

UNIVERSIDADE DO VALE DO RIO DOS SINOS – UNISINOS
UNIDADE ACADÊMICA DE GRADUAÇÃO
CURSO DE SISTEMAS DE INFORMAÇÃO

CÁSSIO LINDEN ALBERT

PROPOSTA DE GAMIFICAÇÃO NOS TESTES DE SOFTWARE

São Leopoldo
2019

Cássio Linden Albert

PROPOSTA DE GAMIFICAÇÃO NOS TESTES DE SOFTWARE

Artigo apresentado como trabalho final
da disciplina de Elaboração de Projetos
pelo Curso de Sistemas de Informação da
Universidade do Vale do Rio dos Sinos –
UNISINOS

Professor: Cristiano Bonato Both

São Leopoldo

2019

PROPOSTA DE GAMIFICAÇÃO NOS TESTES DE SOFTWARE

Cássio Linden Albert*

Cristiano Bonato Both**

Resumo:

Hoje, o nível de exigência em aplicações por parte dos usuários está cada vez maior, e os testes de software têm um papel fundamental neste processo. Várias técnicas são usadas pelos profissionais, mas a maioria delas são repetitivas, como, por exemplo, a análise de valor limite. Para mudar esse panorama, propõe-se o uso da gamificação, que traz elementos de jogos em contextos organizacionais, como entregas de projetos, por exemplo. As pesquisas nessa área são relativamente recentes, e está em constante crescimento, conforme indicado em pesquisas na base de dados do Scopus. Os resultados obtidos pelos pesquisadores são animadores, visto que há um indicativo de que a produtividade do profissional aumenta com o uso da gamificação, porém, sua implantação em grandes empresas pode ser difícil.

Palavras-chave: gamificação, testes, engenharia de software

1 INTRODUÇÃO

Com a facilitação do acesso à tecnologia, cada vez mais pessoas têm acesso a, ao menos, um smartphone. Consequentemente, mais aplicações estão sendo desenvolvidas e utilizadas com mais frequência. Com isso, a exigência de qualidade dos usuários também cresceu. Vendo esse cenário, os desenvolvedores de software estão colocando a qualidade de seu produto como uma de suas maiores prioridades, e o teste de software tem um papel fundamental neste processo.

Como é sabido, nenhum software está livre de problemas, e muitos detalhes que passam despercebidos pelos desenvolvedores, não passam no crivo de um bom testador. Para isso acontecer, técnicas dos mais variados tipos precisam ser postas em prática para o mínimo de problemas possível chegar ao usuário. Essas técnicas, em boa parte, são bastante repetitivas, o que pode acabar no desgaste e até tédio no profissional de testes. Para resolver isso, várias pesquisas relacionadas à aplicação da gamificação na área de testes têm sido feitas. A gamificação consiste em inserir elementos de jogos em contextos onde os mesmos não estão presentes. Para muitos pesquisadores, essa parece ser a solução para o tédio no trabalho, e um grande

* Aluno do curso de Sistemas de Informação. Email: cassiola@edu.unisinos.br

** Professor da Unisinos. Email: cboth@unisinos.br

incentivador na produtividade do profissional.

Neste artigo será proposto uma forma de atrair testadores manuais para a área de testes automatizados de forma intuitiva, mesmo que este profissional tenha pouca experiência com desenvolvimento de software, com o auxílio de elementos de gamificação.

2 FUNDAMENTAÇÃO TEÓRICA

Esta seção visa abordar alguns dos termos que serão importantes no entendimento da análise feita neste trabalho. Começa abordando a definição de gamificação, segue em suas subseções abordando a definição de teste de software, técnicas e níveis de teste, encerrando com uma breve definição sobre *framework*.

2.1 Gamificação

Gamificação é uma forma de inserir elementos de jogos num processo totalmente fora desse contexto (DALE, 2014). Ela está sendo usada para incentivar, motivar e melhorar a desempenho de uma empresa, ou para ajudar no ensino de algum tópico, por exemplo, deixando a atividade mais atrativa para as pessoas. Para (DE JESUS et al., 2018), “gamificação é um meio promissor para a resolução de problemas de teste” e “pode trazer prazer para o desenvolvimento de software.” Para (ELGRABLY; OLIVEIRA, 2018), uma das grandes vantagens do uso da gamificação na educação é “fornecer um sistema no qual os estudantes podem visualizar os efeitos de suas ações e aprendendo conforme a atividade acontece, deixando mais fácil o entendimento da relação das partes com o todo.”

