



## **BENEFÍCIOS DA AUTOMAÇÃO DE TESTES COM O USO DA FERRAMENTA KATALON STUDIO<sup>1</sup>**

### **BENEFITS OF TEST AUTOMATION WITH THE USE OF THE KATALON STUDIO TOOL**

**Ieda Rosana Kolling Wiest<sup>2</sup>, André Fernando Rollwagen<sup>3</sup>, Joseane Amaral<sup>4</sup>, Maikon Cismoski dos Santos<sup>5</sup>, Ricardo Vanni Dallasen<sup>6</sup>, Vanessa Lago Machado<sup>7</sup>**

<sup>1</sup>Monografia apresentada ao Curso de Tecnologia em Sistemas para Internet do Instituto Federal de Educação, Ciência e Tecnologia Sul-Rio-Grandense, câmpus Passo Fundo.

<sup>2</sup>Tecnóloga em Sistemas para Internet do IFSUL – câmpus Passo Fundo.

<sup>3</sup>Orientador Professor Me. do Curso de Tecnologia em Sistemas para Internet do IFSUL – câmpus Passo Fundo.

<sup>4</sup>Professora Dra. do Curso de Tecnologia em Sistemas para Internet do IFSUL – câmpus Passo Fundo.

<sup>5</sup>Professor Me. do Curso de Tecnologia em Sistemas para Internet do IFSUL – câmpus Passo Fundo.

<sup>6</sup>Professor Me. do Curso de Tecnologia em Sistemas para Internet do IFSUL – câmpus Passo Fundo.

<sup>7</sup>Doutoranda em Ciência da Computação na UFSC – câmpus Florianópolis.

## **RESUMO**

O tema proposto neste trabalho está ligado à melhoria da qualidade dos sistemas especificamente no processo de testes manuais e dos diferentes tipos de testes funcionais. A automação de teste é importante, pois em alguns casos reduz tempo e trabalho dos analistas, onde estes irão dedicar maior tempo em análises mais complexas e com maior valor agregado, deixando os testes mais simples para serem realizados automaticamente. Uma falha em um sistema pode gerar muitos prejuízos para uma organização, dependendo do produto e de sua abrangência. O objetivo deste trabalho foi avaliar os ganhos e dificuldades que o processo de automação de testes funcionais em aplicações web possui. Realizar a automação de diferentes tipos de testes funcionais no sistema de inscrição de eventos do Instituto Federal Sul-Rio-Grandense – IFSUL Câmpus Passo Fundo. Para automação do processo de testes de software utilizou-se a ferramenta Katalon Studio. Por fim, realizou-se uma comparação entre os testes manuais e os testes automatizados mostrando através dessa execução as vantagens e possíveis desvantagens na automatização em cada um dos testes executados.

**Palavras-chave:** Automação de testes. Katalon Studio. Testes funcionais.

## **ABSTRACT**

The theme proposed in this work is linked to the improvement of the quality of the systems specifically in the process of manual tests and of the different types of functional tests. Test automation is important, as in some cases it reduces the time and work of analysts, where they will dedicate more time to more complex analyzes and with greater added value, making tests



simpler to be performed automatically. A failure in a system can generate millions in losses for an organization, depending on the product and its scope, these failures are unacceptable, as they can be fatal. The objective of this work was to evaluate the gains and difficulties that the process of automation of functional tests in web applications has. Perform the automation of different types of functional testing in the event registration system of the Instituto Federal Sul-Rio-Grandense – IFSUL CâmpusPasso Fundo. To automate the software testing process, the Katalon Studio tool was used. Finally, a comparison was made between manual tests and automated tests, showing through this execution the advantages and possible disadvantages of automation in each of the tests performed.

**Keywords:** Test automation. Katalon Studio. Functional tests.

## INTRODUÇÃO

A qualidade dos sistemas desenvolvidos é uma preocupação constante nos times de desenvolvimento de software. Teste de software é uma área que tem crescido muito nos últimos tempos, principalmente a automação desses testes.

A (NBR ISO 9000:2005) define que a "qualidade é o grau no qual um conjunto de características inerentes satisfaz aos requisitos" e ainda pode-se afirmar que se algum produto ou serviço atende aos requisitos especificados, este mesmo produto ou serviço possui a qualidade que foi esperada para ele.

A qualidade pode ser medida através do grau de satisfação em que as pessoas avaliam determinado produto ou serviço. Contudo, esse produto ou serviço pode ter qualidade para algumas pessoas e para outras nem tanto, ou seja, a qualidade é subjetiva (NBR ISO 9000:2005).

