

TESTES DE SOFTWARE: OS BENEFÍCIOS DA IMPLEMENTAÇÃO DE TESTES AUTOMATIZADOS

Alisson Zaroni Cesario da Silva¹, Matheus Leandro Ferreira²

Resumo: A crescente demanda por softwares de qualidade tem impulsionado a adoção de processos de validação mais eficientes, tornando os testes de software fundamentais para a garantia da qualidade. Neste contexto, este trabalho teve como objetivo comparar a eficiência dos testes manuais e automatizados com base em métricas de tempo de execução e identificação de falhas. Para isso, foi realizada uma pesquisa quantitativa e experimental utilizando a plataforma Practice Automation Testing, na qual foram elaborados 70 cenários de teste executados manualmente por três profissionais de Quality Assurance com diferentes níveis de experiência e automatizados por meio do framework Cypress. Os resultados demonstraram que a automação apresentou desempenho superior em velocidade de execução e detecção de defeitos, identificando todos os bugs inseridos nos cenários de teste. Em contrapartida, os testes manuais realizados por profissionais mais experientes contribuíram para a identificação de comportamentos não previstos nos casos documentados. Conclui-se que os testes automatizados aumentam a produtividade, a consistência e a cobertura das validações, enquanto os testes manuais permanecem importantes para análises exploratórias e qualitativas. Como trabalhos futuros, recomenda-se a ampliação do estudo para ambientes de maior complexidade e integração contínua.

Palavras-chave: testes de software; automação de testes; qualidade de software.

ABSTRACT: The growing demand for quality software has driven the adoption of more efficient validation processes, making software testing fundamental to quality assurance. In this context, this study aimed to compare the efficiency of manual and automated testing based on metrics of execution time and defect identification. A quantitative and experimental study

¹Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense (Unesc), Criciúma - Santa Catarina - Brasil. alisson.zanonii@gmail.com

²Orientador. Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense (Unesc), Criciúma - Santa Catarina - Brasil. mlf@unesc.net

was conducted using the Practice Automation Testing platform, in which 70 test scenarios were developed, executed manually by three Quality Assurance professionals with different levels of experience, and automated using the Cypress framework. The results showed that automation outperformed manual testing in execution speed and defect detection, identifying all bugs inserted in the test scenarios. In contrast, manual tests performed by more experienced professionals contributed to identifying behaviors not anticipated in the documented cases. It is concluded that automated tests increase productivity, consistency, and validation coverage, while manual tests remain important for exploratory and qualitative analyses. As future work, it is recommended to expand the study to environments of greater complexity and continuous integration.

Keywords: software testing; test automation; software quality.

1 INTRODUÇÃO

A crescente demanda por softwares de qualidade, aliada à necessidade de entregas cada vez mais rápidas, tem levado as organizações a investir em processos mais eficientes de validação. Nesse contexto, a automação de testes consolidou-se como uma das principais estratégias para aumentar a velocidade das verificações e reduzir a ocorrência de falhas em produção. Entretanto, apesar de sua ampla adoção e dos benefícios frequentemente apontados pela literatura, ainda permanece uma questão relevante: em cenários equivalentes de execução, até que ponto uma suíte automatizada é mais eficiente do que a execução manual realizada por profissionais com diferentes níveis de experiência? A resposta a essa questão torna-se importante para orientar equipes de desenvolvimento na definição das estratégias de testes mais adequadas para seus projetos.

A identificação precoce de falhas constitui um dos principais fatores para garantir a qualidade do software, contribuindo para a redução dos custos de manutenção, o aumento da confiabilidade do produto e a minimização de impactos negativos para usuários e organizações. Nesse contexto, os testes de software desempenham papel fundamental ao assegurar que o sistema atenda aos requisitos estabelecidos e funcione corretamente em diferentes condições de uso (Silva, 2019). Além disso, a evolução das tecnologias de desenvolvimento e a adoção de ciclos de entrega cada vez mais curtos têm ampliado a necessidade de mecanismos capazes de validar continuamente as funcionalidades implementadas, reduzindo o risco de

falhas em produção e contribuindo para a estabilidade dos sistemas (Souza et al., 2023).

Os testes de software têm como objetivo verificar se um sistema executa corretamente as funções para as quais foi desenvolvido, além de identificar defeitos antes de sua disponibilização aos usuários finais. Em modelos modernos de desenvolvimento, a realização contínua dessas atividades tornou-se essencial para garantir a qualidade do produto e reduzir custos associados à correção de falhas em etapas posteriores do ciclo de vida do software (Sommerville, 2011). Nesse cenário, tanto os testes manuais quanto os automatizados passaram a desempenhar papel estratégico nos processos de desenvolvimento, cada um apresentando características, benefícios e limitações específicas (Pressman, 2011). A compreensão das diferenças entre essas abordagens torna-se fundamental para avaliar em quais contextos a automação oferece vantagens mais expressivas e em quais situações a atuação humana continua sendo indispensável.

