

IslandTest: jogo educativo para apoiar o processo ensino-aprendizagem de testes de software

Rafael Jesus de Queiroz¹, Fabrício de Sousa Pinto¹, Paulo Caetano da Silva²

¹Faculdade de Tecnologia e Ciências (FTC) – Colegiado de Sistemas de Informação
Rua Ubaldino Figueira, 200 – Recreio, 45020 510 - Vitória da Conquista – BA– Brasil

²Universidade Salvador (Unifacs) - Programa de Pós Graduação em Sistemas e
Computação. Rua Dr. José Peroba, 251 , Stiep , 41.170-235 – Salvador – BA – Brasil

rafyusq@yahoo.com.br, fapinto.vic@ftc.br, paulo.caetano@unifacs.br

Abstract. Educational games have been gaining more space and facilitating the teaching-learning process. Software testing is one of the most important tasks of the software development process, to ensure the quality of the product. Due to this importance and aiming to make the lectures more attractive and motivating, an online tool was developed with the objective of offering an educational game, called IslandTest, to help the teaching-learning process in the area of software testing. For the development of the game, the languages PHP (Hypertext Preprocessor), JavaScript, CSS and HTML and MySQL database were used. The game was validated by students regarding motivation, user experience, learning and learning objectives and the results showed that it facilitated the teaching-learning process of software testing.

Resumo. *Os jogos educativos vêm ganhando cada vez mais espaço e facilitando o processo ensino-aprendizagem. Testes de software é uma das tarefas mais importante do processo de desenvolvimento de software, para garantir a qualidade do produto. Devido a essa importância e objetivando tonar mais atraente e motivadora as aulas expositivas, foi desenvolvida uma ferramenta online com o objetivo de oferecer um jogo educativo, denominado IslandTest, para auxiliar o processo de ensino-aprendizagem na área de testes de software. Para o desenvolvimento do jogo, foram utilizadas as linguagens PHP (Hypertext Preprocessor), JavaScript, CSS e HTML e banco de dados MySQL. O jogo foi validado por estudantes com relação a motivação, experiência do usuário, aprendizagem e objetivos da aprendizagem e os resultados mostraram que facilitou o processo ensino-aprendizagem de testes de software.*

1. Introdução

No contexto atual de ensino é possível notar o frequente uso de aulas expositivas pelos docentes em ambientes educacionais. Este tipo é adequado para a apresentação de conhecimentos teóricos de forma eficiente para turmas grandes, porém possui diversas desvantagens, dentre elas a dispersão, fazendo necessário uso de metodologias que promovam um maior engajamento dos estudantes.

Wangenheim (2012) defende que aulas expositivas infelizmente não estimulam um aprendizado profundo, apenas algo superficial e que não motiva o estudante a aprender. Para contornar isso e levar o estudante a exercitar a aplicação do conhecimento é necessário buscar outras maneiras de ensino. (VALENTIM, 2002, p.12).

O número de intervenções em sala de aula a partir do uso de jogos para estimular a aprendizagem tem aumentado e a sua utilização promove um auxílio na relação teoria-prática, na transmissão de conteúdos essenciais em todos os níveis de ensino.

Segundo Brenelli (1996, p.21) “o jogo é importante para o desenvolvimento físico, intelectual e social e vem ampliando sua importância, deixando de ser um simples divertimento e tornando-se a ponte entre a infância e a vida adulta”. Sendo assim, o seu bom uso estará favorecido pelo contexto lúdico essencial e motivador para todas as idades, pois este oportuniza o sujeito a utilizar mais a sua criatividade e o domínio de si. É também um excelente mecanismo para a afirmação da personalidade, e ainda lida com o imprevisível. Destarte, o que prende o sujeito é o desafio de vencer.

