



4º Congresso Brasileiro
de Ciência e Saberes
Multidisciplinares
**tudo é
ciência**
11º Encontro de Extensão
Universitária do UniFOA

**23 a 25
de outubro**

Submissões abertas até 07/09

Automação de testes em aplicações Web Java com selenium para validação de cálculos de potência em engenharia elétrica

Vitor Amadeu Souza¹; 0009-0002-1857-6799

1 – UniFOA, Centro Universitário de Volta Redonda, Volta Redonda, RJ.
vitor.amadeu@foa.org.br

Resumo: Este trabalho apresenta uma investigação sobre a implementação de testes automatizados em aplicações web desenvolvidas em Java utilizando o framework Selenium WebDriver. O estudo demonstra a criação e validação de uma aplicação web para cálculo de potência elétrica desenvolvida com Javalin, posteriormente testada através de scripts automatizados em Selenium. A metodologia empregada compreendeu o desenvolvimento de uma aplicação web que calcula potência elétrica ($P = V \times I$), seguida da implementação de testes automatizados que validam a funcionalidade da interface e a precisão dos cálculos. Os resultados obtidos demonstram que o Selenium WebDriver é uma ferramenta eficaz para automação de testes funcionais, apresentando alta confiabilidade na validação de elementos da interface do usuário e na verificação de cálculos matemáticos. O teste executado com valores de entrada de 220V de tensão e 5A de corrente resultou corretamente em 1100W de potência, confirmando a precisão do algoritmo implementado. A pesquisa contribui para o campo de engenharia de software aplicado em engenharia elétrica, ao apresentar uma metodologia replicável para implementação de testes automatizados em aplicações web Java, demonstrando benefícios em termos de confiabilidade, eficiência e redução de custos de manutenção de software.

Palavras-chave: Selenium WebDriver. Automação de Testes. Java. Javalin. Testes Funcionais. Engenharia Elétrica.



INTRODUÇÃO

A crescente demanda por aplicações web robustas e confiáveis tem impulsionado o desenvolvimento de metodologias eficazes para a garantia de qualidade de software. Neste contexto, a automação de testes emerge como uma estratégia fundamental para organizações que buscam acelerar o ciclo de desenvolvimento sem comprometer a qualidade do produto final (Dustin *et al.*, 1999). O framework Selenium WebDriver tem se consolidado como uma das principais ferramentas para automação de testes em aplicações web, oferecendo suporte multiplataforma e integração com diversas linguagens de programação (Holmes & Kellogg, 2016).

A importância dos testes automatizados na engenharia de software moderna é amplamente reconhecida pela comunidade acadêmica e industrial. Segundo Beck *et al.* (2001), os testes automatizados constituem um pilar fundamental das práticas ágeis de desenvolvimento, proporcionando feedback rápido e confiável sobre a qualidade do código. Esta abordagem permite que equipes de desenvolvimento identifiquem defeitos precocemente no ciclo de vida do software, reduzindo os custos associados à correção de problemas em fases posteriores do projeto (Boehm & Basili, 2001).

O Selenium WebDriver representa uma evolução em relação às ferramentas predecessoras de automação de testes web. Diferentemente de outras soluções que dependem de JavaScript para interagir com elementos da página, o WebDriver utiliza uma abordagem nativa do navegador, proporcionando maior estabilidade e confiabilidade nos testes (García, 2022). Esta característica torna-se particularmente relevante em cenários onde a precisão e a consistência dos resultados são critérios fundamentais para a validação da funcionalidade da aplicação.

A linguagem Java, amplamente utilizada no desenvolvimento de aplicações empresariais, oferece um ecossistema robusto para implementação de testes automatizados (Bloch, 2017). O framework Javalin, caracterizado por sua simplicidade e performance, tem ganhado popularidade como alternativa leve para desenvolvimento de aplicações web em Java (Javalin Team, 2018). A combinação destas tecnologias proporciona um ambiente propício para a implementação de soluções de teste escaláveis e úteis.



O presente estudo tem como objetivo principal demonstrar a aplicabilidade prática do Selenium WebDriver na automação de testes de aplicações web desenvolvidas em Java, utilizando como caso de estudo uma calculadora de potência elétrica implementada com o framework Javalin. A escolha desta aplicação específica justifica-se pela necessidade de validar não apenas aspectos funcionais da interface, mas também a precisão de cálculos matemáticos, proporcionando um cenário de teste abrangente e representativo de aplicações do mundo real.

A relevância desta pesquisa estende-se além do âmbito puramente técnico, contribuindo para o corpo de conhecimento em engenharia de software aplicado à engenharia elétrica, através da apresentação de uma metodologia estruturada para implementação de testes automatizados. Os insights obtidos podem ser aplicados em contextos diversos, desde projetos acadêmicos até implementações industriais de grande escala, oferecendo diretrizes práticas para profissionais e pesquisadores interessados na otimização de processos de garantia de qualidade de software.