Embora diversos estudos investiguem ferramentas e abordagens de automação de testes, ainda existem discussões sobre os ganhos efetivos proporcionados por essas soluções quando comparadas diretamente à atuação humana em cenários equivalentes de execução. A compreensão dessas diferenças é relevante para apoiar a tomada de decisão das equipes de desenvolvimento quanto à adoção e utilização de estratégias de automação em projetos de software.

A qualidade de software constitui um dos principais fatores para o sucesso de sistemas computacionais, uma vez que está diretamente relacionada à capacidade de atender às necessidades dos usuários e aos requisitos estabelecidos durante o desenvolvimento. Em um cenário caracterizado pela crescente dependência de soluções digitais, falhas de software podem gerar prejuízos financeiros, comprometer a segurança das informações e impactar negativamente a experiência dos usuários. Dessa forma, a busca pela qualidade tornou-se uma preocupação constante das organizações que desenvolvem e mantêm sistemas de software.

De acordo com Sommerville (2011), assegurar que um sistema atenda aos seus requisitos funcionais e não funcionais é um dos principais objetivos dos testes de software, sendo esse processo fundamental para garantir confiabilidade, robustez e manutenibilidade. Nesse contexto, a qualidade não deve ser tratada como uma atividade isolada ao final do desenvolvimento, mas como um processo contínuo presente em todas as etapas do ciclo de vida do software.

A Garantia da Qualidade (Quality Assurance, QA) representa um conjunto de práticas, métodos e procedimentos voltados à prevenção de defeitos e à melhoria contínua dos processos de desenvolvimento. Segundo Pressman (2011), o QA possui um escopo mais amplo do que os testes de software, envolvendo atividades de auditoria, revisão técnica, monitoramento e controle dos processos utilizados pelas equipes de desenvolvimento. Enquanto os testes têm como objetivo validar o comportamento do sistema, a garantia da qualidade busca assegurar que os processos adotados contribuam para a construção de produtos mais confiáveis e alinhados às necessidades dos usuários.

Nesse sentido, os testes de software desempenham papel essencial dentro das estratégias de QA, fornecendo evidências objetivas sobre o funcionamento da aplicação e permitindo identificar falhas antes que elas sejam disponibilizadas aos usuários finais. A integração dessas atividades ao longo do desenvolvimento contribui para a redução de riscos, melhoria da qualidade do produto e aumento da confiança das equipes e dos clientes em relação ao sistema desenvolvido.

Os testes de software podem ser classificados de acordo com diferentes critérios, sendo a classificação por níveis uma das mais utilizadas na Engenharia de Software. Conforme Sommerville (2011), os principais níveis de teste são os testes de unidade, integração, sistema e aceitação, cada um responsável por validar aspectos específicos do software durante o ciclo de desenvolvimento.

Os testes de unidade constituem o nível mais básico de validação, sendo utilizados para verificar individualmente funções, métodos ou componentes específicos da aplicação. Após essa etapa, os testes de integração têm como finalidade avaliar a comunicação entre diferentes módulos do sistema, garantindo que os componentes funcionem corretamente quando combinados. Os testes de sistema analisam o comportamento da aplicação de forma completa, considerando o ambiente operacional e a interação entre seus diversos elementos. Por fim, os testes de aceitação buscam verificar se o software atende às expectativas e necessidades dos usuários ou clientes, servindo como etapa final de validação antes da implantação.

Além da classificação por níveis, os testes também podem ser divididos em funcionais e não funcionais. Os testes funcionais concentram-se na verificação dos requisitos e regras de negócio implementados, avaliando se as funcionalidades desenvolvidas produzem os resultados esperados.

Já os testes não funcionais têm como objetivo analisar atributos relacionados ao desempenho, segurança, usabilidade, compatibilidade, confiabilidade e escalabilidade do sistema (Myers, 2012).

Nos modelos modernos de desenvolvimento, especialmente aqueles baseados em metodologias ágeis, a execução contínua dessas diferentes categorias de teste tornou-se essencial para garantir a qualidade das entregas. Nesse contexto, testes unitários, de integração e de regressão são frequentemente considerados candidatos ideais para automação devido à sua elevada repetibilidade e estabilidade ao longo do projeto (Gandini; Salvatores, 2002).

Os testes de software podem ser executados por meio de abordagens manuais ou automatizadas, sendo ambas amplamente utilizadas pelas equipes de desenvolvimento e qualidade. A escolha entre essas abordagens depende de fatores como objetivos do projeto, frequência de execução dos testes, recursos disponíveis e características do sistema avaliado.

