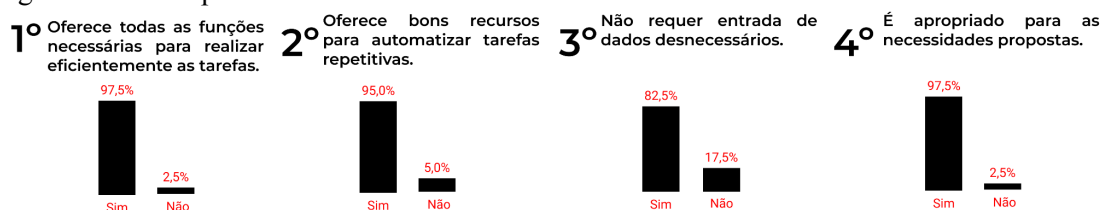
6º Você já pagou ou está pagando uma disciplina de teste de software?



Fonte: AUTORES (2022)

As quatro primeiras perguntas relacionadas à usabilidade abordam o Princípio I - adequação à tarefa, como é possível observar na Figura 13. A primeira questão abordada foi se o protótipo oferece todas as funções necessárias para efetuar de forma eficiente todas as tarefas propostas, 97,5% responderam que sim. Quando questionados se o protótipo oferece bons recursos para automatizar as tarefas repetitivas, 95% responderam que sim.

Figura 13 - Princípio I.



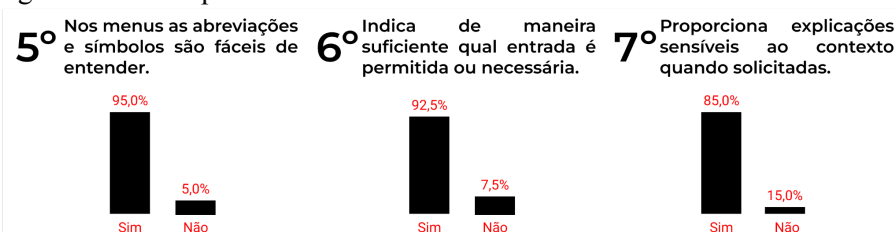
Fonte: AUTORES (2022)

Ainda na Figura 13, foi questionado se o protótipo não requer entrada desnecessária de dados, 82,5% dos participantes responderam que sim. Por fim, foi questionado se o protótipo é apropriado para as necessidades propostas e 97,5% responderam que sim.

Todos os quatro quesitos propostos pelo princípio de Adequação à Tarefa receberam um *feedback* positivo, com aceitação superior a 80% (Figura 13).

Os pontos relacionados ao Princípio II - Auto-Descrição - estão disponíveis na Figura 14. O primeiro ponto questionado foi se os menus, abreviações e símbolos são fáceis de serem entendidos e 95% responderam que sim.

Figura 14 - Princípio II

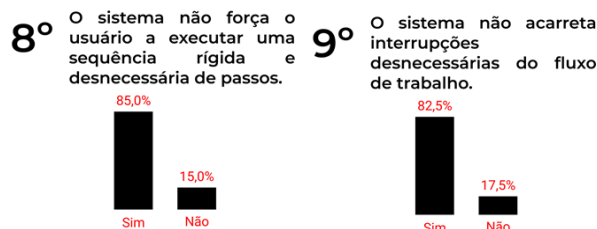


Fonte: AUTORES (2022)

O segundo ponto do Princípio II é se o protótipo indica de maneira suficiente quais as entradas que são permitidas ou necessárias na aplicação, 92,5% responderam que sim. Para finalizar as perguntas do Princípio II, foi questionado se o protótipo proporciona explicações sensíveis ao contexto quando solicitado, 85% dos participantes responderam sim. O protótipo recebeu uma análise positiva em relação ao Princípio II, aceitação maior ou igual a 85%.

Quanto à controlabilidade, Princípio III, pode-se observar o resultado na Figura 15. Foram realizados dois questionamentos, primeiro se o sistema não força o usuário a executar uma sequência rígida e desnecessária de passos, em que 85% responderam que sim. O segundo foi se o sistema não acarreta interrupções desnecessárias durante o fluxo de execução, e 82,5% responderam que o sistema não causa interrupções desnecessárias.

