

# Desenvolvimento e Avaliação de um Framework de Automação para Testes no Ciclo de Vida do Software

Nando Augusto Veloso Tupinambá<sup>1</sup>

<sup>1</sup>Instituto de Informática – Universidade PUC Minas PL (PUCMG)  
Caixa Postal 30-180.100 – Belo Horizonte – MG – Brazil

<sup>2</sup>Department of Computer Science – University of Durham  
Durham, U.K.

<sup>3</sup>Departamento de Sistemas e Computação  
Universidade Regional de Blumenau (FURB) – Blumenau, SC – Brazil

{augustonando5}@gmail.com

**Abstract.** *This paper presents the development and evaluation of a testing automation framework aimed at enhancing efficiency and performance in the software development lifecycle, especially in agile environments. Automation in software testing is critical to ensure productivity, quality, and speed throughout the development process. The proposed framework will be applied in real-world projects and evaluated for its impact on performance improvement, scalability, and resource management.*

**Resumo.** *Este artigo apresenta o desenvolvimento e a avaliação de um framework de automação de testes com o objetivo de aprimorar a eficiência e o desempenho no ciclo de vida do desenvolvimento de software, especialmente em ambientes ágeis. A automação de testes é fundamental para garantir produtividade, qualidade e rapidez durante o processo de desenvolvimento. O framework proposto será aplicado em projetos reais e avaliado quanto ao seu impacto na melhoria de desempenho, escalabilidade e gerenciamento de recursos.*

**Bacharelado em Engenharia de Software - PUC Minas**  
**Trabalho de Conclusão de Curso (TCC)**

Orientador de conteúdo (TCC I): Aline Brito - alinebrito@pucminas.br  
Orientador de conteúdo (TCC I): Leonardo Vilela - leonardocardoso@pucminas.br  
Orientador acadêmico (TCC I): Cleiton Tavares - cleitontavares@pucminas.br  
Orientador do TCC II: (A ser definido no próximo semestre)

Belo Horizonte, DIA de MÊS de ANO.

Conforme discutido em estudos anteriores sobre automação de testes [Marques Borges and Oliveira Lima 2024] [Teste.ai 2024] [?] [Alura 2023] [?] [ClickUp 2024] [ProgramaThor 2024] [Prime Control 2024] [MIT Technology Review Brasil 2024] [Objective 2022] [LogAp 2024] [Prime Control 2017]

## 1. Introdução

**Problema:** A automação de testes no desenvolvimento de software é uma prática fundamental que visa melhorar a eficiência, a produtividade e a qualidade do produto final. Em ambientes ágeis e de entrega contínua, a automação permite que o ciclo de desenvolvimento seja mais rápido e consistente, reduzindo erros e proporcionando feedback imediato para os desenvolvedores. De acordo com o artigo "Inteligência Artificial Aplicada à Qualidade de Software"(Revista FT, 2024), o uso de frameworks como o Selenium tem se destacado na automação de aplicações web, facilitando a execução e a manutenção dos testes automatizados.

Outro aspecto importante da automação de testes é sua capacidade de racionalizar o uso de recursos computacionais. A utilização de frameworks em ambientes de nuvem é explorada no estudo "Como a IA está transformando a automação de testes: técnicas e ferramentas"(Engage AI, 2024), que propõe soluções para execução paralela de testes, aumentando a eficiência do processo em comparação com os métodos tradicionais. A aplicação desse tipo de framework é essencial para empresas que buscam escalar a execução dos testes sem sobrecarregar os recursos internos.

**O problema que este trabalho busca resolver é a ausência de frameworks de automação de testes adequados que atendam às demandas de eficiência e qualidade em ambientes de desenvolvimento ágeis.** A falta de ferramentas robustas e eficazes prejudica a produtividade das equipes de desenvolvimento, resultando em custos elevados e perda de qualidade do produto. O estudo "LambdaTest expande capacidade do KaneAI para aprimorar a eficiência e flexibilidade dos testes"(Estado de Minas - Mundo Corporativo, 2024) reforça a importância da reutilização de artefatos e da execução automatizada sob diversas condições, apontando soluções que podem ser integradas para atender às necessidades de qualidade e entrega contínua.