Os testes manuais consistem na execução direta dos cenários por um profissional responsável pela validação da aplicação. Nessa abordagem, o analista interage com o sistema, realiza ações previamente definidas e verifica se os resultados observados correspondem ao comportamento esperado. Segundo Pressman (2011), os testes manuais são particularmente úteis em atividades exploratórias, avaliações de usabilidade e situações que exigem análise crítica, interpretação contextual e tomada de decisão humana.

Apesar de sua importância, os testes manuais apresentam algumas limitações. A execução repetitiva de cenários pode demandar grande quantidade de tempo e esforço, especialmente em projetos que possuem ciclos frequentes de atualização. Além disso, fatores humanos, como fadiga, distração ou diferenças de interpretação, podem influenciar os resultados obtidos e gerar inconsistências durante as validações (Gandini; Salvatores, 2002).

Como alternativa a essas limitações, a automação de testes tem sido amplamente adotada pelas organizações. Os testes automatizados consistem na criação de scripts capazes de reproduzir automaticamente os passos definidos em cenários de teste, executando verificações sem necessidade de intervenção humana durante sua execução. Essa abordagem proporciona maior rapidez, repetibilidade e padronização, permitindo que os testes sejam executados diversas vezes ao longo do ciclo de desenvol-

vimento com resultados consistentes (Sommerville, 2011).

Entre os principais benefícios da automação destacam-se a redução do tempo necessário para execução dos testes, o aumento da cobertura de validação e a capacidade de identificar rapidamente falhas introduzidas durante modificações no sistema. Em ambientes de integração contínua e entrega contínua, a automação desempenha papel fundamental ao fornecer feedback rápido para as equipes de desenvolvimento, contribuindo para a detecção precoce de defeitos e para a manutenção da qualidade do software (Souza et al., 2023).

Entretanto, a automação não elimina completamente a necessidade de testes manuais. Embora seja altamente eficiente para validações repetitivas e verificações previamente definidas, ela possui limitações relacionadas à identificação de problemas subjetivos, comportamentos inesperados e questões de experiência do usuário. Dessa forma, a literatura destaca que as abordagens manual e automatizada devem ser compreendidas como estratégias complementares, e não excludentes, dentro dos processos de garantia da qualidade.

Diversas ferramentas têm sido desenvolvidas para apoiar a automação de testes em aplicações web. Entre as mais utilizadas destacam-se Selenium, Cypress e Robot Framework, que oferecem recursos para simulação de interações com usuários, validação de elementos de interface e geração de relatórios de execução. Essas ferramentas permitem automatizar diferentes tipos de cenários e são amplamente empregadas em ambientes de desenvolvimento modernos (Souza et al., 2023). Além disso, estudos recentes apontam que projetos que adotam automação de testes apresentam melhorias significativas em aspectos relacionados à estabilidade, manutenção e escalabilidade dos sistemas (Matos, 2020).

Com o objetivo de compreender o estado da arte relacionado à automação de testes de software, foram analisadas pesquisas correlatas que investigam ferramentas, estratégias e impactos da automação em diferentes contextos de desenvolvimento.

A automação de testes tem sido amplamente investigada na literatura devido à sua relevância para a garantia da qualidade de software e para a redução do esforço associado às atividades de validação. Diversos estudos têm buscado comparar ferramentas, avaliar estratégias de automação e analisar os impactos dessa abordagem em ambientes reais de desenvolvimento.

Entre os trabalhos relacionados à comparação de ferramentas

de automação, destaca-se o estudo de Teixeira (2022), que realizou uma análise comparativa entre as ferramentas Capybara e Cypress aplicadas a testes de software. A pesquisa teve como objetivo identificar diferenças relacionadas à facilidade de implementação, desempenho e produtividade durante a execução dos testes. Os resultados indicaram vantagens do Cypress em aspectos associados à agilidade de desenvolvimento, simplicidade de configuração e velocidade de execução, demonstrando seu potencial para utilização em aplicações web modernas.

De forma semelhante, Soares (2022) conduziu uma comparação entre Cypress e Selenium em cenários de testes de ponta a ponta (*end-to-end*). O estudo avaliou características relacionadas à performance, integração com o ambiente de desenvolvimento e facilidade de utilização. Os resultados apontaram que o Cypress apresentou melhor desempenho em diversas situações analisadas, além de proporcionar uma experiência de desenvolvimento mais simplificada quando comparado ao Selenium.

Complementando essas investigações, Farias (2024) realizou uma análise comparativa envolvendo três das principais ferramentas de automação utilizadas atualmente: Selenium WebDriver, Cypress e Playwright. O objetivo da pesquisa foi identificar vantagens e limitações de cada solução considerando critérios relacionados à manutenção, desempenho, compatibilidade e recursos disponíveis. Os resultados evidenciaram que o Playwright apresenta características promissoras para aplicações web modernas, ao mesmo tempo em que confirmaram a relevância do Cypress como uma das ferramentas mais eficientes para automação de testes funcionais.