Figura 15 - Princípio III

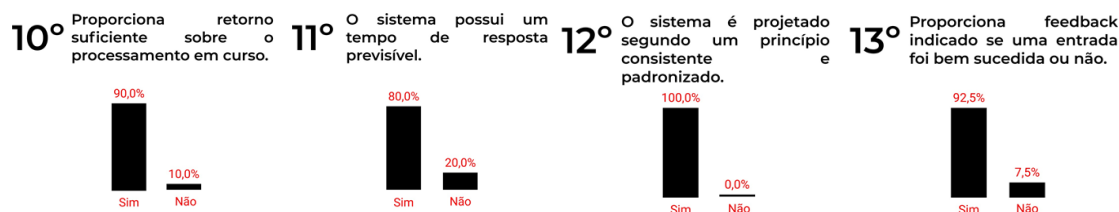


Fonte: AUTORES (2022)

O Princípio IV, conformidade com as expectativas dos usuários, pode ser observado na Figura 16. Para a 10ª pergunta 90% dos participantes responderam que o sistema proporciona retorno suficiente sobre o processamento em curso. Na 11ª pergunta, 80% dos participantes responderam que o sistema possui um tempo de resposta previsível.

Ainda observando a Figura 16, a 12ª pergunta foi respondida por unanimidade, todos responderam que o sistema foi projetado segundo um princípio consistente e padronizado.

Figura 16 - Princípio IV



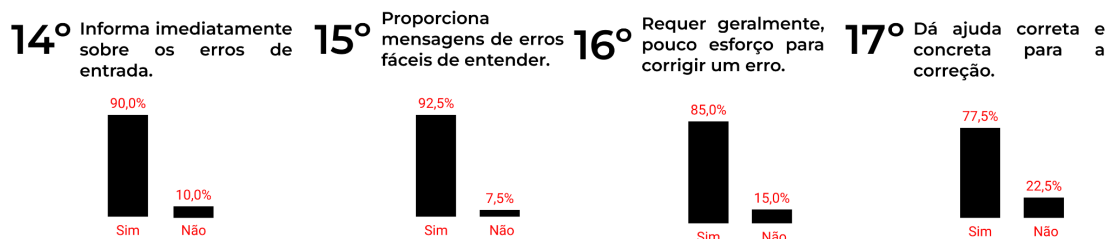
Fonte: AUTORES (2022)

A última pergunta relacionada ao IV princípio é a 13ª: se o sistema proporciona *feedback* indicando se uma entrada foi bem-sucedida ou não; e 92,5% responderam que sim. Como visto na Figura 16, a maioria das respostas foram positivas acerca das questões colocadas no questionário em relação ao cumprimento do Princípio IV.

O Princípio V, tolerância a erros, pode ser visto na Figura 17. A 14ª pergunta é sobre o sistema informar imediatamente sobre os erros de entrada e, 90% dos participantes afirmaram que sim. A 15ª questiona sobre o sistema proporcionar mensagens de erros fáceis de entender, 92,5% responderam que sim.

Ainda relacionando ao Princípio V (Figura 17), a 16ª pergunta questiona se o sistema requer, geralmente, pouco esforço para corrigir um erro; 85% das pessoas responderam que sim. A última pergunta, checka se o sistema dá a ajuda correta e concreta para a correção de erros, 77,5% das pessoas responderam que sim. A Figura 17 também oferece subsídios para uma avaliação positiva sobre o sistema.

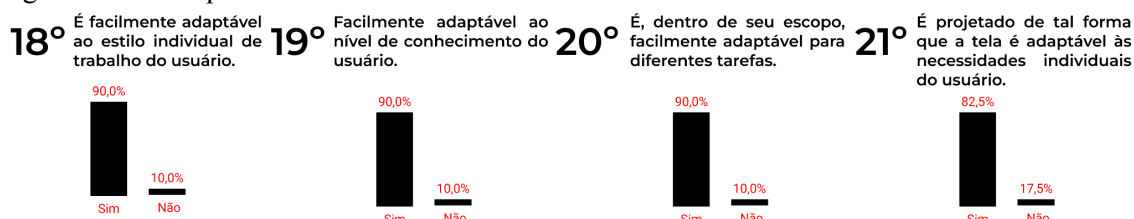
Figura 17 - Princípio V



Fonte: AUTORES (2022)

O VI princípio, suporte à individualização, apresenta quatro questões, disponíveis na Figura 18. A 18ª pergunta é sobre o sistema ser facilmente adaptável ao estilo individual de trabalho do usuário, e 90% responderam que sim. A 19ª questiona sobre o sistema ser facilmente adaptável ao nível de conhecimento do usuário, e 90% concordaram. A 20ª pergunta é sobre se, dentro do escopo, o sistema é facilmente adaptável para diferentes tarefas, e 90% das respostas foram que sim. Por fim, a 21ª pergunta é sobre o sistema ser projetado de tal forma que a tela é adaptável às necessidades individuais do usuário e 82,5% dos participantes concordaram. A Figura 19 traz uma satisfação dos participantes acerca do VI princípio, assim, pode-se considerar que o GameTest oferece suporte à individualização.