Quando se fala em qualidade de software, Pressman (2016) recomenda que ela seja de fato implementada e não somente uma ideia ou desejo que uma organização tenha para seus projetos. Ainda segundo o autor, a qualidade de software consiste, em um conjunto de requisitos, por um produto que atende as necessidades dos clientes, cumprindo todos os requisitos solicitados da melhor forma possível.

Segundo Bastos et. al (2012), até pouco tempo eram realizados somente os testes unitários, em que os testes eram executados pelos próprios programadores. Por conta disso, muitos problemas não eram encontrados. Estes problemas apareciam quando o sistema já estava em ambiente de produção, onde a correção tem seu custo elevado e a credibilidade do



sistema abalada.

Segundo o guia *Software Engineering Body of Knowledge* (SWEBOK, 2013), os testes são uma atividade desempenhada para avaliação da qualidade do produto e para melhorar, identificando defeitos e problemas.

Um programa é testado com o propósito de encontrar defeitos a serem corrigidos ou minimizados. Cada cenário de teste requer um resultado esperado e a partir da definição de “certo” se determina se o programa está em conformidade com as especificações e se ele serve ao seu propósito (TSUI; KARAM, 2013).

Em geral, testar um sistema é um trabalho que exige muito esforço. O tempo de execução dessas tarefas pode ser encurtado por meio do uso de ferramentas apropriadas. Um analista de testes pode usar uma variedade de ferramentas, como por exemplo, um software que executa um script automaticamente a partir de um caso de teste gerado, sem necessidade de repetição manual ou um gerador de massa de dados de teste. Essas ferramentas são úteis para aumentar a eficiência e eficácia dos testes (NAIK; TRIPATHY, 2008).

Ao entrar na definição de qualidade, percebe-se quanto complexa ela é, e os projetos de desenvolvimento de softwares podem não alcançar resultados satisfatórios quando se trata do ciclo de vida do software e a qualidade do sistema. Um dos grandes desafios vem sendo conseguir cumprir prazos e custos estabelecidos e alcançar a qualidade desejada.

Assim, diante disso tem-se a seguinte questão de pesquisa: Quais são os benefícios da automação de diferentes tipos de testes e o impacto ao escolher uma ferramenta ao invés de realizar testes manuais e quando a automatização se torna vantajosa?

Pretende-se, portanto, a partir do desenvolvimento de um roteiro de diferentes tipos de testes automatizados aplicar em um sistema já existente e por fim avaliar quando é vantagem automatizar cada um dos tipos executados.

Em consequência disso, este trabalho justifica-se pela importância do processo de automação de testes funcionais de software, os quais, quando aplicados corretamente, podem trazer inúmeros benefícios para o desenvolvimento de um software otimizando seu tempo e melhorando sua qualidade.

O objetivo deste trabalho é avaliar a importância da automação de teste funcional em diferentes tipos de testes. Demonstrar os benefícios da automação com uso da ferramenta



Katalon Studio no sistema de Eventos do Instituto Federal de Educação, Ciência e Tecnologia Sul-Rio-Grandense - Câmpus Passo Fundo, e fazer uma comparação dos métodos manuais de teste, mostrando como a ferramenta pode contribuir para obtenção da qualidade final esperada do produto.

## REFERENCIAL TEORICO

Para o desenvolvimento deste trabalho foi necessário estudar alguns conceitos e tecnologias que serviram como base para serem empregadas no projeto.

### Teste de software

Para Pressman (2016), teste de software é um grupo de atividades construídas e aplicadas metodicamente antes mesmo do desenvolvimento do software estar concluído, com o propósito de verificar se o produto realiza as ações requisitadas pelo cliente, desde sua interface e layout até mesmo se o código fonte foi implementado corretamente.

*O teste é destinado a mostrar que um programa faz o que é proposto a fazer e para descobrir os defeitos do programa antes do uso. Quando se testa o software, o programa é executado usando dados fictícios. Os resultados do teste são verificados à procura de erros, anomalias ou informações sobre os atributos não funcionais do programa (SOMMERVILLE, 2011, p. 144).*

Segundo Sommerville, geralmente o sistema de software comercial tem de passar por três estágios de teste.

1. Teste em desenvolvimento, em que o sistema é testado durante o processo de desenvolvimento para descobrir *bugs* e defeitos.
2. Teste de release, onde uma equipe de teste independente testa uma versão completa do sistema antes do software ser liberado ao usuário verificando assim se atende aos requisitos definidos.
3. Teste de usuário ou aceitação, em que os usuários testam o sistema em seu próprio ambiente.

Nos próximos tópicos são abordados os conceitos sobre os tipos de testes de



software.

