

TDD E LARAVEL¹ PARA O ENSINO DE PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

Ruana Richelly da Silva Marques²
Sebastião Fellipe Pinto Lopes³
Clara Leticia Lopes de Oliveira⁴
Marina Bezerra Matias Lopes⁵
Daniel Aguiar da Silva Oliveira Carvalho⁶

INTRODUÇÃO

No fim da década de 60 cunhou-se um termo conhecido como “Crise do *Software*”, foi nessa época que os profissionais voltados para a criação de *software* encontraram dificuldades decorrentes da falta de técnicas preestabelecidas e isso refletia em programas que não supriam as necessidades do cliente. De acordo com Bartié (2002), “mais de 70% dos projetos falham nas entregas de funcionalidade”. Dessa forma, para atenuar a ocorrência de falhas criaram-se os Testes de *Software*. Tal prática visa garantir a qualidade do serviço e a estabilidade da aplicação. De acordo com Neto (2007, p. 54)

“o conceito de teste de *software* pode ser compreendido através de uma visão intuitiva ou mesmo de uma maneira formal. [...] De uma forma simples, testar um *software* significa verificar através de uma execução controlada se o seu comportamento corre de acordo com o especificado. O objetivo principal desta tarefa é revelar o número máximo de falhas dispondo do mínimo de esforço, ou seja, mostrar aos que desenvolvem se os resultados estão ou não de acordo com os padrões estabelecido”(Neto, 2007 *apud* Correia, 2021).

¹ O artigo é resultado de um projeto de pesquisa.

² Aluna do Curso Técnico Integrado em Informática da Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte - IFRN, ruanarichelly04@gmail.com;

³ Aluno do Curso Técnico Integrado em Informática da Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte - IFRN, sebaslopes369@gmail.com;

⁴ Aluna do Curso Técnico Integrado em Informática da Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte - IFRN, claralet1346@gmail.com;

⁵ Aluna do Curso Técnico Integrado em Informática da Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte - IFRN, marina.216b@gmail.com;

⁶ Mestre em Sistemas e Computação da Universidade Federal do Rio Grande do Norte - UF, daniel.aguiar@ifrn.edu.br;

Bartié explica ainda que foi Myers que definiu os Testes como um processo de trabalho que tem como a finalidade encontrar erros no *software*, já que sua intenção não era comprovar a usabilidade do programa, pelo contrário, o programa deveria ser submetido a uma série de situações que forcem o erro da aplicação possibilitando encontrar a maior quantidade de falhas possíveis, permitindo que elas sejam corrigidas.

Nesse contexto, em seu trabalho “Um estudo sobre o impacto do uso de Desenvolvimento Orientado por Testes na melhoria da qualidade de *software*”, Daniela Soares (2007) também afirma que para que uma aplicação funcione corretamente, o *Test Driven Development* (TDD) — ou Desenvolvimento Guiado por Testes — é um passo imprescindível para a manutenção da integridade do *software*. A autora ainda define o TDD com uma série de iterações realizadas em um código com o objetivo de identificar erros e permitir a refatoração desses, de modo que existe um passo a passo, o qual consiste em criar testes, executá-los, identificar erros e refatorar o código.

Desse modo, tendo em vista a importância dos testes para a qualidade do *software*, este trabalho tem por objetivo aplicar e avaliar técnicas do TDD, implementadas no Laravel⁷, no contexto da disciplina de Projeto de Desenvolvimento de Software. A abordagem proposta neste trabalho será avaliada utilizando como estudo de caso o processo de desenvolvimento do sistema de divulgação de horários e notícias do Programa de Tutoria e Aprendizagem em Laboratório (TAL).

METODOLOGIA

Inicialmente foram estabelecidas três etapas de execução para a melhor realização das atividades de pesquisa a fim de permitir os resultados mais satisfatórios possíveis. A primeira etapa consiste na análise do estado da arte, com a busca e fichamento de trabalhos relacionados para melhor embasar e justificar o processo de testes de *software* no contexto da disciplina de Projeto de Software.

Na segunda etapa, foi realizado um levantamento de trabalhos sobre a utilização de *frameworks* e como eles auxiliam no processo de desenvolvimento e testagem de *software*, tornando possível analisar e entender o processo realizado nestes *frameworks*, além de

⁷ <https://laravel.com/>

viabilizar a coleta de informações sobre outras tecnologias que facilitam o processo de testagem de aplicações.