2.1.1 Elementos e mecânicas

Alguns elementos a ser implementados são essenciais para considerar um processo gamificado. Entre os mais comuns estão: pontos, conquistas, níveis, missões, jornadas (um conjunto de missões), quadro de líderes, notificações (para motivar o jogador diante de uma ação correta) e mecânicas anti-jogo (para limitar até onde um

comportamento pode ser recompensado) (DALE, 2014).

Além desses elementos, deve ser considerada a motivação de cada pessoa, e ela pode ocorrer das seguintes maneiras:

- a expressão, onde as pessoas são motivadas e recompensadas por mostrar quem elas são (redes sociais são grandes exemplos disso);
- a competição, onde as pessoas disputam entre si (ou com elas mesmas);
- a exploração, onde as pessoas são motivadas por ter cada vez mais informações sobre algum conteúdo;
- e a colaboração, onde as pessoas são motivadas por se sentir parte do todo, através de projetos colaborativos;

Também é importante ressaltar que recompensas intrínsecas (como o aprendizado e contribuição para o todo), geralmente, contribuem mais para motivação de uma pessoa do que uma recompensa extrínseca (como dinheiro e selos, por exemplo). A próxima subseção irá tratar sobre testes de software, que é uma parte do fluxo de desenvolvimento de software que pode se beneficiar bastante da gamificação.

2.2 Teste de software

Há várias definições possíveis para teste de software, sendo que mais comumente pode ser citado como “um processo de executar um software com a intenção de encontrar defeitos” (MYERS, 1979), ou verificar se o software está fazendo o que deveria de acordo com seus requisitos (RIOS; MOREIRA, 2002). Para o ISTQB (*International Software Testing Qualification Board*), é “o processo que consiste em todas as atividades do ciclo de vida, tanto estáticas quanto dinâmicas, preocupado com a preparação, planejamento e avaliação de produtos de software e produtos de trabalho relacionados para determinar que eles atendem os requisitos especificados, para demonstrar que eles cumprem sua finalidade e para detectar defeitos.” (ISTQB, 2012)

2.2.1 Técnicas de teste

Entre as técnicas de teste é possível citar as funcionais e as estruturais. As técnicas funcionais, também conhecidas como teste da caixa-preta, são baseadas num software que já está implementado, em que o usuário (ou um testador) testa se as funcionalidades do programa condizem com os requisitos. Entre as empresas que têm uma estrutura de testes, esta é a técnica mais comum de ser utilizada, apesar de não ser a mais eficiente na resolução de certos tipos de bugs, por exemplo.

As técnicas estruturais, conhecidas como “caixa-branca”, são baseadas no código do software desenvolvido. Diferente das técnicas funcionais, esses testes são comumente feitos pelos programadores, e não testam as funcionalidades propriamente ditas. O mais comum de ocorrer é a verificação de comunicação entre os componentes, se o código segue boas práticas, se está bem otimizado, etc.

As técnicas baseadas em falhas, mais conhecidas como teste de mutação, consistem no desenvolvedor inserir erros no software propositalmente para verificar como ele irá se comportar, além de ser bastante utilizado pelos profissionais de teste para aprimorar casos de teste automatizados. Para esses casos, têm-se os mutantes, que são os programas modificados, e têm os casos de testes já desenvolvidos. Se esses casos de testes consigam detectar os mutantes, diz-se que este caso “matou” o mutante. Porém, nessa técnica, deve-se ter cuidado com os mutantes equivalentes, que são alterações em que o resultado da execução será o mesmo, com ou sem o equivalente, pois eles precisariam ser analisados a fim de verificar, de forma manual, se eles são ou não equivalentes (GUILHON, 2015). Isso elimina a praticidade de uma das formas mais práticas de detectar mutantes, que é através de testes automatizados.

2.2.2 Níveis de teste

Há três principais níveis de teste: de unidade, de integração e de sistema. Os testes de unidade fazem parte dos testes de caixa-branca, onde trechos específicos do código são testados, como métodos. Como esse tipo de teste se faz no código-fonte, ele deve ser feito pelo desenvolvedor. Tem como regra ser automatizado e reproduzível.

