

Gii Tester: Uma Ferramenta Para Manutenção de Testes Automáticos de Software em Sistemas WEB

Allan Kaio Brito Macedo¹, Alessandro Rodrigues e Silva²

¹Instituto Federal de Educação, Ciência e Tecnologia de Goiás – Campus Anápolis

allankaio276@gmail.com, mestre.alessandros@gmail.com

Abstract. *WEB systems, especially those exposed to the Internet, require constant quality assurance achieved through automated testing, among other artifacts. In Software Engineering, when using an incremental model in development, there is a need for continuous test updates during the maintenance phase, consuming time and consequently increasing development costs. This paper presents the development of an automated testing tool called Gii tester, which utilizes template techniques along with code standardization strategies to generate automatic functional tests.*

Resumo. *Sistemas WEB, principalmente aqueles expostos à Internet, necessitam de constante asserção de qualidade, obtida por meio de testes automáticos, dentre outros artefatos. Na Engenharia de Software, ao utilizar um modelo incremental no desenvolvimento, é necessário constante atualização dos testes durante a fase de manutenção, demandando tempo e consequentemente elevando o custo de desenvolvimento. Nesse trabalho é apresentado o desenvolvimento de uma ferramenta de produção de testes automáticos, denominada Gii tester. Que utiliza de técnicas de templates, juntamente com estratégias de padronização de código, para geração de testes funcionais automáticos.*

1. Introdução

A área de desenvolvimento de Sistemas *WEB* tem experimentado um notável crescimento no mercado, demandando a criação de sistemas seguros e eficazes para atender às necessidades dos clientes. Essa abordagem de desenvolvimento *web* se destaca por oferecer acessibilidade por meio de um navegador, economia de espaço em disco para o cliente em comparação com aplicações *Desktop*, compatibilidade com dispositivos que suportem navegadores modernos e facilidade na atualização da aplicação diretamente no servidor, sem a necessidade de intervenção do usuário.

No entanto, as aplicações *WEB* também trazem consigo preocupações significativas quanto à qualidade do *software*, particularmente no que diz respeito à sua resiliência a erros. Para garantir a qualidade e a longevidade do *software*, é crucial aplicar os princípios da engenharia de *software* (Devmedia, 2018). A qualidade do *software* é influenciada por diversos fatores, e os testes de *software* desempenham um papel fundamental na detecção de erros durante o desenvolvimento e manutenção do *software*. A pesquisa realizada por Delamaro et al. (2017) classifica os testes de *software* em três técnicas: funcionais, estruturais e baseadas em defeitos (ou erros). Os testes funcionais são baseados na especificação de requisitos e casos de uso, permitindo ao testador verificar se o comportamento do *software* está de acordo com o esperado sem a necessidade de acesso ao código-fonte. Tendo outros tipos como os de integração, unidade, regressão e desempenho, cada um com sua função específica.

A realização manual de testes de *software* é uma tarefa demorada, destacando a vantagem de testes automatizados (Daniel e Vitor 2021). Para abordar essa questão, a automação de testes é uma solução viável, envolvendo a execução de *scripts* automatizados para tarefas que, de outra forma, seriam executadas manualmente. Nesse contexto, a implementação de uma ferramenta de código aberto visa à criação e manutenção automatizada de testes funcionais, reduzindo os custos de desenvolvimento e aprimorando a qualidade do *software* em processos de desenvolvimento incrementais. A ferramenta atualiza e gera *scripts* de testes de forma rápida com mínima intervenção do programador. Para avaliar quantitativamente o impacto dessa ferramenta, propõe-se a realização de uma pesquisa experimental.

2. Materiais e Métodos

O processo de desenvolvimento da ferramenta Gii Tester seguiu a metodologia ágil *Kanban*, visando melhorar a agilidade no desenvolvimento do trabalho. Isso envolveu o levantamento dos requisitos a serem desenvolvidos, bem como a definição das etapas e prazos de entrega. Para fins de teste, a ferramenta foi aplicada a um sistema em produção desde 2016, que serviu como estudo de caso para avaliar os *scripts* de testes funcionais gerados pela ferramenta, os quais seguem o formato estabelecido pelo *framework* Codeception (Codeception, 2018). Desenvolvido para o *Framework Yii* (*Framework*, Y. 2018), embora o método de desenvolvimento seja independente disso.

No desenvolvimento dos testes, a ferramenta acessa a base de dados do sistema para obter informações sobre nomes de tabelas e tipos de dados, além de consultar o modelo da aplicação para adquirir regras e mensagens de retorno do sistema. Posteriormente, a ferramenta gera automaticamente os testes funcionais utilizando templates preenchidos com os dados obtidos. Em casos nos quais a ferramenta não consegue preencher determinados campos, devido à sua natureza específica, esses campos são identificados nos próprios testes, requerendo uma intervenção mínima por parte do programador para completar o processo de programação dos testes automáticos.

Mediante da emulação de requisições HTTP, a ferramenta interage com o servidor, simulando o comportamento de um cliente. Ela realiza testes nas aplicações de formulários do sistema, detalhando todo o processo, desde o acesso às rotas até a aplicação dos formulários de testes e a análise dos resultados da execução dos testes.

3. Resultados

Após o desenvolvimento, a ferramenta Gii Tester mostrou ser rápida na geração de testes automáticos utilizando modelos de testes funcionais predefinidos para testar operações básicas do sistema como ler, escrever, editar e deletar. Essa geração ocorre de maneira ágil, exigindo intervenção mínima por parte do programador durante o processo de desenvolvimento. A ferramenta também fornece orientações ao programador quando ocorrem limitações que exigem ajustes nos campos dos testes gerados. Tendo um script de execução gerado com os testes com a ordem correta de execução baseado nas dependências das tabelas do banco de dados.