## Tipos de Teste

Segundo o ISTQB (International Software Testing Qualifications Board) (SYLLABUS CTFL-2018) os testes podem ser classificados em função:

1. Testes durante o ciclo de vida do produto.
  - Teste de integração: Caracteriza-se em testar tanto testes de integração entre componentes quanto de integração com outros sistemas.
  - Teste de sistema: Trata o comportamento de todo do sistema definido no levantamento de requisitos. Abrange os requisitos funcionais e não funcionais. Neste tipo de teste o ambiente de teste deve ser o mais semelhante possível ao ambiente de produção, a fim de potencializar a identificação de falhas específicas de ambiente.
  - Teste de aceite: Muitas vezes também chamado como teste beta ou teste de campo é realizado com possíveis clientes ou usuários antes do software ser disponibilizado para comercialização ou para uso efetivo.
  - Teste de manutenção: Este teste é realizado quando sistemas ficam por muito tempo em funcionamento e onde são realizadas pequenas modificações de emergência ou há migrações e integrações com outros sistemas.

## Testes conforme o objetivo.

- Teste funcional: Esse tipo de teste é caracterizado por possuir o objetivo de validar se as funcionalidades disponíveis pelo software estão de acordo com o especificado pelo cliente na documentação do produto, por exemplo, requisitos de negócios e requisitos técnicos. O teste funcional considera o comportamento externo do software (teste de caixa preta).
- Testes não funcionais: Testam as qualidades técnicas do sistema, por exemplo, testes de carga, teste de segurança. Os testes não funcionais podem ser realizados em todos os níveis de teste.
- Teste estrutural: É o teste que visa validar a estrutura desenvolvida na parte do



código, como no uso da ferramenta JUnit para testar classes ou métodos desenvolvidos em Java.

- Teste de regressão: O teste de regressão é realizado quando é alterado ou quando tem a inclusão de alguma funcionalidade nova em um sistema já em funcionamento, onde na maioria das vezes é necessário retestar o sistema como um todo para verificar se as funcionalidades não foram impactadas.

## Automação de Teste

Segundo Sommerville (2011), na automatização os casos de testes e seus cenários são todos codificados no software que a empresa estiver utilizando e será realizada toda vez que for feita uma mudança na aplicação. Ainda enfatiza que a automatização tem aumentado significativamente e destaca que os testes nunca poderão ser totalmente automatizados, pois o teste automatizado só irá fazer o que lhe foi proposto.

Muitas empresas possuem testadores para executar os testes manualmente, o que implica a necessidade de uma ou mais pessoas dedicando, planejando, efetuando e validando os resultados todas as vezes que houver essa necessidade. Ao fazer com o uso de um processo automatizado essas tarefas seriam necessárias apenas uma vez e quando surgir novamente à necessidade, os testes automatizados podem ser executados para gerar as validações, além de que o teste manual abre margens para erros, o que poderia ser evitado utilizando o processo automatizado (KUMAR; MISHRA, 2016).

Na hora de escolher entre teste manual e teste automatizado, de acordo com Gomes (2017) é preciso avaliar as características do cliente e do projeto, o tipo de orçamento disponível e qual é a metodologia de teste que beneficiará os prazos estabelecidos, as expectativas dos clientes e assim por diante. Destaca ainda de quando se trata de testes exploratórios e usabilidade, a melhor maneira de testar é através dos testes manuais. Quando se trata dos testes de desempenho, testes de carga, testes de regressão e testes repetidos, a melhor maneira é o teste automatizado, pois poderá ganhar no tempo de execução sendo que nestes tipos de teste o mesmo precisa ser repetido muitas vezes, e o caso de teste não tem mudança significativa.



## METODOLOGIA

Neste capítulo são descritas as etapas de desenvolvimento deste projeto, considerando a melhora na execução dos testes aplicados de forma automatizada. Assim, a execução de testes criada e executada em uma ferramenta de automação chamada Katalon Studio é proposta, mostrando ao final, o ganho de tempo para futuros testes de regressão na mesma funcionalidade. Desta forma, considera-se primeiramente a execução de forma manual, onde a ferramenta citada possibilita a gravação dos passos realizados pelo usuário em tempo real, reproduzindo-os em forma de código e convertendo em um caso de teste, sendo possível executá-lo de forma automática posteriormente.

Primeiramente foi realizado o planejamento dos testes onde foi criado um Plano de Teste, para diferentes tipos de teste. Este foi elaborado com base na análise de necessidades básicas e fluxos básicos do sistema de inscrição de eventos do Instituto Federal de Educação, Ciência e Tecnologia Sul-Rio-Grandense, câmpus Passo Fundo.