Os testes de integração também fazem parte dos testes de caixa-branca, onde se verifica se a comunicação entre todos componentes do software implementado está funcionando corretamente, ou até mesmo a integração entre softwares diferentes. É feito após os testes de unidade e, diferente dele, os testes de integração pode envolver a equipe de testes. Os testes de sistema testam toda a aplicação desenvolvida, simulando um usuário final em todos os aspectos: desde a forma de uso até o ambiente, que deve ser o mais próximo possível que o usuário final terá. A próxima subseção abordará de forma breve os *frameworks*.

2.2.3 Framework

Entende-se um *Framework* como uma solução única para problemas que são semelhantes, sendo implementada de forma diferente em cada problema, visto que ele deve ser flexível e extensível para isso. No contexto de desenvolvimento de software, o *framework* é uma aplicação quase completa, cujos os “pedaços” faltantes serão implementados pelo desenvolvedor que utilizará o *framework* como base para sua aplicação (SAUVÉ, 2000). A próxima seção irá abordar os trabalhos relacionados com a proposta deste artigo e como se chegou neles.

3 TRABALHOS RELACIONADOS

3.1 Metodologia de pesquisa

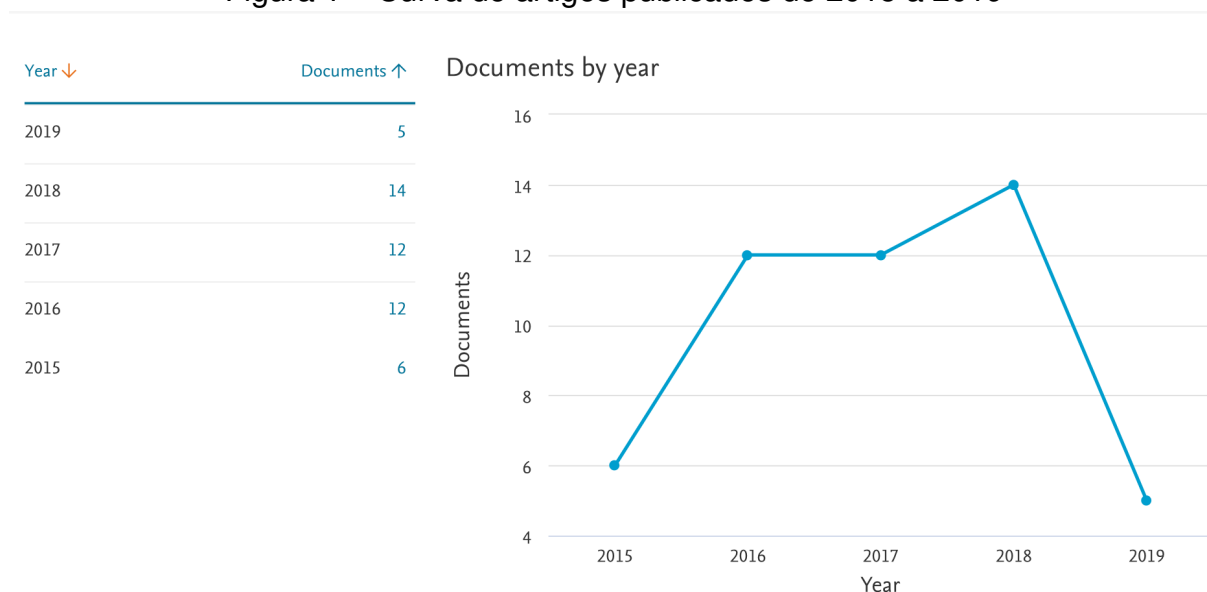
Para começar esse trabalho, foi feita uma pesquisa sistemática, cujo objetivo foi confirmar que a gamificação nos testes de software é um tema relevante. Para isso, utilizou-se a ferramenta Scopus, que é um grande repositório de citações e resumos revisados por especialistas das mais diversas áreas, oferecendo várias estatísticas e gráficos. Para os artigos utilizados nesse trabalho, os seguintes termos foram pesquisados:

(*TITLE-ABS-KEY (gamification) AND TITLE-ABS-KEY (software AND testing)*)AND (*LIMIT-TO (PUBYEAR , 2019) OR LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO(PUBYEAR , 2017) OR LIMIT-TO (PUBYEAR , 2016) OR LIMIT-TO (PUBYEAR*

,2015)) AND (LIMIT-TO (SUBJAREA , "COMP"))

Essa *query* de pesquisa filtra artigos publicados entre 2015 e 2019 que envolvem termos estritamente relacionados a gamificação e testes de software, limitando-se à área da computação. Em 9 de abril de 2019, a *query* retornou 49 resultados. A figura 1 traz um gráfico mostrando o número de artigos publicados desde 2015. Considerando que a pesquisa foi feita no início de 2019, essa curva tem uma leve queda, mas a tendência de crescimento nesta área é grande.