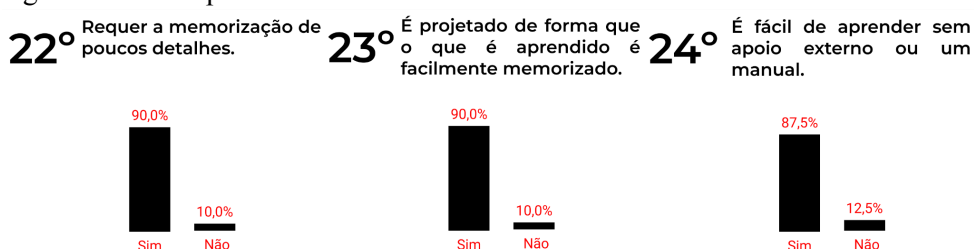
Figura 18 - Princípio VI.



Fonte: AUTORES (2022)

Por fim, o Princípio VII, adequação ao aprendizado, está disponível na Figura 19. Neste princípio, foram realizadas três questões: a primeira, se o protótipo requer a memorização de poucos detalhes, 90% responderam sim.

Figura 19 - Princípio VII.



Fonte: AUTORES (2022)

Ainda referente à Figura 19, o segundo questionamento foi se o sistema é projetado de forma que o aprendizado seja facilmente memorizado e 90% dos participantes responderam que sim. Para finalizar o questionário, com respeito aos princípios, foi questionado se o protótipo apresenta um sistema que é fácil de aprender sem o apoio de manuais e 87,5% responderam que sim.

Ao final da análise dos dados obtidos a partir dos princípios questionados é possível afirmar que em todos os sete princípios, a avaliação do protótipo foi positiva.

Para finalizar o questionário foi solicitado um *feedback* discursivo sobre o protótipo, em que foi solicitado que os participantes fornecessem suas críticas e sugestões de melhorias para o protótipo. Foram obtidas 15 respostas analisadas à luz da teoria fundamentada dos dados. As respostas foram divididas em quatro categorias: satisfação de usuário, em que foi levado em consideração os trechos de comentários nos quais os participantes destacaram os pontos positivos da aplicação; recomendações para o protótipo, sendo levado em consideração os argumentos que sugerem melhorias e novas ideias; recomendações para o questionário, que tratam de trechos com sugestões para uma melhor abordagem do questionário aplicado, e, por fim, respostas fora do escopo, que tratam de pontos fora do contexto da pesquisa.

7 CONSIDERAÇÕES FINAIS

No ensino de Teste de Software tem-se uma grande dificuldade quanto à utilização de ferramentas voltadas para a parte prática da disciplina. Nesse sentido, os jogos educacionais podem ser uma boa ferramenta para minimizar essa dificuldade e auxiliar na parte prática da disciplina de forma divertida e motivadora, bem como melhorar o processo de ensino-aprendizagem dos conteúdos fundamentais para o entendimento da disciplina. Neste trabalho, foi proposto o desenvolvimento de um protótipo de jogo educacional visando auxiliar na parte prática do ensino de teste de software.

O desenvolvimento do protótipo foi inspirado nos filmes e livros de Harry Potter, e conta a história de João, um garoto de uma cidade do interior que tinha o sonho de estudar em uma das maiores escolas de software do mundo. O protótipo buscou abordar três níveis de dificuldade: nível fácil, médio e difícil. Além disso, aborda conceitos de testes funcionais, testes unitários e ambos.

A validação do protótipo consistiu na aplicação de um formulário online com base na ISO 9241, e sua avaliação também considerou a teoria fundamentada nos dados. A partir dos resultados obtidos, percebe-se uma boa aceitação do protótipo proposto de modo que ele pode vir a auxiliar na parte prática da disciplina. Como trabalhos futuros, pretende-se realizar o desenvolvimento do jogo e fazer sua validação com outros usuários baseando-se em metodologias voltadas para a avaliação de jogos.

REFERÊNCIAS

ABNT, A. d. N. et al. **Requisitos ergonômicos para trabalho de escritórios com computadores parte 11—orientações sobre usabilidade**. AAB d. N. Técnicas (Ed.). Brasil: ABNT— Associação Brasileira de Normas Técnicas, 2002.

ARAGÃO, J. W. M. d.; NETA, M. A. H. M. **Metodologia científica. Superintendência de Educação a Distância**, 2017.

BENITTI, F. B. V.; ALBANO, E. L. **Teste de software: o que e como é ensinado**. In: WEI XX Workshop sobre Educação em Computação. [S.l.: s.n.], 2012.