Além das comparações entre ferramentas, alguns estudos analisaram os impactos da automação em ambientes reais de desenvolvimento. Matos (2020) investigou a aplicação de testes automatizados no sistema e-SUS AB Atividade Coletiva em conjunto com práticas de integração contínua. O estudo teve como objetivo avaliar a influência da automação na estabilidade do software e no processo de entrega de novas versões. Os resultados demonstraram melhorias significativas na qualidade das entregas, redução de falhas em produção e maior agilidade na obtenção de feedback durante o desenvolvimento.

A evolução das tecnologias aplicadas à automação também tem motivado novas pesquisas. Nesse contexto, Ferreira et al. (2025) investigaram o uso de modelos de linguagem de grande escala (Large Language Models, LLMs) para geração automática de testes utilizando o framework Cypress. O estudo buscou avaliar a capacidade dos modelos de inteligên-

cia artificial em produzir cenários de teste válidos a partir de documentação e requisitos. Os resultados obtidos indicaram aproximadamente 92% de aproveitamento dos testes gerados, evidenciando o potencial da inteligência artificial como ferramenta de apoio aos processos de qualidade de software.

Por sua vez, Lima (2017) realizou um estudo de caso utilizando Selenium para automação de testes funcionais em sistemas web com base no modelo MPT-Br. A pesquisa teve como objetivo avaliar os impactos da automação na detecção de defeitos e na produtividade das atividades de validação. Os resultados demonstraram aumento na taxa de identificação de falhas e redução do tempo necessário para execução dos testes, reforçando os benefícios da automação em ambientes de desenvolvimento de software.

A análise dos trabalhos apresentados demonstra que a literatura tem concentrado esforços principalmente na comparação entre ferramentas de automação e na avaliação dos benefícios proporcionados pelos testes automatizados em diferentes contextos de aplicação. Entretanto, observa-se que a maioria dos estudos está direcionada à análise de tecnologias específicas ou à avaliação da automação em ambientes organizacionais, havendo menor quantidade de pesquisas voltadas à comparação direta entre a execução humana e a execução automatizada sob condições equivalentes.

Dessa forma, identifica-se uma lacuna relacionada à avaliação comparativa do desempenho de profissionais de Quality Assurance com diferentes níveis de senioridade em relação a uma suíte automatizada construída sobre os mesmos cenários de teste. Nesse contexto, o presente estudo busca contribuir para a literatura ao analisar simultaneamente métricas de tempo de execução, capacidade de detecção de falhas e ocorrência de falsos positivos em testes manuais e automatizados, permitindo uma compreensão mais abrangente das vantagens e limitações de cada abordagem.

Para alcançar esse objetivo, busca-se compreender a importância dos testes automatizados no desenvolvimento de software, comparar o tempo de execução entre abordagens manuais e automatizadas, analisar a capacidade de detecção de falhas e a ocorrência de falsos positivos em ambas as abordagens e avaliar a contribuição da automação para os processos modernos de garantia da qualidade de software.

Além desta introdução, o artigo está organizado em três seções

principais. A Seção 2 apresenta os Materiais e Métodos empregados na pesquisa. A Seção 3 apresenta os Resultados e Discussão. Por fim, a Seção 4 apresenta as conclusões e perspectivas para trabalhos futuros.

2 MATERIAIS E MÉTODOS

Esta pesquisa caracteriza-se como um estudo experimental de abordagem quantitativa, desenvolvido com o objetivo de comparar a eficiência dos testes manuais e automatizados no contexto da garantia da qualidade de software. A investigação foi conduzida por meio da execução controlada de cenários de teste em uma aplicação web, permitindo a coleta de métricas relacionadas ao tempo de execução, à capacidade de detecção de falhas e à ocorrência de falsos positivos.

2.1 AMBIENTE EXPERIMENTAL

Os experimentos foram realizados utilizando a plataforma web Practice Automation Testing, amplamente empregada para fins educacionais e validação de técnicas de automação de testes. A aplicação foi selecionada por disponibilizar funcionalidades típicas de sistemas de comércio eletrônico, contemplando operações de navegação, autenticação, cadastro, gerenciamento de carrinho de compras e validações de interface.

Para a execução automatizada dos testes foi utilizado o framework Cypress, ferramenta amplamente adotada no desenvolvimento de aplicações web devido à sua integração nativa com navegadores modernos, facilidade de configuração e capacidade de geração de relatórios detalhados. Neste estudo, a suíte automatizada corresponde ao conjunto de scripts desenvolvidos para executar automaticamente os 70 cenários de teste definidos no plano experimental, reproduzindo os mesmos procedimentos utilizados na execução manual. Durante sua implementação, foram utilizados comandos customizados para encapsular ações recorrentes e padronizar fluxos comuns entre diferentes cenários, facilitando a organização da suíte e garantindo a rastreabilidade entre os cenários manuais e automatizados.