**Objetivo Geral: Desenvolver e avaliar um framework de automação para testes que melhore a eficiência e o desempenho no ciclo de vida do desenvolvimento de software.**

Para atingir este objetivo, serão perseguidos os seguintes objetivos específicos:

**Desenvolver um framework de automação que seja facilmente integrável ao processo de desenvolvimento ágil, suportando diferentes abordagens e metodologias. Avaliar o desempenho e eficiência do framework desenvolvido utilizando métricas de qualidade de software em projetos reais. Comparar o framework proposto com outras soluções existentes no mercado para determinar seus pontos fortes e possíveis limitações.** Espera-se que o framework desenvolvido promova ganhos expressivos de eficiência nas fases de desenvolvimento e manutenção de software, reduzindo o tempo gasto com testes manuais e aumentando a confiabilidade dos resultados. Além disso, a solução deve melhorar a alocação de recursos, minimizar a latência e contribuir para a criação de um ambiente de desenvolvimento mais ágil e eficaz.

Dessa forma, este trabalho está estruturado em oito seções. A Seção 2 aborda a fundamentação teórica, enquanto a Seção 3 apresenta os trabalhos relacionados. A Seção 4 detalha os materiais e métodos adotados. As Seções 5 e 6 discutem os resultados, respectivamente. Na Seção 7, são discutidas as ameaças à validade. Por fim, a Seção 8 conclui o trabalho, indicando direções para trabalhos futuros.

## 2. Fundamentação Teórica

Para melhor compreensão dos conceitos que compõem as soluções abordadas neste estudo, esta seção apresenta uma fundamentação teórica sobre: Automação de Testes de Software, Eficiência em Ambientes Ágeis, Frameworks de Testes Automatizados e Qualidade de Software.

### 2.1. Automação de Testes de Software

A automação de testes é um dos pilares da Engenharia de Software, pois visa melhorar a eficiência e a qualidade dos processos de desenvolvimento de sistemas. Segundo o artigo *Inteligência Artificial Aplicada à Qualidade de Software*, publicado pela Revista FT (2024), os testes automatizados têm como objetivo verificar a funcionalidade de um sistema de forma rápida e consistente, garantindo assim uma maior cobertura dos cenários de uso. A automação de testes permite reduzir o tempo de execução dos testes manuais e minimizar a ocorrência de erros humanos, tornando o processo de desenvolvimento mais ágil e seguro.

A automação, no entanto, não consegue garantir a ausência de defeitos em um sistema, pois depende da qualidade dos casos de teste elaborados. Ainda assim, é uma prática essencial para a redução de riscos e para a entrega contínua em ambientes ágeis (Pressman and Maxim, 2021).

### 2.2. Eficiência em Ambientes Ágeis

A eficiência em ambientes de desenvolvimento ágeis é um dos principais desafios enfrentados pelas equipes de desenvolvimento de software. A entrega contínua, característica desses ambientes, demanda a execução de testes constantes e rápidos. Segundo o artigo publicado pela Alura (2023), a integração de práticas de DevOps com a automação de testes pode melhorar significativamente a qualidade do software, permitindo a identificação e resolução de problemas antes da implantação. Essa abordagem promove a colaboração entre as equipes de QA e DevOps, facilitando a automação de diversos tipos de testes e garantindo uma entrega contínua.

O uso de frameworks de automação, como o Selenium, visa justamente melhorar a eficiência dos processos de teste, permitindo que os desenvolvedores foquem na evolução das funcionalidades do sistema. A abordagem ágil de desenvolvimento e teste contínuo é capaz de promover ganhos expressivos de produtividade e qualidade (Larman, 2021).

### 2.3. Frameworks de Testes Automatizados

Frameworks de testes automatizados são ferramentas fundamentais para a execução de testes em aplicações web e outros sistemas. O artigo *12 frameworks de Automação de testes de software para utilizar em 2024*, publicado pela Teste.ai (2024), apresenta uma variedade de frameworks que podem ser utilizados, cada um com características específicas que atendem a diferentes necessidades de projetos. A escolha do framework adequado é crucial para garantir a eficiência e a qualidade dos testes automatizados, sendo um fator fundamental para o desenvolvimento do framework proposto neste trabalho.