CASSIANI, S. H. D. B.; CALIRI, M. H. L.; PELÁ, N. T. R. **A teoria fundamentada nos dados como abordagem da pesquisa interpretativa**. Revista latino-americana de enfermagem, SciELO Brasil, v. 4, p. 75 - 88, 1996.

COUTINHO, E. F.; BEZERRA, C. I. M. **Uma avaliação inicial do jogo para o ensino de testes de software itestlearning sob a ótica de um software educativo**. In: Congresso sobre Tecnologias na Educação. [S.l.: s.n.], v. 3, 2018.

DELAMARO, M.; JINO, M.; MALDONADO, J. **Introdução Ao Teste De Software**. [S.l.]: Elsevier Brasil, 2013.

DINIZ, L.; DAZZI, R. **Jogo digital para o apoio ao ensino do teste de caixa-preta**. In: **Anais do X Simpósio Brasileiro de Qualidade de Software**. Porto Alegre, RS, Brasil: SBC, p. 231–245, 2011.

FARIAS, V. et al. **itest learning: Um jogo para o ensino do planejamento de testes de software**. Anais do V Fórum de Educação em Engenharia de Software–FEES, p. 1–8, 2012.

FRANCISCO, R. E. et al. **Juiz online no ensino de CS1 - lições aprendidas e proposta de uma ferramenta**. *Revista Brasileira de Informática na Educação*, v. 26, n. 03, p. 163, 2018.

GALVÃO, M. C. B. **O levantamento bibliográfico e a pesquisa científica. Fundamentos de epidemiologia**. 2ed. A, v. 398, p. 1–377, 2010.

HENRIQUE, M. S. **EDUCATALOG4RE: um catálogo de requisitos para auxiliar o desenvolvimento de softwares educacionais**. Dissertação (Mestrado) - Universidade Federal de Pernambuco, 2016.

LOZZA, R.; RINALDI, G. P. **O uso dos jogos para a aprendizagem no ensino superior**. *Caderno PAIC*, v. 18, n. 1, p. 575–592, 2017.

MOREIRA, J. A.; MONTEIRO, W. M. **O uso da computação desplugada em um contexto de gamificação para o ensino de estrutura de dados**. *RENOTE*, v. 16, n. 2, p. 546–555, 2018.

PONTES, M. B. **Introdução a testes de software**. *Engenharia de Software Magazine*. Ano, v. 1, n. 11, 2009.

PRESSMAN, R.; MAXIM, B. **Engenharia de Software-8ª Edição**. [S.l.]: McGraw Hill Brasil, 2016.

QUEIROZ, R.; PINTO, F.; SILVA, P. **Islandtest: jogo educativo para apoiar o processo ensino-aprendizagem de testes de software**. In: SBC. *Anais do XXVII Workshop sobre Educação em Computação*. [S.l.], p. 533–542, 2019.

REZENDE, D. A. **Engenharia de software e sistemas de informação**. [S.l.]: Brasport, 2006.

RIBEIRO, L. O. M.; TIMM, M. I.; ZARO, M. A. **Modificações em jogos digitais e seu uso potencial como tecnologia educacional para o ensino de engenharia**. *RENOTE-Revista Novas Tecnologias na Educação*, v. 4, n. 1, 2006.

RIBEIRO, T. P. B. **ilearntest: Jogo educativo para aprendizagem de testes de software**. 2014.

SALOMÃO, R. G. **Análise da relevância de testes de regressão para o mercado de desenvolvimento de software do triângulo mineiro**. Universidade Federal de Uberlândia, 2016.

SILVA, C. et al. **gtest learning: Um jogo para ensino básico de teste de software**. In: Congresso Regional sobre Tecnologias na Educação—Recife-PE. [S.l.: s.n.], 2016.

SILVA, T. G. d.; MÜLLER, F. M.; BERNARDI, G. **Panorama do ensino de engenharia de software em cursos de graduação focado em teste de software: uma proposta de aprendizagem baseada em jogos**. UFRGS, 2011.

SILVA, G. G. da et al. **Análise da usabilidade conforme as recomendações da norma iso 9241-um estudo de caso**. Blucher Design Proceedings, v. 2, n. 3, p. 256–261, 2015.

SOMMERVILLE, I. **Software Engineering: 9ª Edição**. [S.l.]: Addison-Wesley, 2011.

VAZQUEZ, C. E.; SIMÕES, G. S. **Engenharia de Requisitos: software orientado ao negócio**. [S.l.]: Brasport, 2016.

WANGENHEIM, C. G. v.; WANGENHEIM, A. v. **Ensinando computação com jogos**. [S.l.]: Bookess Editora, 2012.