2.2 PARTICIPANTES DA PESQUISA

A etapa de execução manual contou com a participação de três profissionais atuantes na área de Quality Assurance (QA), representando diferentes níveis de senioridade profissional:

- QA Júnior;

- QA Pleno;
- QA Sênior.

A inclusão de profissionais com diferentes níveis de experiência teve como objetivo avaliar a influência da senioridade na identificação de defeitos e no desempenho da execução dos testes. Para preservar a identidade dos participantes, seus nomes e informações pessoais não foram divulgados.

2.3 ELABORAÇÃO DOS CENÁRIOS DE TESTE

Foi elaborado um plano contendo 70 cenários de teste, estruturados a partir das funcionalidades disponibilizadas pela plataforma analisada. Os cenários foram documentados de forma padronizada, contendo identificação, descrição, pré-condições, passos de execução e resultado esperado.

Dos 70 cenários desenvolvidos, 63 representavam comportamentos considerados corretos da aplicação, enquanto 7 cenários foram intencionalmente construídos para representar situações de falha previamente conhecidas, caracterizando casos de bug-tracking. Esses cenários permitiram avaliar a capacidade de detecção de defeitos dos participantes e da automação.

Todos os cenários utilizados na automação reproduziram exatamente os mesmos procedimentos definidos para a execução manual, assegurando equivalência metodológica entre as abordagens comparadas.

2.4 IMPLEMENTAÇÃO DA AUTOMAÇÃO

A automação dos testes foi desenvolvida utilizando o framework Cypress. Cada cenário manual foi convertido em scripts automatizados que reproduziam integralmente os passos documentados no plano de testes.

Durante o desenvolvimento foram implementados comandos customizados para encapsular ações recorrentes e padronizar fluxos compartilhados entre diferentes cenários de teste, favorecendo a organização da suíte, reduzindo a duplicação de implementações e facilitando sua manutenção. A suíte automatizada foi submetida a validações preliminares para garantir que os resultados produzidos refletissem fielmente os critérios estabelecidos nos cenários originais.

O esforço necessário para implementação da suíte correspondeu a 23 horas e 40 minutos de trabalho, contemplando atividades de con-

figuração do ambiente, codificação dos testes, ajustes de execução e validação dos resultados. Esse tempo não foi considerado na comparação direta das execuções, sendo utilizado apenas para análise de viabilidade e amortização do investimento em automação.

2.5 PROCEDIMENTOS DE COLETA DE DADOS

A coleta de dados ocorreu em duas etapas distintas.

Na primeira etapa, os três profissionais executaram manualmente os 70 cenários de teste. Durante essa atividade foram registrados:

- tempo total de execução;
- identificação dos defeitos existentes;
- quantidade de bugs não detectados;
- ocorrência de falsos positivos;
- observações registradas pelos participantes.

Na segunda etapa, a suíte automatizada foi executada utilizando os mesmos cenários. Foram coletadas as mesmas métricas utilizadas na execução manual, permitindo a realização de uma comparação direta entre as abordagens.

Os resultados produzidos pela automação foram utilizados como referência para validação da execução dos cenários de bug-tracking, uma vez que todos os defeitos previamente planejados foram corretamente identificados pela suíte automatizada.

2.6 MÉTRICAS DE AVALIAÇÃO

A comparação entre testes manuais e automatizados foi realizada com base nas seguintes métricas:

- **Tempo de execução:** corresponde ao período necessário para conclusão da suíte completa de testes.
- **Taxa de detecção de falhas:** representa a proporção de defeitos corretamente identificados em relação ao total de bugs inseridos nos cenários.
- **Bugs perdidos:** quantidade de defeitos existentes que não foram identificados durante a execução.

- **Falsos positivos:** cenários classificados como falhos apesar de estarem corretos segundo o resultado esperado.
- **Precisão:** relação entre os defeitos efetivamente identificados e o total de registros classificados como falhas.

2.7 ANÁLISE DOS DADOS

Os dados coletados foram organizados em planilhas eletrônicas e analisados por meio de estatística descritiva. Foram calculados tempos médios de execução, taxas de detecção de falhas, quantidade de falsos positivos e indicadores de precisão para cada participante e para a suíte automatizada.

A análise comparativa permitiu avaliar as vantagens e limitações de cada abordagem, bem como identificar em quais situações a automação apresentou desempenho superior e em quais contextos a experiência humana agregou valor por meio da análise exploratória e da interpretação dos comportamentos observados durante os testes.

3 RESULTADOS E DISCUSSÃO

Nesta seção são apresentados e discutidos os resultados obtidos a partir da comparação entre testes manuais e automatizados. O experimento foi conduzido por meio da execução de 70 cenários de teste na plataforma *Practice Automation Testing*, sendo 63 cenários funcionais e 7 cenários desenvolvidos especificamente para avaliação da capacidade de detecção de falhas. Os testes manuais foram executados por três profissionais de Quality Assurance (QA) com diferentes níveis de senioridade, enquanto os testes automatizados foram implementados e executados utilizando o framework Cypress.