Figura 1 – Curva de artigos publicados de 2015 a 2019



Fonte: Scopus

Para selecionar os artigos que serão abordados nessa pesquisa, foram verificados os títulos de cada artigo retornado pela *query* pesquisada no Scopus que tivesse relação direta com o tópico desejado, passando pela leitura do resumo de cada um e a verificação da disponibilidade dos mesmos na base de dados da UNISINOS. Após a leitura dos artigos, os mesmos foram divididos em três categorias: Estudo, *Framework* e Jogo. Esta análise resultou na tabela 1:

Tabela 1 – Artigos selecionados

Autores	Título	Ano	Categoria
Fraser G.	Gamification of software testing	2017	Estudo
De Jesus G.M., Ferrari F.C., De Paula Porto D., Fabbri S.C.P.F.	Gamification in software testing: A characterization study	2018	Estudo
Elgrably I.S., Oliveira S.R.B.	Gamification and evaluation of the use the agile tests in software quality subjects: The application of case studies	2018	Estudo
Pedreira O., García F., Brisaboa N., Piattini M.	Gamification in software engineering - A systematic mapping	2015	Estudo
Parizi R.M.	On the gamification of human-centric traceability tasks in software testing and coding	2016	Framework
García F., Pedreira O., Piattini M., Cerdeira- Pena A., Penabad M.	A framework for gamification in software engineering	2017	Framework
Houshmand M., Paydar S.	EMVille: A gamification-based approach to address the equivalent mutant problem	2017	Framework
Rojas J.M., Fraser G.	Code Defenders: A Mutation Testing Game	2016	Jogo

Fonte: elaborada pelo autor

Na subseção “discussão dos trabalhos”, os artigos filtrados serão brevemente abordados, considerando suas respectivas categorias.

3.2 Discussão dos trabalhos

3.2.1 Estudo

(FRASER, 2017) elenca alguns problemas no âmbito do desenvolvimento de software que testes feitos de forma inadequada podem causar, e explora uma forma de melhorar o trabalho do testador: a gamificação. Ele afirma que o uso dessa técnica pode ajudar no trabalho em equipe e no ensino de testes de software nas escolas, além de melhorar a qualidade dos testes.

(DE JESUS et al., 2018) apresentaram um cenário geral de outros artigos que abordam gamificação nos testes de software. Esses artigos foram aplicados na indústria

e na educação; os elementos abordados foram pontos, emblemas e quadro de líderes; os objetivos eram o desenvolvimento da criatividade, colaboração, engajamento, motivação e satisfação; as técnicas de teste citadas foram a funcional, estrutural e baseados em falhas, sendo a última a mais utilizada; os níveis de teste elencados foram os de unidade, integração e de sistema. Apesar de estarem focado nos testes, os autores afirmam que praticamente todos os artigos estudados por eles podem ser aplicados em todo o processo, com exceção do planejamento e na configuração de dados e ambiente.

(ELGRABLY; OLIVEIRA, 2018) trouxeram um estudo feito com estudantes de graduação e pós-graduação relacionado ao ensino de testes ágeis nas disciplinas de Qualidade de Software ensinadas nessas modalidades de ensino. A motivação dos autores é que o ensino de métodos ágeis só faz sentido para o aluno se ele viver aquilo na prática. Como elementos de gamificação, os autores também utilizaram pontos, emblemas e quadro de líderes, além de premiações.

(PEDREIRA et al., 2015) mostraram um panorama geral de gamificação no desenvolvimento de software. Os autores afirmam que, junto com as áreas de desenvolvimento e requisitos, a área de testes é bastante adequada para a técnica. Assim como afirmado por (FRASER, 2017), os autores verificaram que a gamificação ajuda no trabalho em equipe. Os elementos abordados foram os pontos, emblemas, quadro de líderes e jornadas. Eles também afirmaram que a gamificação deve ser aplicada com o maior número de elementos possível (além da pontuação) para ela não cair em desuso no decorrer no tempo.