Dentro do currículo dos cursos de computação, um dos componentes obrigatórios é engenharia de software, que aborda, dentre outros assuntos, testes de software, conteúdo muito importante para a criação de sistemas computacionais de qualidade. Segundo Myers (2004, p. 6), “Teste de Software é o processo de execução de um programa ou sistema com a intenção de encontrar erros”. Contudo, esse conteúdo ainda é repassado, na maioria das vezes, para os alunos na forma de aulas expositivas, que de acordo com Wangenheim *et al.* (2012, p. 14) “são eficientes para apresentar conhecimento teórico para um público grande, porém têm diversas desvantagens, uma delas é a falta de concentração depois de aproximadamente 15 minutos”.

Neste sentido, os jogos educativos podem vir a contribuir para a demanda prática, nos cursos de computação, de forma lúdica e motivadora, que auxilie o processo ensino-aprendizagem de conteúdos essenciais para a formação de um bom profissional da área de TI.

Pensando nesta perspectiva, o objetivo dessa pesquisa foi o desenvolvimento de um jogo para apoiar o processo ensino-aprendizado de testes de software, parte técnica da disciplina de engenharia de software, conteúdo este, essencial para a formação dos futuros profissionais da área. O uso de jogos para ensino de testes de software tem grande potencial, pois oferece ao aluno a possibilidade de ter experiências práticas de aprendizado, aprenderem com os erros e receberem *feedback* imediato.

Com o intuito de suprir uma das lacunas de atividades práticas que abordassem testes de software, este trabalho se propõe a desenvolver um *serious game*, que não fosse apenas um quiz, *mas que* abordasse um conjunto de tipos de testes (unidade, integração, validação e sistema) e os processos do plano de testes, de forma contextualizada e lúdica, com desafios para instigar e motivar o jogador a aprendê-lo.

Este artigo apresenta-se com a seguinte estrutura: na Seção 2 são apresentados o processo de testes de Software. Na Seção 3 é mostrada a metodologia utilizada para o desenvolvimento do trabalho, na Seção 4 é descrito o processo de desenvolvimento do jogo educativo: *IslandTest*, na Seção 5 é apresentada a validação do jogo, na Seção 6

são discutidos os trabalhos correlatos e na Seção 7 as considerações finais e conclusão são discutidas.

2. O Processo de Testes na Engenharia de Software

O processo de desenvolvimento de um software possui as atividades de: análise, projeto, codificação, testes e manutenção. Existem várias tentativas no sentido de definir as atividades de testes, desde a visão intuitiva sobre testes, até uma definição formal. Para Peters e Pedrycz (2001) os testes de software determinam quando um sistema de software pode ser liberado para uso atendendo corretamente as necessidades dos usuários.

Os testes são mais do que meios de detecção e correção de erros são, além de tudo, indicadores da qualidade do produto. Segundo Rios (2007, p. 349), “o teste de software é uma atividade na qual um produto, sistema ou componente é executado sob condições especificadas, com observação e registro dos resultados e avaliação de um ou mais aspectos”. É um procedimento técnico de qualidade que deve fazer parte da rotina de um desenvolvedor.

A qualidade do software está atrelada à quantidade de investimentos feitos no processo de testes, no qual uma organização poderá ter grandes prejuízos, caso o software tenha sido mal testado. Dessa forma, a atividade de teste de software assume um importante papel, pois consiste em executar o software de forma controlada e monitorada, a fim de avaliar seu comportamento (Crespo et al., 2004). Esse conteúdo programático deve ser aprendido desde a graduação, algo que ainda é incipiente.

O processo de teste é composto das seguintes fases (conforme ilustrado na Figura 01): as fases de *Planejamento* e *Preparação* seguem em paralelo às demais fases, pois continuam em andamento durante todo o projeto. O produto da fase de planejamento, uma vez concluído, deve ser acompanhado e controlado. Na fase de preparação os ambientes de gerência de mudanças e de gerência de configuração são adequados ao projeto de testes, além de ser preparada a infraestrutura a ser utilizada no projeto (hardware, software e pessoal).