Outro aspecto importante dos frameworks de automação de testes é a sua capacidade de integração com ambientes de nuvem. Segundo o estudo publicado pela Engage AI (2024), a utilização de recursos de nuvem para execução de testes paralelos permite

escalar a capacidade de execução, aumentando a eficiência e reduzindo o tempo de resposta.

## **2.4. Qualidade de Software**

A qualidade de software é um conceito amplo que engloba a confiabilidade, manutenibilidade, eficiência e usabilidade do produto final. Segundo o *Glossário da IEEE (1990)*, qualidade de software é definida como o grau de conformidade de um sistema, componente ou processo com as necessidades e expectativas do usuário. Em ambientes ágeis, a qualidade de software está diretamente ligada à capacidade de se adaptar rapidamente às mudanças de requisitos, mantendo a estabilidade e a funcionalidade esperada.

No contexto deste trabalho, a qualidade de software está relacionada à capacidade do framework proposto de reduzir defeitos, melhorar a cobertura dos testes e promover uma execução mais eficiente dos processos de validação, garantindo assim que o software entregue atenda às expectativas dos usuários e stakeholders. Segundo o estudo *Automação de Testes: transformando a qualidade do software* da Spread (2024), a automação de testes facilita a detecção precoce de defeitos, contribuindo para a manutenção da qualidade durante todo o ciclo de vida do desenvolvimento do software.

## **3. Trabalhos Relacionados**

Nesta seção, são apresentados cinco trabalhos relacionados que abordam questões semelhantes ou correlatas ao tema proposto neste estudo, que trata do desenvolvimento e avaliação de um framework de automação de testes no ciclo de vida do software. Esses artigos foram selecionados com base na relevância de suas contribuições ao tema de automação de testes, inteligência artificial aplicada, integração de frameworks e eficiência em ambientes ágeis.

### **3.1. Inteligência Artificial Aplicada à Qualidade de Software**

O primeiro trabalho discutido é o artigo "Inteligência Artificial Aplicada à Qualidade de Software", publicado pela Revista FT (2024). Este artigo explora como a aplicação de Inteligência Artificial (IA) tem impactado positivamente a automação de testes e o monitoramento de desempenho de software, integrando-se aos processos tradicionais de desenvolvimento. A relação com o trabalho proposto está na utilização de IA para aprimorar a eficiência dos testes automatizados, objetivo similar ao framework em desenvolvimento neste trabalho. A contribuição deste artigo é especialmente importante para justificar o uso de técnicas de IA em automação de testes, mostrando que há um avanço crescente e oportunidades de aplicação no cenário atual.

### **3.2. Transformação da Automação de Testes com IA**

Outro trabalho relevante é o estudo publicado pela Engage AI (2024), "Como a IA está transformando a automação de testes: técnicas e ferramentas". Este artigo aborda as técnicas e ferramentas que utilizam IA para criar scripts de testes mais eficientes e identificar erros de forma mais ágil. A abordagem descrita neste artigo complementa os objetivos deste estudo, que também busca melhorar a eficiência dos testes automatizados através de práticas inovadoras e mais inteligentes. O estudo fornece uma visão aprofundada sobre como a IA pode ser usada para aumentar a capacidade de detecção de defeitos e otimizar o processo de automação, proporcionando insights importantes para o desenvolvimento do framework.

### **3.3. Expansão da Capacidade do KaneAI**

O terceiro trabalho analisado é o artigo "LambdaTest expande capacidade do KaneAI para aprimorar a eficiência e flexibilidade dos testes", publicado pelo Estado de Minas - Mundo Corporativo (2024). O artigo destaca como a expansão da capacidade do KaneAI visa aprimorar a eficiência e a flexibilidade dos testes automatizados, aspectos que são fundamentais para ambientes de desenvolvimento ágeis. Esse estudo é relevante para o trabalho proposto, pois apresenta práticas que podem ser incorporadas ao framework de automação para melhorar a escalabilidade e flexibilidade dos testes. Essa flexibilidade é crucial em projetos ágeis onde mudanças são frequentes, e o framework proposto busca atender a essas necessidades.