3.2.2 *Framework*

(PARIZI, 2016) trouxe um problema de rastreabilidade entre testes e o código-fonte, e propõe um *framework* chamado GamiTracify para engajar e motivar os desenvolvedores nas tarefas de rastreabilidade. Os elementos de gamificação utilizados aqui estão os pontos, quadro de líderes, avatar e jornadas.

(GARCÍA et al., 2017) propuseram o GOAL (Gamification on Application Lifecycle Management), um *framework* que possibilita aplicar a gamificação em qualquer fase do desenvolvimento de software. Os autores testaram o *framework* numa empresa

nas áreas de gerenciamento de requisitos, gestão de projetos e teste de software (onde foi mencionado apenas os testes unitários como objeto de estudo) e aplicaram os elementos de pontos, níveis, distintivos, quadro de líderes, gráfico social e desafio. O objetivo é de engajar usuários e melhorar seu desempenho, que foi alcançado, segundo os autores.

(HOUSHMAND; PAYDAR, 2017) mostraram um *framework* para combater os mutantes equivalentes de forma eficiente. A ideia dos autores envolve o uso de *machine-learning* com a gamificação para detectar mutantes equivalentes de forma automática. Os elementos utilizados foram os pontos e jornadas.

3.2.3 Jogo

(ROJAS; FRASER, 2016) trouxeram o desenvolvimento de um jogo chamado *Code Defenders*, cujo objetivo é desenvolver e ensinar testes de mutantes. Nele, dois jogadores, sendo um atacante e um defensor, através da técnica de testes de mutantes, devem melhorar tanto seus casos de teste quanto o próprio código-fonte. O jogo é bastante promissor, e pode trazer ainda mais pessoas para a área, tanto de desenvolvimento quanto para testes.

4 METODOLOGIA

A metodologia proposta irá focar numa forma amigável de ensinar testadores manuais a como projetar casos de testes automatizados através de um jogo integrado com a extensão Selenium IDE, disponível para navegadores web modernos como o *Google Chrome* e o *Mozilla Firefox*. A extensão disponibiliza uma API de integração, que entre suas funcionalidades está o envio dos comandos que serão executados pelo Selenium, além da extração desses comandos em forma de código Java, já revelando o primeiro pré-requisito para o usuário começar a jogar. O segundo requisito é a utilização de um dos navegadores web citados, com a extensão Selenium IDE instalada.

O jogo terá foco em ensinar o uso da ferramenta Selenium. A primeira etapa funcionará como um tutorial, e passará pela detecção de elementos básicos usados na maioria dos sites existentes, como campos de texto, caixas de seleção e rádio, avi-

sos para o usuário e botões. O usuário usará um site feito exclusivamente para ele aprender a detecção dos elementos do mesmo. A ferramenta dará a possibilidade de selecionar algum elemento específico, e ela mostrará formas de detectar aquele elemento através do Selenium, juntamente com o código Java usado para chegar naquele resultado. Ao final dessa etapa, o usuário passará por um desafio para exercitar o que foi passado durante o tutorial. Ao concluir o desafio, o usuário ganhará um selo chamado: “Começando com o pé direito!”.

A etapa seguinte começa com um tutorial mais detalhado, mostrando como fazer assertivas com os elementos previamente apresentados. Ele funcionará de forma similar ao tutorial da primeira etapa, com a diferença de que o usuário terá que encontrar o elemento desejado através do próprio navegador, aumentando a dificuldade.

Na terceira e última etapa, o usuário será mostrado a um site mais detalhado, e deverá desenvolver o código de forma independente, sem ajuda direta da ferramenta. Caso ele precisar, poderá pedir até três dicas de como completar o código; porém, se ele completar o desafio sem pedir as dicas, ganhará um bônus.

Inicialmente, todos os jogadores finalizarão o jogo com a mesma pontuação, mas haverá algumas exceções. Todas as etapas terão um tempo mínimo para serem concluídas; isso será feito para que o usuário não seja motivado a procurar uma solução pronta na internet. Caso o desafio seja respondido muito rapidamente, o jogo descontará alguns pontos do usuário. Futuramente, será pensado em outras formas de recompensar o usuário, ou puni-lo caso ele tente algum tipo de trapaça.