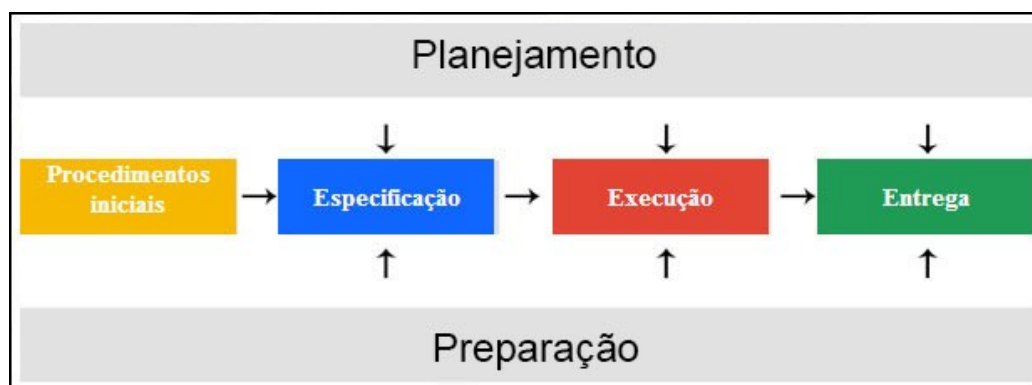


Figura 01: Fases de um processo de testes

Fonte: Rios (2013)

Rios (2013) descreve cada parte do processo de teste. São elas: (i) *Planejamento e controle*: Elaboração e revisão da Estratégia de Testes e do Plano de Testes; (ii)

Preparação: Preparo do ambiente de testes, incluindo equipamentos, rede, pessoal, software e ferramentas; (iii) *Especificação*: Elaboração e revisão dos casos de testes, *scripts* e dos roteiros; (iv) *Execução*: Execução dos testes planejados conforme os casos de testes, *scripts* e dos roteiros / cenário de testes; (v) *Entrega*: Conclusão do processo de testes com a entrega do sistema para o ambiente de produção.

O processo de testes deve ser realizado paralelamente ao desenvolvimento do sistema e deve basear-se em uma metodologia aderente, com pessoal técnico qualificado, em ambientes e ferramentas adequadas. A metodologia de testes deve ser o documento básico para organizar a atividade de testar aplicações no contexto da organização. Para Bartié (2002, p. 16) “qualidade de software é um processo que focaliza todas as etapas com o objetivo de garantir a conformidade de processo e produtos, prevenindo e eliminando defeitos”. E que deve ser adotada por todos os profissionais desenvolvedores. Sendo assim, a grande necessidade de tais conteúdos serem trabalhados nos cursos, de forma prática, aplicando o que se discute na teoria.

3. Metodologia

Segundo Gil (2007) a ciência tem como objetivo fundamental chegar a veracidade dos fatos e para que o conhecimento possa ser considerado científico é necessário identificar as operações mentais e técnicas que possibilitam a sua verificação. No Quadro 01 é apresentada a metodologia adotada nesse artigo.

Quadro 01: Metodologia da pesquisa

Metodologia	
Tipo de Pesquisa	Exploratória e observação direta
População e amostra	População: Instituição de ensino Superior (IES) Amostras: Quatro turmas de graduação
Coleta de dados	Questionário, entrevista e observação.
Análise de dados	Qualitativa.

As técnicas de pesquisa adotada neste artigo são do tipo exploratória e observação direta. Inicialmente foi feita uma revisão bibliográfica objetivando conhecer os elementos de um jogo educativo e sobre os testes de software. Posteriormente foi iniciado o processo de desenvolvimento do jogo educativo: *IslandTest*. Definiu-se: os personagens, desafios, objetivos, regras e restrições, enredo, recompensas e feedback. Foi escolhido o enredo de um seriado americano de sucesso, *Lost*, pois seria mais atrativo e ao mesmo tempo, seria motivador jogá-lo. Após a execução das atividades da engenharia de requisitos e modelagem do jogo utilizando UML (*Unified Modeling Language*), teve-se início o processo de codificação. As linguagens utilizadas para o desenvolvimento do jogo foram: CSS (*Cascading Style Sheets*), *JavaScript*, HTML (*HyperText Markup Language*), PHP (*Hypertext Preprocessor*) e o banco de dados foi

MySQL.