A terceira fase consiste na realização dos testes selecionados durante a segunda fase, neste caso, o teste unitário. Este têm como prática a verificação sobre a menor unidade do projeto, podendo assim sobrevir a lógica e a implementação em cada um dos módulos no qual dividiu o código, orientado pela técnica estrutural inserida no código-fonte para tratar eventuais problemas (PELIZZA *et al.*, 2018). Por fim, o TDD com Laravel será introduzido no contexto dos projetos da disciplina de Projeto de Desenvolvimento de Software.

REFERENCIAL TEÓRICO

O Desenvolvimento Guiado por Testes (TDD) é uma técnica proposta para solucionar e aprimorar a realização de análises e testes que tem como objetivo a melhora da estrutura interna de um *software* sem prejudicar a experiência do usuário com a estrutura externa (WAGNER e FANTONI, 2021) e assim entregar um produto de qualidade em um curto período de tempo. Dessa maneira, o ciclo do TDD é composto por 3 fases: 1) escrever um teste que falhe; 2) desenvolver uma aplicação que passe no teste recém criado; e 3) refatorar a aplicação (ROCHA, 2019).

Tendo por objetivo facilitar a escrita de códigos e testes, é comum a utilização de *frameworks*, códigos pré-estabelecidos que padronizam processos comuns com o objetivo de agilizar o desenvolvimento do *software*, que em sua maioria possuem modelos semelhantes entre si. Neste sentido, um dos *frameworks* que se destaca na realização de TDDs é o Laravel (PELIZZA *et al.*, 2018).

A aplicação se sobressai não só por possuir testes já definidos, como também pela estrutura conter métodos auxiliares convenientes. Esses métodos permitem testar expressamente suas aplicações, assim como há a possibilidade de integração com a biblioteca PHPUnit, a qual é um acervo de testes unitários. Ou seja, testam pequenas partes específicas do código buscando reduzir a repetição de processos e otimização do tempo de teste (PELIZZA *et al.*, 2018).

A eficácia do Laravel se dá graças a arquitetura implementada a ele de *Model-View-Controller* (MVC), a qual é conhecida por sua utilidade para assegurar a estabilidade das plataformas que utilizam dessa arquitetura (PELIZZA *et al.*, 2018). De

acordo com a pesquisa de Josué Luciano (2017) o MVC consiste na modularização do *software*, o dividindo em: *Model*, que gerencia e mantém a aplicação funcionando, armazenando os dados atribuídos a ela; *View*, que é a *interface* da aplicação, ou seja, o modo como ela se apresenta ao usuário com o objetivo simplificar a sua utilização; e *Controller*, traduz a interação do cliente com o *view* em informações compreensíveis ao *model*, fazendo a solicitação dos dados requisitados e os devolvendo ao usuário.

Nesta conjuntura, em sua pesquisa, Luis Guanolema realiza uma triagem entre cinco frameworks: Laravel, CodeInger, Synfony, Cake PHP e Yii 2, destacando as vantagens e desvantagens de cada um e ao final, percebeu que a utilização do Laravel seria mais vantajosa, uma vez que a ferramenta permite economizar o tempo, facilidade de manutenção e escalabilidade. Para obter melhores resultados Guanolema utiliza, ainda, a metodologia de TDD pois possibilita otimizar processos, garantindo uma melhoria significativa no desempenho da equipe de desenvolvimento, pois entrega como resultado final um produto de *software* robusto, sustentável e de qualidade. Após a aplicação do TDD com o Laravel, a utilização dessa abordagem fornece confiança e segurança sobre o software criado, já que a metodologia TDD ajuda a minimizar o número de erros e a produzir um programa altamente modular reutilizável e pronto para mudança. Após a implementação do sistema *web*, percebe-se o sucesso da aplicação, e das metodologias utilizadas, pois o uso da ferramenta proporcionou perceber um erro da microempresa RAZA, ensejando a resolução da falha.

Com base no que foi exposto, o presente trabalho tem por objetivo aplicar técnicas de TDD utilizando do *framework* Laravel para a realização dos testes de *software* no site em desenvolvimento para o Instituto Federal de Ciência e Tecnologia do Rio Grande do Norte (IFRN) - Campus Ipanguaçu, TAL-Horários.

RESULTADOS E DISCUSSÃO

A partir do início do desenvolvimento deste projeto, obteve-se um leque de conhecimento apurando a eficácia do TDD e o uso do Laravel nesse contexto. Foi possível ainda pesquisar sobre os variados tipos de testes de *software*, tendo como o escolhido o teste unitário para a presente pesquisa. Também definiu-se um estudo de caso para a validação da proposta no contexto da disciplina de Projeto de Desenvolvimento de Software.