### **3.4. Frameworks de Automação de Testes para 2024**

O artigo "12 frameworks de Automação de testes de software para utilizar em 2024", publicado pela Teste.ai (2024), apresenta uma variedade de frameworks que podem ser utilizados na automação de testes de software, cada um com características específicas que atendem a diferentes necessidades de projetos. A escolha do framework adequado, como destacado no artigo, é crucial para garantir a eficiência e qualidade dos testes automatizados, sendo um fator fundamental para o desenvolvimento do framework proposto neste trabalho. Este artigo também ajuda a contextualizar o estado da arte na escolha de ferramentas e métodos para automação, demonstrando as opções e limitações que estão disponíveis no mercado atualmente.

### **3.5. Automação de Testes e DevOps**

Por fim, o trabalho "A sintonia perfeita: automação de testes e DevOps", publicado pela Alura (2023), explora como a integração de práticas de DevOps com a automação de testes pode melhorar a qualidade do software. A colaboração entre equipes de QA e DevOps facilita a automação de diversos tipos de testes, permitindo a identificação e resolução de problemas antes da implantação do software. Esta abordagem é particularmente relevante para o framework proposto, que visa melhorar a integração entre testes automatizados e o ciclo de desenvolvimento contínuo. O artigo fornece uma base sólida sobre como DevOps e automação de testes podem operar juntos para maximizar a entrega contínua de qualidade.

## **4. Materiais e Métodos**

A relação desta pesquisa com trabalhos relacionados é fundamental na determinação dos materiais e métodos empregados neste estudo. A abordagem adotada se baseia nas técnicas e práticas já estabelecidas, conforme detalhado na seção anterior.

### **4.1. Procedimentos**

Esta pesquisa conduz um estudo aplicado, com o objetivo de coletar dados quantitativos e qualitativos acerca da utilização de um framework de automação de testes no desenvolvimento de software em ambientes ágeis. A justificativa da utilização desse método advém dos resultados encontrados em trabalhos relacionados, que indicam a necessidade de aprimoramento na eficiência e qualidade dos testes automatizados. Assim, foi iniciado um teste piloto em um ambiente de desenvolvimento simulado, utilizando o framework

Selenium e a plataforma Spring Boot, ambos amplamente utilizados em automação de testes e desenvolvimento de aplicações web.

O ambiente de desenvolvimento foi configurado para incluir a implementação do framework em Java, utilizando o Spring Boot como base. O Selenium foi escolhido devido à sua flexibilidade e ao suporte que oferece para a criação e execução de testes em aplicações web, sendo uma ferramenta consolidada e de código aberto. A implementação e a execução dos testes foram realizadas em grupos de desenvolvedores subdivididos para testar diferentes abordagens, incluindo automação com e sem a utilização de metodologias ágeis.

Durante o experimento piloto, observou-se que os desenvolvedores enfrentaram desafios relacionados à configuração do ambiente de desenvolvimento e à integração com o framework. Esses desafios foram documentados e usados para aprimorar a segunda execução do experimento, na qual os desenvolvedores receberam suporte adicional e instruções detalhadas para superar as dificuldades iniciais.

## 4.2. Ferramentas

No decorrer da pesquisa, foram utilizados o Selenium para automação de testes e o Spring Boot para desenvolvimento da aplicação web. O ambiente de desenvolvimento foi preparado utilizando Java (versão JDK 21.0.1), com os repositórios criados e disponibilizados na plataforma GitHub. Os desenvolvedores puderam clonar o repositório e criar suas próprias versões a partir do modelo fornecido. Além disso, foi utilizado o framework JUnit para a criação e execução dos testes automatizados.