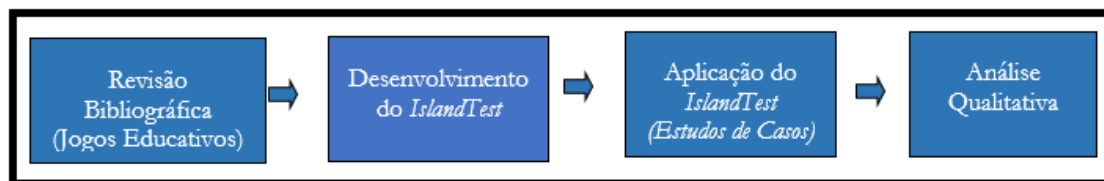


Figura 01: Fases de um processo de testes

Com relação à classificação da fonte de pesquisa é de campo, pois recolheu dados *in natura* como percebidos pelo pesquisador através dos estudos de caso. Com relação a validação do jogo, ele foi utilizado em quatro turmas de graduação em Sistemas de Informação, na disciplina Engenharia de Software. No primeiro estudo de caso, após os *feedback* dos estudantes foram sugeridas algumas melhorias que foram acatadas. Além disso, foi aplicado um questionário para a turma avaliar o *IslandTest*. Com relação a análise dos dados foi conduzida qualitativamente.

Com relação ao questionário foi composto de vinte e sete questões objetivas baseados em SAVI, et al (2011), dividido em quatro partes, avaliando o jogo quanto a sua motivação, experiência do usuário, aprendizagem e objetivos da aprendizagem. Foi aplicado após a utilização do jogo, de forma presencial.

4. *IslandTest*: jogo educativo para apoiar o processo ensino-aprendizagem de testes de software.

A versão final do jogo pode ser acessada de forma gratuita através do link: <http://www.itaz.com.br/islandtest>. Os códigos-fontes desse projeto encontram-se disponíveis no repositório: <http://sourceforge.net/projects/islandtest>, permitindo que se faça análises do código fonte do jogo, e.g. refatoração, inspeção do software, análise de métricas. Nas seções seguintes serão apresentados o enredo, objetivo e mecânica do jogo. Na seção a seguir será descrito o processo de desenvolvimento do *IslandTest*.

4.1 Projeto

Neste trabalho, teve-se a preocupação de desenvolver um jogo com vários recursos de animações que o tornasse atraente, isso foi possível programando os efeitos com a linguagem: *JavaScript*. Assim foi possível criar várias interações no jogo: boxes para serem movimentados, arrastar itens que não fazem parte de um determinado tipo de teste, para uma fogueira e queimá-los, ligar um notebook virtual, mudar cores de objetos, interagir com um celular para enviar mensagem, decodificar um celular, selecionar determinadas rotas em um mapa, arrastar as pedras da caverna, digitar código de um enigma, ligar e desligar um painel de controle, desativar uma bomba, acionar um painel de controle, desarmar bomba, dentre outros recursos. Além disso, foi criado um enredo com aventura, ação e drama que motivasse e prendesse a atenção do jogador para ir até o fim do jogo.

Com a linguagem de marcação HTML foi possível criar projetos que rodam em qualquer dispositivo que possua um navegador web, justificando assim o seu uso neste projeto. Neste projeto foi utilizado a arquitetura cliente/servidor.