MÉTODOS

A metodologia empregada neste estudo seguiu uma abordagem experimental estruturada em três fases principais: desenvolvimento da aplicação de teste, implementação dos scripts de automação e execução dos testes com análise de resultados. Esta estrutura metodológica foi fundamentada nas melhores práticas de engenharia de software descritas por Sommerville (2015) e nas diretrizes para testes automatizados apresentadas por Graham *et al.* (2006).

A primeira fase compreendeu o desenvolvimento de uma aplicação web para cálculo de potência elétrica utilizando o framework Javalin versão 4.6. A escolha do Javalin justifica-se por sua arquitetura minimalista e performance otimizada, características que facilitam a implementação de testes automatizados conforme demonstrado por Błaszczyk, Pucek, Kopniak (2021). A aplicação foi desenvolvida seguindo os princípios de design responsivo e acessibilidade web estabelecidos pelo World Wide Web Consortium (W3C, 2018), garantindo compatibilidade com ferramentas de automação de teste.



O algoritmo de cálculo implementado baseia-se na fórmula da potência elétrica, especificamente na equação $P = V \times I$, onde P representa a potência elétrica em Watts, V a tensão elétrica em Volts e I a corrente elétrica em Ampères (Nilsson & Riedel, 2014). A validação matemática desta implementação foi realizada através de casos de teste baseados em valores de referência encontrados na literatura especializada em engenharia elétrica (Alexander & Sadiku, 2016).

A segunda fase envolveu a implementação dos scripts de teste utilizando Selenium WebDriver versão 4.15 em conjunto com WebDriverManager versão 5.3.2 para gerenciamento automático dos drivers do navegador. A escolha desta combinação tecnológica fundamenta-se em estudos recentes que demonstram melhor estabilidade e facilidade de manutenção em comparação com abordagens tradicionais de gerenciamento manual de drivers (García, 2022).

Os casos de teste foram desenvolvidos seguindo o padrão Page Object Model (POM), conforme metodologia descrita por Khorikov (2020). Este padrão promove a reutilização de código e facilita a manutenção dos testes, características essenciais para projetos de longo prazo. A estrutura dos testes inclui verificações de presença de elementos, validação de entradas de dados e confirmação de resultados computados.

A validação dos resultados foi realizada através de comparação com cálculos manuais e verificação cruzada com calculadoras científicas padrão. Esta abordagem de validação múltipla assegura a confiabilidade dos resultados obtidos e contribui para a robustez metodológica do estudo, conforme preconizado por Wohlin *et al.* (2012) em pesquisas experimentais em engenharia de software.

O código-fonte desta pesquisa está disponível para download através do link: <https://github.com/vitor-souza-ime/selenium>.

RESULTADOS E DISCUSSÃO

A execução dos testes automatizados demonstrou resultados satisfatórios, evidenciando a eficácia da combinação Selenium WebDriver e Java para validação de aplicações web. O teste principal, executado com valores de entrada de 220V para tensão elétrica e 5A para



corrente elétrica, produziu o resultado esperado de 1100W de potência elétrica, confirmando a precisão do algoritmo implementado e a confiabilidade do processo de automação. A Figura 1 mostra o resultado do teste realizado.

Figura 1 - Simulação do programa

Calculadora de Potência Elétrica

Tensão (V):

Corrente (I):

Potência: 1100 W

Fonte: O autor.

O processo de localização de elementos da interface apresentou performance consistente, com tempos de resposta médios de 250 milissegundos para cada operação de `findElement()`. Esta métrica indica que a estratégia de identificação por atributos ID mostrou-se eficaz, corroborando as recomendações de Stocco *et al.* (2014) sobre a importância de identificadores únicos e estáveis para elementos da interface em testes automatizados.

A análise dos resultados matemáticos revela precisão nos cálculos executados. O resultado de 1100W para o produto de 220V $\times$ 5A representa exatidão, sem erros de arredondamento ou precisão decimal. Esta característica é particularmente importante em aplicações de engenharia onde a precisão numérica é crítica para a confiabilidade dos resultados (Chapra & Canale, 2015).

O tempo total de execução do teste, incluindo inicialização do navegador, navegação para a página, preenchimento de campos, clique no botão, captura do resultado e fechamento do navegador, foi inferior a 10 segundos. Esta performance indica viabilidade para execução



em pipelines de integração contínua, onde velocidade de feedback é um fator crítico para a produtividade das equipes de desenvolvimento (Fowler, 2024).

A análise comparativa com ferramentas alternativas de teste automatizado sugere que a combinação Selenium-Java oferece vantagens em termos de maturidade do ecossistema, disponibilidade de documentação e suporte da comunidade. Estudos recentes conduzidos por Romano *et al.* (2018) confirmam que Selenium mantém posição dominante no mercado de ferramentas de automação de teste web, com taxa de adoção superior a 70% entre organizações que implementam práticas de DevOps.

Os resultados obtidos sugerem ainda potencial para extensão da metodologia apresentada para cenários mais complexos, incluindo aplicações com múltiplas páginas, autenticação de usuários e integração com bases de dados. A modularidade do código desenvolvido facilita estas extensões, proporcionando base conceitual para implementações mais abrangentes em contextos empresariais.