Neste planejamento foi definido o escopo de teste, onde foram testados, os requisitos funcionais do sistema, ou seja, foi delimitada a execução dos fluxos principais.

A execução dos testes foi realizada pela ferramenta Katalon Studio, uma ferramenta gratuita de automação de teste tanto web quanto mobile, possui uma IDE própria a base da linguagem Java, uma solução com um conjunto abrangente e integrado de recursos, desde a gravação das etapas de teste e geração de scripts até a execução e a geração de relatórios dos resultados dos testes. Tal ferramenta foi escolhida justamente por propiciar facilidade no uso e também por oferecer uma interface amigável e de fácil entendimento tanto na criação dos testes bem como na sua execução. A qual é baseada em tecnologia open source, organiza os testes em projetos e possui relatórios precisos e detalhados, permitindo também uma customização bem ampla para suas informações.

A automação e a montagem de um teste automatizado foram realizadas de forma manual no primeiro teste e a partir deste realizada a gravação dos passos. Para isso se fez uso da opção *Record* do Katalon. Com o *Record Web* basta inserir a URL desejada e clicar no ícone do navegador, fazendo com que uma página da web seja aberta na URL especificada. Após isso, qualquer ação efetuada neste navegador é registrada como uma instrução



(*Keyword*) do Katalon resultando em um script de teste automatizado ao fechar a janela, possibilitando assim a execução automática do mesmo teste posteriormente.

No final da execução dos testes foram apresentadas soluções para problemas que acontecem na validação e execução de testes manuais de um sistema, fazendo um comparativo de tempo gasto com o teste manual e o mesmo teste sendo automatizado. Mostrou-se a possibilidade de avaliar as funcionalidades de um sistema de maneira mais rápida eficiente e sem muito esforço, ou seja, esse processo é caracterizado pela facilidade de repetir esforços contínuos e repetidos para testar um sistema, além disso, mostrar que é possível um retorno mais rápido da equipe de testes, Dessa forma, os *bugs* ocasionais podem ser resolvidos mais rapidamente.

Também realizou-se uma avaliação sobre as vantagens na execução de testes manuais e automatizados, elencando as melhores escolhas. Ainda apresentou-se a possibilidade de ganho de eficiência, pois testes manuais costumam demandar grande quantidade de tempo do ciclo de desenvolvimento de um sistema. Em consequência disso, as melhorias mais simples podem fazer uma grande diferença no custo final de um projeto.

## RESULTADOS

Neste projeto foram executados diferentes tipos de teste onde cada um teve seus resultados, os quais foram descritos nesta seção.

Primeiramente foi realizada a execução de casos de Teste Baseado em Dados, onde o objetivo foi buscar dados de uma tabela para realizar testes dinâmicos, repetitivos e automáticos para web onde foi utilizado recurso objeto *Data File*, do Katalon Studio que pode consultar dados de fontes externas: arquivo CSV, arquivo ".xls", banco de dados relacional. Também suporta testes baseados em dados e objeto de arquivo de dados, como resultados apontam-se: o tempo de toda execução automatizada em 1m - 14,447s, esse tempo poderá variar dependendo da quantidade de execuções, geração automática de relatórios que facilitam a verificação de possíveis erros caso falhar, a precisão é maior do que a executada manualmente.

Outro tipo de teste executado foi de Teste de Segurança, acesso com diferentes perfis, onde se verificou o acesso dos perfis administrativo, docente e aluno a fim de validar as





devidas permissões, onde obteve-se como resultado ganho no tempo de execução, que foi de 25,636s. A geração automática de relatórios, por meio da automação de testes, facilita a verificação de possíveis erros caso hajam falhas e garantia maior na qualidade, pois, o risco em esquecer algum dos requisitos, tende a ser menor quando já estão automatizados, considerando o trabalho exaustivo na execução de testes manuais.

Para os tipos de teste de Testes Funcionais foram executados alguns testes referentes as funcionalidades básicas como, por exemplo, registro de presença em evento a partir de dados de uma tabela onde o objetivo deste teste foi medir a qualidade funcional dos requisitos principais do sistema de eventos. Ao executar o teste automatizado foram registradas presenças para todos os códigos vindos da tabela, para um evento também previamente configurado sem a necessidade de registrar individualmente de forma manual para cada usuário, onde novamente obteve ganho de tempo e agilidade em um processo que seria realizado de forma manual. O tempo depende da quantidade de vezes ou de dados que o caso de teste é executado, como neste caso foram dois códigos, o tempo foi de 35,438s.