Um dos elementos de um jogo é a competição. A pontuação do jogador será definida através de seu desempenho, baseado na quantidade dos desafios que ele venceu e nos *quizes* que acertou, a qual será utilizada para formar o *ranking* do jogo.

Para criar uma boa aparência no jogo, foi utilizado o CSS (*Cascading Style Sheets*). O *Adobe Photoshop CC* foi utilizado para editar e criar as imagens do jogo. Em paralelo ao desenvolvimento do jogo foram aplicados os testes de software para garantir que ele funcionasse da maneira esperada. Aplicaram-se os testes de unidade para verificar os menores componentes do código, os testes de integração para verificar se os componentes funcionavam juntos da maneira correta, os testes de sistema para validar a exatidão e perfeição na execução de suas funções. Por fim, os testes de validação, realizado por usuários finais a fim de verificar as funcionalidades e a usabilidade do jogo.

4.2 Enredo

O jogo proposto foi desenvolvido inspirado no seriado Americano *Lost*, de J.J. Abrams. Na trama, *Jack* é um estudante do curso de Sistemas de Informação e é fascinado por desafios. Ele decide fazer uma viagem para participar do *Congresso da Brasileiro de Engenharia de Software* (CBES), contudo, o avião em que ele está cai em uma misteriosa ilha tropical. Milagrosamente, Jack e outras pessoas sobreviveram. Agora ele precisa encontrar uma maneira de sair da ilha. Jack vai em busca de seu notebook nos destroços do avião, pois estava desenvolvendo um aplicativo de mensagens instantâneas para smartphones. Com muita sorte, ele encontra seu notebook intacto e a bateria estava 100% carregada, porém ainda faltava realizar os testes no aplicativo. A aventura começa como a missão de preparar o plano de testes e executá-lo, mas para isso eles precisam correr contra o tempo, e para isso viverão uma aventura cheia de desafios a serem vencidos, e.g. maldição na ilha, libertar prisioneiro, desativar bombar, lidar com estranhas criaturas da ilha, e como plano de fundo é utilizando a analogia para ensinar os vários tipos de testes de software.

O público-alvo deste jogo são os estudantes e docentes da área de Computação e áreas afins que tenham interesse em aprender/ensinar os conceitos de teste de software, geralmente conteúdo ministrado na disciplina de Engenharia de Software. Os assuntos abordados são: fases de um processo de teste, testes de: unidade, integração, sistemas, validação, caixa branca, caixa preta, segurança, carga, desempenho, alfa e beta. Além disso, são apresentados os conceitos de: defeito, erro, falha, depuração e casos de testes.

IslandTest é um jogo do tipo *single-player* (termo utilizado para definir um jogo que possibilita a participação de um jogador por vez), mas pode-se utilizar o ranking para competições. Ao iniciar o jogo, o jogador terá acesso à tela inicial, conforme ilustrado na Figura 02



Figura 02: Tela inicial do jogo *IslandTest*

5 Validação do Jogo

O *IslandTest* foi disponibilizado aos alunos do curso de Sistemas de Informação da Faculdade de Tecnologia e Ciências (FTC), Campus Vitória da Conquista – BA, que cursaram a disciplina Engenharia de Software. Posteriormente foi aplicado um questionário abordando questões referentes à utilização, importância, sugestões de melhoria e pontos fortes do jogo. Ele foi aplicado em quatro turmas, a turma A com 21 alunos, a turma B com 25 alunos, turma C com 19 alunos e a turma D com 20 alunos. O jogo foi aplicado antes do docente ministrar o conteúdo de testes de software.

Os oitenta e cinco alunos responderam o questionário disponibilizado por Savi, *et al.* (2011). Foram 27 questões que o usuário pode responder se ele concorda ou discorda do seu enunciado. Esse método avaliativo tinha como opção de resposta: discordo fortemente, discordo parcialmente, concordo parcialmente, concordo, concordo fortemente. Na seção de conceitos: o aluno teve suas respostas avaliadas em relação ao que ele sabia antes e depois de ter jogado o *IslandTest*. O questionário foi dividido em quatro partes, avaliando o jogo quanto a sua motivação, experiência do usuário, aprendizagem e objetivos da aprendizagem.