3.1 TEMPO DE EXECUÇÃO

O tempo de execução representa uma das métricas mais relevantes para avaliação da eficiência operacional das abordagens investigadas. Em ambientes modernos de desenvolvimento, a capacidade de executar testes rapidamente influencia diretamente a obtenção de feedback contínuo e a velocidade de entrega de novas versões do software.

Tabela 1 - Comparativo de tempo de execução (70 cenários)

Executor	Nível	Tempo Total	Tempo médio/cenário
QA Júnior	Júnior	1h 35min	≈ 1min 21s
QA Pleno	Pleno	2h 23min	≈ 2min 02s
QA Sênior	Sênior	2h 00min	≈ 1min 42s
Cypress	Automação	28min 35s	≈ 24s

Fonte: Do autor.

Os dados apresentados na Tabela 1 demonstram que a automação concluiu os 70 cenários em apenas 28 minutos e 35 segundos, tempo significativamente inferior ao registrado pelos participantes humanos. Entre os profissionais, o QA júnior apresentou o menor tempo de execução manual, totalizando 1 hora e 35 minutos, enquanto o QA pleno registrou o maior tempo, com 2 horas e 23 minutos. O QA sênior executou os cenários em 2 horas.

A diferença observada evidencia uma das principais vantagens da automação de testes: a capacidade de executar validações repetitivas de forma rápida, padronizada e escalável. Enquanto os profissionais precisaram analisar resultados, registrar evidências e tomar decisões durante a execução dos cenários, a suíte automatizada reproduziu todas as verificações previamente programadas sem interrupções ou variações de desempenho.

Sob a perspectiva da Engenharia de Software, essa redução no tempo de execução possui impacto direto em ambientes de integração contínua e entrega contínua, nos quais a obtenção rápida de feedback constitui fator essencial para a identificação precoce de defeitos. Esse resultado está alinhado aos achados de Matos (2020), que observou ganhos relacionados à produtividade e à eficiência dos processos de validação em ambientes que utilizam automação de testes.

Além da velocidade, observa-se que a automação proporciona maior previsibilidade operacional. Diferentemente da execução manual, cujo desempenho pode variar em função da experiência do profissional, fadiga ou fatores contextuais, a automação executa os mesmos procedimentos de maneira consistente a cada nova execução. Essa característica torna os testes automatizados particularmente adequados para validações frequentes e atividades de regressão.

3.2 DETECÇÃO DE FALHAS, BUG TRACKING

A capacidade de identificar defeitos constitui um dos principais indicadores de eficácia dos testes de software. Para essa análise, foram inseridos intencionalmente sete cenários contendo comportamentos incorretos previamente documentados, permitindo avaliar a eficiência de cada executor na identificação das falhas presentes no sistema.

Tabela 2 - Detecção de bugs por cenário e executor

ID	Descrição resumida	Júnior	Pleno	Sênior	Cypress
HP-02	Arrivals: quatro itens esperados	Sim	Não	Sim	Sim
SH-10	Destaque de promoção na 1ª fileira	Não	Não	Sim	Sim
LG-08	<i>Remember me</i> marcado por padrão	Sim	Sim	Sim	Sim
RG-03	Botão <i>Register</i> ativo com e-mail vazio	Não	Não	Sim	Sim
RG-04	Botão <i>Register</i> ativo com senha vazia	Não	Não	Sim	Sim
RG-05	Botão <i>Register</i> ativo com campos vazios	Não	Não	Sim	Sim
CA-12	Limite máximo de produtos no carrinho	Sim	Não	Não	Sim
Total detectado		3/7	1/7	6/7	7/7

Fonte: Do autor.

Os resultados apresentados na Tabela 2 evidenciam diferenças significativas entre os executores. O Cypress identificou todos os defeitos inseridos experimentalmente, enquanto os participantes humanos apresentaram níveis distintos de desempenho. O QA sênior obteve o melhor resultado entre os profissionais, detectando seis dos sete defeitos propostos. O QA júnior identificou três falhas e o QA pleno detectou apenas um dos bugs inseridos.

Com o objetivo de consolidar os resultados relacionados à detecção de defeitos, a Tabela 3 apresenta os indicadores gerais de desempenho de cada executor.

Tabela 3 - Taxa de detecção de bugs e falsos positivos por executor

Executor	Bugs detectados	Taxa de detecção	Bugs perdidos	Falsos positivos	Precisão
Júnior	3/7	42,9%	4	3	50,0%
Pleno	1/7	14,3%	6	0	100,0%
Sênior	6/7	85,7%	1	6	50,0%
Cypress	7/7	100,0%	0	0	100,0%

Fonte: Do autor (2026).