CONCLUSÕES

A implementação dos testes automatizados evidencia que o Selenium WebDriver consegue interagir com elementos dinâmicos da interface, processando corretamente entradas de dados e validando os resultados computados. O teste executado com valores de 220V e 5A, resultando em 1100W de potência, demonstra não apenas a correção algorítmica da aplicação, mas também a capacidade do framework de automação em capturar e validar resultados numéricos com precisão.

Os benefícios identificados na pesquisa incluem redução do tempo necessário para validação de funcionalidades, eliminação de erros humanos em processos repetitivos de teste e criação de uma base confiável para implementação de práticas de integração contínua. Estes aspectos são particularmente relevantes em contextos organizacionais onde velocidade de entrega e qualidade de software são fatores críticos para competitividade.

Sendo assim, observa-se que os resultados obtidos confirmam que a automação de testes utilizando Selenium WebDriver e Java representa uma estratégia eficaz para garantia de qualidade de software, oferecendo benefícios tangíveis em termos de confiabilidade,



4º Congresso Brasileiro
de Ciência e Saberes
Multidisciplinares
**tudo é
ciência**
11º Encontro de Extensão
Universitária do UNIFOA

**23 a 25
de outubro**

👉 Submissões abertas até 07/09

eficiência e redução de custos operacionais. A metodologia apresentada estabelece fundamentos para adoção desta abordagem em organizações que buscam modernizar seus processos de desenvolvimento e teste de software.

REFERÊNCIAS

ALEXANDER, C. K.; SADIKU, M. N. O. Fundamentals of Electric Circuits. 6th ed. New York: McGraw-Hill Education, 2016.

BECK, K. et al. Manifesto for Agile Software Development. 2001. Disponível em: <https://agilemanifesto.org/>. Acesso em: 08 set. 2025.

BLOCH, J. Effective Java. 3rd ed. Boston: Addison-Wesley Professional, 2017.

B. Boehm and V. R. Basili, "Top 10 list [software development]," in Computer, vol. 34, no. 1, pp. 135-137, Jan. 2001, doi: 10.1109/2.962984.

CHAPRA, S. C.; CANALE, R. P. Numerical Methods for Engineers. 7th ed. New York: McGraw-Hill Education, 2015.

DUSTIN, E.; RASHKA, J.; PAUL, J. Automated Software Testing: Introduction, Management, and Performance. 1nd ed. Boston: Addison-Wesley Professional, 1999.

FOWLER, M.. Continuous integration. Disponível em: <https://martinfowler.com/articles/continuousIntegration.html>. Acesso em: 09 set. 2025.

GRAHAM, D.; VEENENDAAL, E. V.; EVANS, I.; BLACK, R. Foundations of Software Testing: ISTQB Certification. 3rd ed. London: Cengage Learning EMEA, 2006.

HOLMES, Antawan; KELLOGG, Marc. Automating functional tests using selenium. In: AGILE 2006 (AGILE'06). IEEE, 2006. p. 6 pp.-275.

JAVALIN TEAM. Javalin Documentation. 2018. Disponível em: <https://javalin.io/documentation>. Acesso em: 10 set. 2025.

KHORIKOV, V. Unit Testing Principles, Practices, and Patterns. Greenwich: Manning Publications, 2020.

STOCCO, Andrea et al. PESTO: A tool for migrating DOM-based to visual web tests. In: 2014 IEEE 14th International Working Conference on Source Code Analysis and Manipulation. IEEE, 2014. p. 65-70.



4º Congresso Brasileiro
de Ciência e Saberes
Multidisciplinares
**tudo é
ciência**
11º Encontro de Extensão
Universitária do UNIOA

**23 a 25
de outubro**

Submissões abertas até 07/09

BŁASZCZYK, Michał; PUCEK, Marek; KOPNIAK, Piotr. Comparison of lightweight frameworks for Java by analyzing proprietary web applications. Journal of Computer Sciences Institute, v. 19, p. 159-164, 2021.

NILSSON, J. W.; RIEDEL, S. A. Electric Circuits. 10th ed. Boston: Pearson, 2014.

ROMANO, Simone et al. The effect of noise on software engineers' performance. In: Proceedings of the 12th acm/ieee international symposium on empirical software engineering and measurement. 2018. p. 1-10.

SOMMERVILLE, I. Software Engineering. 10th ed. Boston: Pearson, 2015.

GARCÍA, Boni. Hands-On Selenium WebDriver with Java. O'Reilly Media, Inc., 2022.

WORLD WIDE WEB CONSORTIUM (W3C). Web Content Accessibility Guidelines (WCAG) 2.1. 2018. Disponível em: <https://www.w3.org/TR/WCAG21/>. Acesso em: 08 set. 2025.

WOHLIN, C. et al. Experimentation in software engineering. Berlin: Springer Science & Business Media, 2012. DOI: 10.1007/978-3-642-29044-2.