Para a motivação, conforme Figura 03 foram analisadas perguntas que atendem requisitos como: satisfação, confiança, relevância e a atenção do jogador durante e depois do jogo. Questionados quanto à satisfação, 60% dos alunos assinalaram que concordam com o nível de satisfação do jogo, 20% concordam parcialmente e os outros 20% concordam fortemente. Quanto à confiança 70% dos estudantes disseram que concordam com a forma de ensino de jogo, 10% concordam parcialmente e 20% concordam fortemente.

Ao avaliar os questionamentos dos alunos sobre a experiência do usuário no jogo, obtiveram-se resultados excelentes, principalmente tratando-se da diversão e desafios propostos aos jogadores. Os outros requisitos foram: Competência, Interação e Imersão obtiveram ótimos resultados.

Quando questionados sobre a experiência que o jogo contribuiu para o desempenho na vida profissional 10% dos alunos concordaram fortemente, 70% concordam e 20% alunos concordam parcialmente.

Os aspectos fortes citados foram: usabilidade, jogabilidade, conteúdo, qualidade das funcionalidades, boa explicação do conteúdo, torna o aprendizado mais fácil,

divertido, torna fácil a memorização do conteúdo, ajuda memorizar o conteúdo e, a forma de interação com o jogador. Sugestões de melhorias: deixar as perguntas mais simples, explicar mais cada fase, justificar erros, melhorar explicações das regras. Incluir níveis de dificuldade, implementar placar por tempo e inserir perguntas dinâmicas.

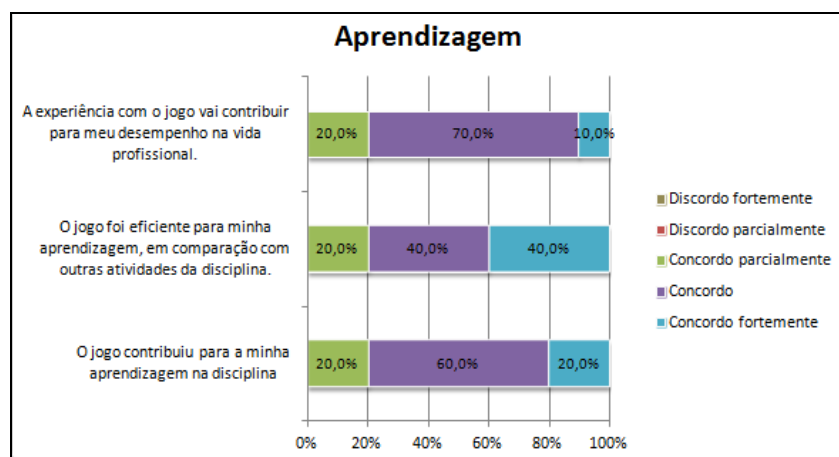


Figura 03: Análise da Aprendizagem do Jogo *IslandTest*

6 Trabalhos Correlatos

Alguns trabalhos relacionados são encontrados na literatura: *iTest Learning*, O Jogo das 7 Falhas, *uTest* e As aventuras de Jack Test. O *iTest Learning* de Farias *et al.* (2012) é um jogo voltado especificamente para a área de ensino de teste de software em sua fase de planejamento, simulando um projeto hipotético. O *Jogo das 7 Falhas* (Diniz, 2011) é um jogo de simulação *single-player*, no qual o jogador assume o papel de testador em uma equipe de teste de software. O objetivo do jogo é descobrir as falhas existentes nas funcionalidades do software a ser testado no menor tempo possível. *As aventuras de Jack Test* de Macêdo (2014) é um jogo de aventura e estratégia, onde o jogador assume o papel de engenheiro de teste para solucionar problemas que abordam conceitos das fases de teste de software.