Por fim foi executado o teste de compatibilidade onde o objetivo foi realizar teste funcional de um caso de teste de função a fim de validar sua compatibilidade em diferentes navegadores, ou seja, validar se a navegação e a execução do teste funcional suportam diversos navegadores de Internet. Para isso, foram escolhidos três navegadores popularmente conhecidos e utilizados - Google Chrome, Mozilla Firefox e Microsoft Edge. Entretanto é possível executar em qualquer navegador, os resultados foram: para o tempo execução no Google Chrome: 25,211s, no Mozilla Firefox : 27,376s, e no Microsoft Edge: 26,351s, sem intervenção e a necessidade de repetir os testes para cada navegador.

## CONSIDERAÇÕES FINAIS

O presente estudo teve como objetivo avaliar a importância da automação de teste funcional em diferentes tipos de testes. Demonstrar os benefícios da automação de testes no sistema de Eventos do Instituto Federal de Educação, Ciência e Tecnologia Sul-Rio-Grandense - Câmpus Passo Fundo, e fazer uma comparação com métodos manuais de teste.

Para início do desenvolvimento foi necessário construir um plano de testes, para



elencar os diferentes tipos de teste que seriam executados e, para execução e automação deste trabalho foi escolhida a ferramenta de automação de testes chamada Katalon Studio.

Ao finalizar a execução dos testes manuais e automatizados nos tipos de testes levantados e descritos anteriormente conclui-se que, obtém redução do tempo de execução, e que a automatização é adequada quando há casos de teste que precisam ser executados repetidamente por um longo período de tempo. Neste contexto, são mais confiáveis, pois as ferramentas e os scripts realizam o teste automatizado, possibilitando ainda que outros testadores possam executar os mesmos testes e visualizar os resultados progressivamente.

Por fim, destaca-se a importância de avaliar a real necessidade de automatizar testes de software, pois quando há necessidade de identificar problemas relacionados à aparência visual do sistema e problemas de usabilidade os testes automatizados não conseguem identificar essas lacunas.

É importante citar que a ferramenta pode não conseguir validar alguns campos ao gravar os passos para ser feita a automação, fato que prejudica a execução no desenvolvimento do teste e, conseqüentemente demora mais que um teste manual, sendo necessária a alteração do script.

Como trabalhos futuros é sugerida a continuação desse estudo ampliando para outros tipos de testes não funcionais, como testes de desempenho, integrações com outras ferramentas de teste e testes de serviço, mobile, desktop, ou seja, explorar outras possibilidades que a ferramenta dispõe.

## REFERÊNCIAS BIBLIOGRÁFICAS

BASTOS, A., RIOS, E., CRISTALLI, R., e MOREIRA, T., Base de conhecimento em teste de software (Rio de Janeiro: Alta Books, 2012).

GOMES, Medeiros, Marcus. Teste manual vs automação de tester. 2017. Disponível em: <https://medium.com/@mvmgmg/teste-manual-vs-automa%C3%A7%C3%A3o-de-teste-8fcf4f9832b1>. Acesso em: 10 mar. 2021.

KUMAR, D.; MISHRA, K. The impacts of test automation on software's cost, quality and time to market. *Procedia Computer Science*, Elsevier, v. 79, 2016.

MARQUES DA COSTA, CAMILA. Caso De Uso: Integração De Ferramentas Para Automação De Testes Funcionais De Software. Disponível em:



<https://painel.passofundo.ifsul.edu.br/uploads/arq/20190221150108727886174.pdf>. Acesso em: 20 abr. 2021.

NBR ISO 9000:2005. Sistemas de Gestão da Qualidade Fundamentos e vocabulário. Rio de Janeiro.

NAIK, K.; TRIPATHY, P. SOFTWARE TESTING AND QUALITY ASSURANCE.[S.l.]: Wiley, 2008.

PRESSMAN, Roger; MAXIM, Bruce. Engenharia de Software: 8ª Ed., São Paulo: McGraw-Hill, 2016.

SWEBOK2013. Disponível em: <<https://ieeecsmedia.computer.org/media/education/swebok/swebok-v3.pdf>>. Acesso em: 08 abr 2021.

TSUI, Frank; KARAMAN, Orlando. Fundamentos da Engenharia de Software. Tradução e Revisão de Edson Tanaka. 2.ª Edição. Rio de Janeiro: LTC, 2013.

SOMMERVILLE, Ian. Engenharia de Software. 9ª Ed., São Paulo: Addison Wesley, 2011.

SYLLABUS-CTFL, 2018. Disponível em:<[https://www.bstqb.org.br/uploads/syllabus/syllabus\\_ctfl\\_2018br.pdf](https://www.bstqb.org.br/uploads/syllabus/syllabus_ctfl_2018br.pdf)>. Acesso em: 09 abr 2020.