Em seguida, espera-se usufruir das funcionalidades do *framework* Laravel, a fim de otimizar o processo de programação da aplicação desenvolvida, como também facilitar o acesso para outros pesquisadores interessados em reproduzir este estudo, assim como aproveitar os conhecimentos obtidos a partir do estudo de caso, realizado com a finalidade de ter uma base experimental por outros autores que alcançaram o mesmo objetivo. Também almeja-se utilizar o teste unitário, uma forma de TDD a qual tem como objetivo checar as partes menores do código a fim de evitar repetições.

CONSIDERAÇÕES FINAIS

A evolução tecnológica exige que cada vez mais os *softwares* desenvolvidos sejam capazes de funcionar com muita eficiência em um período de tempo extremamente curto, com o objetivo de atender às demandas do mercado consumidor. Assim, os processos de testes de *software* necessitam ser executados rapidamente a fim de assegurar a escalabilidade das aplicações criadas, ou seja, certificar-se de que mesmo com a alta requisição de serviços, todos os pedidos, ou ao menos a sua grande maioria, serão atendidos.

Desse modo, a criação de *frameworks* torna-se um ponto muito positivo em termos de economia de tempo e garantia de funcionalidade. Ao analisar-se por exemplo o Laravel, objeto de pesquisa deste estudo, ficou claro que a praticidade de possuir códigos genéricos para questões mais ordinárias no ambiente de programação, o desenvolvimento de sites e aplicativos exige muito menos trabalho do programador, que passa então a destinar esse tempo sobressalente para a testabilidade do *software*, que poderá ser realizada de maneira mais abrangente e concisa, impedindo que o usuário final venha a ter problemas quando utilizar deste serviço. Ademais, a plataforma também se mostrou muito eficiente no quesito de experimentações, visto que, a preexistência de testes servem não somente para garantir a funcionalidade da aplicação, mas também como um guia para que programadores menos experientes possam ter uma linha de raciocínio sobre por onde começar e o que analisar nesse processo vital para a aplicabilidade cada vez mais necessária de novos *softwares* no mercado.

Palavras-chave: Testes, Software, Laravel, Framework.

REFERÊNCIAS

- BARTIÉ, A. **Garantia da qualidade de software**. Rio de Janeiro: Campus, 2002.
- CORREIA, João Marcelo Gomes; FERREIRA, João Victor Custódio; PIRES, Daniel Facciolo. **IMPORTÂNCIA DOS TESTES PARA A QUALIDADE DO SOFTWARE**. Revista Eletrônica de Computação Aplicada, v. 2, n. 2, 2021. Acesso em: 3 de maio de 2022.
- FEITOSA, Daniela Soares. **Um estudo sobre o impacto do uso de desenvolvimento orientado por testes na melhoria da qualidade de software**. Monografia (Bacharelado em Ciência da Computação), 2007. Acesso em: 07 de junho de 2022.
- GUANOLEMA CHOCA, Luis Alberto. **Desarrollo de un sistema web para automatizar el proceso de compra y venta en la microempresa raza utilizando la tecnología Laravel y Vue.js bajo un enfoque de desarrollo dirigido por pruebas (TDD)**. 2019. Trabalho de Conclusão de Curso. Escuela Superior Politécnica de Chimborazo.
- LUCIANO, Josué; ALVES, Wallison Joel Barberá. **Padrão de arquitetura MVC: Model-view-controller**. EPeQ Fafibe, v. 1, n. 3a, p. 102-107, 2017.
- NETO, A. C. D. **Introdução a teste de software**. Engenharia de Software Magazine, v. 1, p. 22, 2007.
- PELIZZA, Angelica Caetane; BERTOLINI, Cristiano; SILVEIRA, Sidnei Renato. **Um estudo sobre técnicas de teste de software no Framework Laravel**. Revista Eletrônica de Sistemas de Informação e Gestão Tecnológica, v. 9, n. 3, 2018. Acesso em: 10 de maio de 2022.
- ROCHA, Fabio G. et al. **Agile Teaching Practices: Using TDD and BDD in Software Development Teaching**. In: **Proceedings of the XXXIII Brazilian Symposium on Software Engineering**. 2019. Acesso em: 3 de maio de 2022.
- WAGNER, Daniel Pinheiro; FANTONI, Felipe Silva. **Aplicando Desenvolvimento Guiado por Testes para Auxiliar na Evolução de Arquiteturas Monolíticas para Microsserviços**. Acesso em: 3 de maio de 2022.