Os jogos encontrados utilizam a taxonomia de Bloom (FERRAZ, 2010) e são restritos apenas ao nível de aprendizagem lembrar, se restringindo a *quizes*, conforme ilustrado na Figura 04. Já no *IslandTest*, os níveis aplicados são o de lembrar e entender.

No nível 1: lembrar, o jogador no *IslandTest* buscar reconhecer e recordar informações. Ele terá que lembrar as fases de um processo de testes. Como exemplo, temos as fases do processo de testes que são exibidas e através da interação: manipulação direita, utilizando a ação de arrastar, o jogador irá posicioná-las na sequência correta. Já no nível 2: entender, é o elemento capaz de fazer com que o jogador busque fazer sua própria interpretação do material educacional. Através do enredo do *IslandTest* é feita uma analogia com os tipos de testes de software. O jogador visualiza no enredo os ensinamentos aplicados, fazendo com que compreenda o que se passa em cada fase do processo de teste. O jogador irá adquirir conhecimento de testes de software que, posteriormente, deverão ser aplicados em novas situações.

Com relação a dinâmica do jogo foi inspirado no jogo educativo: *Ilha dos Requisitos*, embora o conteúdo abordado é sobre engenharia de requisitos (THIRY *et al.*, 2010) e (GONÇALVES, 2011).

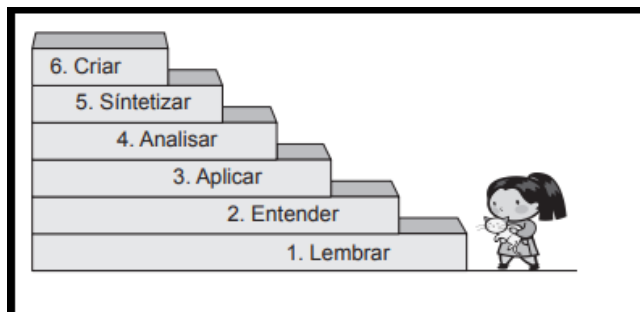


Figura 04: Taxonomia Revisada de Bloom

Fonte: Ferraz (2010)

Assim, um dos diferenciais do *IslandTest* é que cobre duas áreas da Taxonomia de Bloom, não se limitando a um quiz e é o único que está disponível online para ser utilizado.

7 Conclusão

Na fase de validação, na qual os alunos do curso puderam jogar, experimentar e praticar o conteúdo abordado neste trabalho, ficou claro, a partir da aplicação dos questionários pós jogo, que o objetivo de ensinar os conceitos básicos de teste de software foram alcançados. Os resultados do jogo mostraram-se satisfatórios, uma vez que na aplicação do questionário de avaliação cada conceito passou pela avaliação onde o aluno atribuiu uma nota de 1 (mais baixa) a 5 (mais alta) para seu nível de conhecimento antes e depois do jogo. As notas são avaliadas a partir de três fatores: Lembrar o que é: o aluno avalia se o jogo foi eficiente à aprendizagem e se jogo ajuda na memorização dos conceitos aplicados; compreender como funciona: é ponderada a compreensão do aluno antes e depois de ter jogado o *IslandTest*; Aplicar na prática: o discente analisou os conceitos quanto à sua aplicação na disciplina e/ou vida profissional. Obtiveram-se notas altas. Uma das limitações do jogo são as perguntas estáticas.

É importante ressaltar que os jogos educativos têm exercido um papel essencial no processo de ensino-aprendizagem, pois através do ato de jogar, o aluno pode interagir de forma lúdica e dinâmica, agregando assim, mais valor ao conhecimento, tornando-o mais significativo e atrativo. Além disto, a partir das pesquisas realizadas durante a construção e desenvolvimento desta proposta, notou-se que há uma lacuna no ensino prático da disciplina de Engenharia de Software. Foi constatado que, comparando aos trabalhos correlatos o *IslandTest* possui o diferencial de atingir níveis de entender e aplicar, enquanto os demais se restringi apenas ao nível de lembrar. Outro diferencial é o seu enredo baseado em uma aventura, o que motiva o jogador. Atualmente o *IslandTest* é único que se encontra disponível para uso no Brasil.