A análise da Tabela 3 demonstra que o Cypress foi o único executor capaz de atingir taxa de detecção de 100%, sem registrar falsos positivos. Esse resultado evidencia o elevado nível de consistência proporcionado pela automação quando os critérios de validação estão claramente definidos e implementados nos scripts de teste.

Entre os participantes humanos, observa-se uma relação direta entre experiência profissional e capacidade de identificação de falhas. O QA sênior apresentou taxa de detecção de 85,7%, enquanto os desempenhos do QA júnior e do QA pleno foram significativamente inferiores. Esse comportamento sugere que a eficácia dos testes manuais depende não apenas da execução dos cenários, mas também da experiência prática, da capacidade analítica e da interpretação dos critérios de aceite por parte do profissional responsável pela validação.

Os resultados também demonstram uma característica importante dos testes automatizados: a independência em relação a fatores humanos. Diferentemente dos participantes, cujos resultados variaram significativamente mesmo diante dos mesmos cenários e condições experimentais, a automação produziu resultados determinísticos e reproduzíveis durante todas as execuções realizadas.

Esses achados reforçam uma das principais vantagens da automação destacadas na literatura, relacionada à capacidade de executar verificações de forma consistente e padronizada, reduzindo a variabilidade inerente à análise humana e aumentando a confiabilidade dos resultados obtidos.

3.3 FALSOS POSITIVOS E ANÁLISE QUALITATIVA

Além da identificação de defeitos reais, também foram analisados os falsos positivos registrados durante o experimento. Consideram-se falsos positivos os cenários classificados como falha pelos executores, embora estivessem corretos de acordo com o comportamento esperado definido no gabarito da pesquisa.

Tabela 4 - Falsos positivos registrados por executor

Executor	Qtd.	Cenários e observações registradas
Júnior	3	LG-04 (sem observação registrada); EN-09 e EN-14 (campo “Address 2” reportado como não visível)
Pleno	0	—
Sênior	6	HP-05 (redirecionamento inesperado ao clicar no carrinho vazio); SH-06 e SH-07 (ordenação de preços por filtro incorreta); LG-05 (exibição de apenas uma mensagem de validação); EN-13 (campo de estado/região como caixa de texto); CA-08 (mensagem de cupom divergente da esperada)
Cypress	0	—

Fonte: Do autor.

Os resultados indicam que o Cypress e o QA pleno não registraram falsos positivos durante a execução dos cenários. O QA júnior apresentou três ocorrências, enquanto o QA sênior registrou seis casos classificados inicialmente como falsos positivos.

Entretanto, uma análise mais detalhada das observações registradas pelo QA sênior revela uma característica importante dos testes manuais. Em diversos cenários, as marcações realizadas não decorreram de erros de interpretação simples, mas da identificação de comportamentos potencialmente problemáticos que não estavam contemplados pelos critérios automatizados de validação.

Entre as observações registradas destacam-se inconsistências relacionadas à ordenação de produtos por filtros de preço, divergências em mensagens apresentadas ao usuário e comportamentos específicos de elementos da interface. Embora essas situações não fossem classificadas como falhas segundo os critérios definidos no experimento, elas representam indícios de possíveis problemas de usabilidade ou conformidade funcional que mereceriam investigação adicional.

Esse resultado evidencia uma limitação inerente aos testes au-

tomatizados. Os scripts executam apenas verificações previamente programadas e, portanto, tendem a ignorar situações não contempladas pelos critérios implementados. Em contrapartida, profissionais experientes possuem capacidade de analisar o comportamento da aplicação de forma contextualizada, identificando inconsistências que extrapolam os requisitos formalmente documentados.

Tal característica demonstra que os testes manuais continuam desempenhando papel relevante em atividades exploratórias, avaliações de usabilidade e análises qualitativas, nas quais a interpretação humana constitui elemento fundamental para a identificação de comportamentos inesperados.

3.4 SÍNTESE DOS ACHADOS

Os resultados obtidos neste estudo demonstram que as abordagens manual e automatizada apresentam características complementares dentro dos processos de garantia da qualidade de software. Sob a perspectiva da eficiência operacional, a automação demonstrou superioridade ao executar os cenários em tempo significativamente inferior ao observado entre os participantes humanos. Além disso, apresentou elevada consistência, identificando todos os defeitos inseridos experimentalmente sem registrar falsos positivos.

Por outro lado, os testes manuais evidenciaram contribuições importantes relacionadas à análise exploratória e à identificação de comportamentos não previstos nos critérios formais de validação. O desempenho do QA sênior demonstrou que profissionais experientes são capazes de identificar situações potencialmente problemáticas mesmo quando estas não estão explicitamente contempladas pelos cenários documentados.

Os achados observados convergem com os resultados apresentados por Teixeira (2022), Soares (2022) e Matos (2020), que destacam ganhos relacionados à produtividade, velocidade de execução e confiabilidade proporcionados pela automação de testes. Ao mesmo tempo, os resultados reforçam a importância da participação humana em atividades que exigem interpretação contextual, julgamento crítico e análise qualitativa do comportamento do sistema.