Após a reestruturação do ambiente de desenvolvimento, foi utilizada também a linguagem Python em conjunto com o Google Colab, para facilitar a colaboração e o gerenciamento dos experimentos. As ferramentas de testes unitários unittest e MutPy foram empregadas para a realização dos testes de mutação e verificação dos resultados. Essa abordagem proporcionou maior controle e gerenciamento dos dados produzidos pelos desenvolvedores, bem como uma visão detalhada das métricas de avaliação.

## 4.3. Métricas de Avaliação dos Resultados

Para a avaliação das métricas, são aplicados os seguintes critérios:

**Tempo de execução dos testes:** Avaliar a eficiência do framework na execução dos testes automatizados, comparando o tempo necessário para a execução dos testes manuais e automatizados.

**Produtividade dos desenvolvedores:** Medir o impacto da automação na produtividade dos desenvolvedores, incluindo a quantidade de funcionalidades testadas e entregues em cada ciclo de desenvolvimento.

**Capacidade de detecção de falhas:** Avaliar a capacidade dos testes automatizados de identificar bugs previamente inseridos no código. Essa métrica visa mensurar a eficácia dos testes em garantir a qualidade do software.

**Resiliência dos testes a mudanças nos requisitos:** Introduzir mudanças nos requisitos durante o ciclo de desenvolvimento para avaliar se os testes se mantêm eficazes diante de alterações no código.

#### 4.4. Cronograma de Execução

O cronograma de execução das atividades foi dividido em etapas quinzenais, abrangendo desde o levantamento de requisitos até o desenvolvimento, teste e validação do framework. As etapas incluem:

**Primeira quinzena de fevereiro/2025:** Levantamento de requisitos e definição do escopo do framework;

**Segunda quinzena de fevereiro/2025:** Configuração do ambiente de desenvolvimento e escolha das ferramentas;

**Primeira quinzena de março/2025:** Desenvolvimento dos módulos iniciais de automação;

**Segunda quinzena de março/2025:** Testes preliminares e coleta de feedback;

**Primeira quinzena de abril/2025:** Aperfeiçoamento dos módulos e execução de testes adicionais;

**Segunda quinzena de abril/2025:** Análise dos resultados e ajustes no framework;

**Primeira quinzena de maio/2025:** Desenvolvimento das funcionalidades finais e integração;

**Segunda quinzena de maio/2025:** Testes de validação e coleta de dados para análise;

**Primeira quinzena de junho/2025:** Análise dos dados e conclusão do estudo;

**Segunda quinzena de junho/2025:** Finalização do relatório e apresentação dos resultados.

#### 4.5. Resultados Preliminares

Alguns testes preliminares já foram realizados utilizando o framework Selenium em um ambiente de teste controlado. Os resultados indicaram uma redução significativa no tempo de execução dos testes automatizados em comparação aos testes manuais, além de uma melhoria na cobertura dos testes. O código inicial do framework também foi desenvolvido e testado em um cenário simulado, mostrando-se promissor para futuras implementações e validações. Esses resultados iniciais servem como base para ajustes e melhorias nas próximas etapas da pesquisa.

#### Referências

- [Alura 2023] Alura (2023). A sintonia perfeita: automação de testes e devops.
- [ClickUp 2024] ClickUp (2024). 20 melhores ferramentas de teste de automação 2024.
- [LogAp 2024] LogAp (2024). Ferramentas de teste automatizado: 5 opções e como usá-las.
- [Marques Borges and Oliveira Lima 2024] Marques Borges, G. D. and Oliveira Lima, S. (2024). Inteligência artificial aplicada à qualidade de software. *Revista FT*, 28(138).
- [MIT Technology Review Brasil 2024] MIT Technology Review Brasil (2024). A evolução dos testes de software através da ia.
- [Objective 2022] Objective (2022). Framework de testes automatizados: o que é e quais os benefícios.

[Prime Control 2017] Prime Control (2017). 10 ferramentas de automação de testes mais usadas.

[Prime Control 2024] Prime Control (2024). 6 ferramentas de automação de testes com as que você precisa conhecer.

[ProgramaThor 2024] ProgramaThor (2024). Frameworks para automação de testes.

[Teste.ai 2024] Teste.ai (2024). 12 frameworks de automação de testes de software para utilizar em 2024.