Como trabalhos futuros sugere-se usar enredos de outros seriados de sucesso, cobrindo outras áreas da engenharia de software e fazer melhorias no jogo conforme a sugestões dos participantes e o desenvolvimento de uma metodologia para aplicação da gamificação em engenharia de software.

REFERÊNCIAS

- Brenelli, R. P. (1996) “O Jogo como espaço para pensar: A construção de noções lógicas e aritméticas”, Campinas: Papirus.
- Crespo, A. N.; Silva, O. J.; Borges, C. A.; Salviano, C. F.; Teive, M.; Jino, M. (2004) “Uma metodologia para teste de software no contexto da melhoria de processo”. In: III Simpósio Brasileiro de Qualidade de Software. pp. 271- 285.
- Diniz, Lúcio Lopes, Dazzi, Rudimar Luís Scaranto. (2011) “Jogo digital para o apoio ao ensino do teste de caixa-preta”. In: X Simpósio Brasileiro de Qualidade de Software, Curitiba.
- Farias, Virgínia, Moreira, Carla, Coutinho, Emanuel, Santos, Ismayle S. (2017) “iTest Learning: um jogo para o ensino do planejamento de testes de software”. Disponível em: <https://sistemas.quixada.ufc.br/iTestLearning/descricao_projeto.jsf>. Acessado em: 05 de dezembro de 2017.
- Ferraz, A. P. C. M. et al. (2010) “Taxonomia de Bloom: revisão teórica e apresentação das adequações do instrumento para definição de objetivos instrucionais”. Gest. Prod., São Carlos, v. 17, n. 2, p. 421-431.
- Gil, Antonio Carlos. (2012) “Como elaborar projetos de pesquisa”. São Paulo, v. 5, n. 61, p. 16-17.
- Gonçalves, R. Q., Thiry, M., & Zoucas, A. (2011) “Avaliação da aprendizagem em experimentos com jogo educativo de engenharia de requisitos”. X Simpósio Brasileiro de Qualidade de Software (SBQS), Curitiba.
- Macêdo, Kelly Santos. (2014) “As Aventuras de Jack Test: jogo educacional para o apoio ao ensino de teste de software”. Trabalho de conclusão de curso.
- Myers, Glenford. (2004) “The Art of Software Testing”. In : Willand, Bruna de Quadros; Costa, Edemar. Desenvolvimento de aplicação voltada para geração automática de casos de teste. 2ª ed. Canoas.
- Peters, James, Pedrycz, Witold. (2001) “Engenharia de software”. Rio de Janeiro: Campus.
- Rios, E. Moreira, T. (2013) “Teste de Software”. 3 ed. Rio de Janeiro: Alta Books.
- Savi, R.; Gresse Von Wangenheim, C.; Borgatto, A. (2011) “Um Modelo de Avaliação de Jogos Educacionais na Engenharia de Software”. 25th Brazilian Symposium on Software Engineering (SBES)/São Paulo/Brazil.
- Thiry, M., Zoucas, A., & Gonçalves, R. Q. (2010) “Promovendo a Aprendizagem de Engenharia de Requisitos de Software através de um Jogo Educativo”. In Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE) (Vol. 1, No. 1).
- Valentim, M.O.S.V. (2002) “Brincadeiras infantis: importância para o desenvolvimento neuropsicológico”. Trabalho (Pós-graduação) Psicomotricidade, UCAM.
- Wangenheim, C.G; Wangenheim, A. (2012) “Ensinando Computação com Jogos”. Florianópolis: Bookess Editora.