Para avaliar a eficácia dessa metodologia, será feita uma comparação entre dois grupos de cinco testadores cada: quem aprendeu a usar o Selenium através de um curso convencional e quem aprendeu através da metodologia gamificada, com o uso do plugin desenvolvido. A avaliação será feita através do código de teste desenvolvido, e os quesitos considerados serão o tempo para resolver um determinado desafio e a qualidade do código desenvolvido (de acordo com as ferramentas que o Selenium oferece). Ao final, será entregue um questionário aos integrantes do grupo que usou a metodologia gamificada, com perguntas sobre a usabilidade da ferramenta e um espaço para deixar um *feedback* sobre ela.

5 CONCLUSÃO

Apesar de ser relativamente recente, a gamificação é um tema que está em constante crescimento na engenharia de software. Conforme observado, ela tem apresentado bons resultados, tanto na indústria quanto no ensino, e tem chegado no seu propósito de incentivar, motivar e melhorar o desempenho de funcionários, porém, a aplicação dessa metodologia em empresas reais aparenta ser bem difícil, principalmente em empresas maiores que já têm culturas e processos bem definidos.

Em trabalhos futuros tem-se a ideia de aprofundar ainda mais no tema de gamificação, focando em todo o processo de testes de software, e elaborar um plano de aplicação dessa metodologia em empresas de pequeno porte, onde os processos ainda não estão bem definidos.

REFERÊNCIAS

- DALE, S. Gamification: making work fun, or making fun of work? **Business Information Review**, [S.l.], v. 31, n. 2, p. 82–90, 2014. Cited By :48.
- DE JESUS, G. M. et al. Gamification in software testing: a characterization study. In: ACM INTERNATIONAL CONFERENCE PROCEEDING SERIES, 2018. **Anais...** [S.l.: s.n.], 2018. p. 39–48.
- ELGRABLY, I. S.; OLIVEIRA, S. R. B. Gamification and evaluation of the use the agile tests in software quality subjects: the application of case studies. In: ENASE 2018 - PROCEEDINGS OF THE 13TH INTERNATIONAL CONFERENCE ON EVALUATION OF NOVEL APPROACHES TO SOFTWARE ENGINEERING, 2018. **Anais...** [S.l.: s.n.], 2018. v. 2018-March, p. 416–423.
- FRASER, G. Gamification of software testing. In: IEEE/ACM 12TH INTERNATIONAL WORKSHOP ON AUTOMATION OF SOFTWARE TESTING, AST 2017, 2017., 2017. **Proceedings...** [S.l.: s.n.], 2017. p. 2–7. Cited By :7.
- GARCÍA, F. et al. A framework for gamification in software engineering. **Journal of Systems and Software**, [S.l.], v. 132, p. 21–40, 2017. Cited By :19.
- GUILHON, R. J. Geração automática de suíte de teste para gui a partir de rede de petri. **PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO - PUC-RIO**, [S.l.], p. 70–77, Apr 2015.
- HOUSHMAND, M.; PAYDAR, S. Emville: a gamification-based approach to address the equivalent mutant problem. In: INTERNATIONAL CONFERENCE ON COMPUTER AND KNOWLEDGE ENGINEERING, ICCKE 2017, 2017., 2017. **Anais...** [S.l.: s.n.], 2017. v. 2017-January, p. 326–332.
- ISTQB, I. S. T. Q. B. **Standard glossary of terms used in software testing**. 2012. v. Version 2.2.
- MYERS, G. J. **Art of software testing**. New York, NY, USA: John Wiley & Sons, Inc., 1979.
- PARIZI, R. M. On the gamification of human-centric traceability tasks in software testing and coding. In: IEEE/ACIS 14TH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING RESEARCH, MANAGEMENT AND APPLICATIONS, SERA 2016, 2016., 2016. **Anais...** [S.l.: s.n.], 2016. p. 193–200. Cited By :11.
- PEDREIRA, O. et al. Gamification in software engineering - a systematic mapping. **Information and Software Technology**, [S.l.], v. 57, n. 1, p. 157–168, 2015. Cited By :116.
- RIOS, E.; MOREIRA, T. **Teste de software**. [S.l.: s.n.], 2002.
- ROJAS, J. M.; FRASER, G. Code defenders: a mutation testing game. In: IEEE INTERNATIONAL CONFERENCE ON SOFTWARE TESTING, VERIFICATION AND VALIDATION WORKSHOPS, ICSTW 2016, 2016., 2016. **Proceedings...** [S.l.: s.n.], 2016. p. 162–167. Cited By :11.

SAUVÉ, J. P. **O que são frameworks?** Disponível em:
<http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/frame/oque.htm>.
Acessado em 22 abr. 2019.