Entre as funcionalidades da ferramenta Gii Tester, destacam-se a capacidade de gerar testes para criação, atualização e exclusão de dados, aplicando formulários ao modelo da aplicação. Além disso, a ferramenta é capaz de testar diversos tipos de dados, como campos não nulos, campos numéricos e outros tipos, como e-mail, data, máximos ou mínimos, endereços eletrônicos e campos lógicos. Após a aplicação dos formulários,

os testes verificam se as respostas da aplicação estão de acordo com o esperado e fornecem resultados sobre a execução do teste específico para o tipo de dado testado.

A Figura 1 apresenta um exemplo de um *template* de teste funcional automático implementado em PHP, que tem como objetivo testar a aplicação de um formulário à aplicação, verificando campos não nulos e as mensagens de retorno ao usuário de acordo com o modelo.

```
public function <?= $modelName . 'NotNullFields' ?>(FunctionalTester $I)
{
    $I->wantTo('Verify exception for not null fields');
    $I->amOnRoute('<?= $route ?>/create');
    $I->submitForm('form', []);
<?php foreach ($tableSchema->columns as $column): ?>
<?php if (($column->allowNull==false && $column->name!=$pk[0]) || $helper->isThisRule($column->name, $modelRules,
<?php $options = $require->getClientOptions($model, $column->name)?>
<?php $isCusMens = $helper->isCustomMessage($modelRules, rule: 'required', $column->name)?>
<?php if ($isCusMens[0]):?>
    $I->see('<?= $isCusMens[1]?>');
<?php else:?>
    $I->see('<?= $options['message']?>');
<?php endif;?>
<?php endif;?>
<?php endforeach;?>
}
```

Figura 1: Demonstração de um *template* para geração de um teste automático funcional para testar campos não nulos.

Os *templates* seguem um padrão de desenvolvimento que envolve a leitura de todas as colunas da tabela selecionada na web. Por meio de verificações, apenas os dados relevantes para os testes são selecionados por meio de funções auxiliares. No caso da Figura 1, apenas os dados obrigatórios a serem fornecidos ao formulário são filtrados ao verificar quais dados são obrigatórios na aplicação para o modelo e no banco de dados. As mensagens de resposta são capturadas pela aplicação sendo as mesmas que ocorre no formulário para o usuário, simulando a interação real. Após os campos preenchidos e gerado o arquivo de testes funcional.

A Figura 2 ilustra o resultado do *template* apresentado na Figura 1. Após a coleta de informações sobre a aplicação e a geração do teste funcional automático, são realizados testes para campos não nulos e verificação das mensagens de retorno ao usuário com base no modelo da aplicação.

```
public function PessoaNotNullFields(FunctionalTester $I)
{
    $I->wantTo( text: 'Verify exception for not null fields');
    $I->amOnRoute( route: 'pessoa/create');
    $I->submitForm( selector: 'form', []);
    $I->see( text: 'Nome cannot be blank. ');
    $I->see( text: 'CPF cannot be blank. ');
}
```

Figura 2: Teste funcional automático para testar campos não nulos da tabela/Modelo Pessoa, gerado através do *template* da figura 1.

Os testes gerados seguem o padrão do *framework* Codeception (Codeception, 2018). O preenchimento das funções é realizado automaticamente no *template*, com base nas

informações fornecidas pela aplicação. Isso torna possível implementar um único *template* e gerar testes automáticos para todos os modelos da aplicação de forma automatizada. Podendo ser executados de forma manual ou automática com o *script* executável gerado com os testes, que está na ordem correta de execução.

4. Conclusão

A ferramenta Gii Tester, disponível em <https://github.com/AllanKaio21/GiiTester>, mostra ser uma valiosa contribuição para o desenvolvimento de testes funcionais automáticos em aplicações web, agilizando significativamente o processo de geração de testes com uma melhor eficiência no desenvolvimento de *softwares*. Ao automatizar a criação de testes funcionais com base em modelos predefinidos, a ferramenta reduz a intervenção manual, aumentando a eficácia dos testes e a qualidade do *software* resultante. Com seu potencial para lidar com uma variedade de tipos de dados e casos de teste, a Gii Tester oferece uma abordagem poderosa para a garantia da qualidade em sistemas web, contribuindo para a criação de aplicações mais robustas e confiáveis. Propomos a condução de um experimento para avaliação quantitativa da eficácia da ferramenta em sua função, visando inclusive a expansão de suas funcionalidades.

Referências

- CODECEPTION (2018). “Introduction”, <https://codeception.com/docs/Introduction>, Novembro.
- DEVMEDIA (2018). “Testabilidade e planejamento de testes de software”.
- DECLAMARO, M., Jino, M., and Maldonado, J. (2017). “Introdução ao teste de software”. Elsevier Brasil.
- FANTINATO, M., da Cunha, A. C. R., Dias, S. V., Cardoso, S. A. M., and Cunha, C. A. Q. (2004). “Autotest—um framework reutilizável para a automação de teste funcional de software”. Simpósio Brasileiro de Qualidade de Software.
- FRAMEWORK, Y. (2018). “Guia definitivo para yii 2.0”.
- FRANCH, X.; Ribó, J. (2002). “Supporting Process Reuse in PROMENADE”. Research
- MACORATTI, J.C. “Padrões de projetos. 2010. O modelo MVC – Model View Controller”. Disponível em <http://www.macoratti.net/vbn_mvc.htm>. Acesso em 26/set/2022.
- DANIEL, A., E. and Vitor, N., M. (2021). “Testes automaticados: Teoria e prática”. Disponível em <https://www.unifal-mg.edu.br/dcc/wp-content/uploads/sites/221/2022/01/TCC_DanielEscobar_e_VitorMuniz.pdf>. Acesso em 23/nov/2023.