Dessa forma, conclui-se que testes manuais e automatizados não devem ser compreendidos como abordagens concorrentes, mas como estratégias complementares dentro dos processos de Quality Assurance. Enquanto a automação oferece velocidade, repetibilidade e cobertura siste-

mática dos requisitos, os testes manuais agregam valor por meio da percepção humana, da análise exploratória e da identificação de comportamentos inesperados. A combinação equilibrada dessas abordagens tende a proporcionar níveis mais elevados de qualidade, confiabilidade e maturidade nos processos de desenvolvimento de software.

4 CONCLUSÃO

A crescente adoção de práticas ágeis e de processos de integração contínua tem ampliado a necessidade de mecanismos eficientes para validação de software. Nesse contexto, os testes automatizados têm assumido papel estratégico nas atividades de garantia da qualidade, tornando relevante a compreensão de seus benefícios e limitações quando comparados às abordagens tradicionais de teste manual.

Diante desse cenário, o presente estudo teve como objetivo comparar a eficiência dos testes manuais e automatizados no desenvolvimento de software, considerando métricas relacionadas ao tempo de execução, à detecção de falhas e à precisão dos resultados obtidos. Para isso, foi conduzido um experimento controlado utilizando a plataforma *Practice Automation Testing*, no qual 70 cenários de teste foram executados por três profissionais de Quality Assurance com diferentes níveis de senioridade e por uma suíte automatizada desenvolvida com o framework Cypress.

Os resultados obtidos demonstraram que a automação apresentou desempenho superior em aspectos relacionados à velocidade de execução, consistência dos resultados e capacidade de detecção de defeitos. O Cypress foi capaz de identificar integralmente os bugs inseridos experimentalmente, além de executar os cenários em tempo significativamente inferior ao necessário para os participantes humanos. Esses resultados reforçam as vantagens da automação em cenários caracterizados por alta repetitividade, necessidade de regressão frequente e integração contínua dos processos de desenvolvimento.

Por outro lado, os resultados também evidenciaram que os testes manuais continuam desempenhando papel relevante dentro das estratégias de garantia da qualidade. A análise realizada pelo QA sênior demonstrou que profissionais experientes são capazes de identificar comportamentos potencialmente problemáticos que extrapolam os critérios previamente documentados, contribuindo para avaliações exploratórias e qualitativas que dificilmente seriam reproduzidas por scripts automatizados.

Como contribuição deste estudo, destaca-se a comparação ex-

perimental entre profissionais de QA com diferentes níveis de senioridade e uma suíte automatizada construída sobre os mesmos cenários de teste, permitindo avaliar simultaneamente aspectos relacionados à eficiência operacional, capacidade de detecção de falhas e ocorrência de falsos positivos. Os resultados obtidos contribuem para uma compreensão mais abrangente das vantagens e limitações associadas às abordagens manual e automatizada, fornecendo subsídios para a tomada de decisão em projetos de software.

Entre as limitações da pesquisa, destaca-se a utilização de uma única plataforma de testes, um conjunto restrito de 70 cenários e a participação de apenas um profissional por nível de senioridade, fatores que limitam a generalização dos resultados para outros contextos de desenvolvimento. Além disso, não foram analisados aspectos relacionados ao custo de implementação e manutenção da automação ao longo do tempo, o que pode influenciar a adoção dessas soluções em ambientes reais.

Como trabalhos futuros, sugere-se a replicação do experimento em sistemas de maior complexidade e com um número mais amplo de participantes, permitindo análises estatísticas mais robustas. Recomenda-se ainda a investigação dos custos associados à manutenção de suítes automatizadas, a integração dos testes em ambientes de CI/CD e a avaliação do uso de modelos de linguagem de grande escala na geração automática de casos de teste, ampliando as possibilidades de aplicação da automação nos processos modernos de desenvolvimento de software (Ferreira et al., 2025).

Conclui-se, portanto, que testes manuais e automatizados não devem ser compreendidos como abordagens concorrentes, mas como estratégias complementares dentro dos processos de Quality Assurance. Enquanto a automação oferece ganhos expressivos de velocidade, repetibilidade e cobertura sistemática dos requisitos, os testes manuais agregam valor por meio da análise contextual, da percepção humana e da identificação de comportamentos inesperados. A utilização combinada dessas abordagens tende a proporcionar níveis mais elevados de qualidade, confiabilidade e maturidade no desenvolvimento de software.

DECLARAÇÃO DE USO DE INTELIGÊNCIA ARTIFICIAL

Os autores declaram que ferramentas de Inteligência Artificial generativa foram utilizadas exclusivamente como apoio auxiliar à elaboração deste trabalho, incluindo assistência na revisão linguística, estrutura-

ção textual e organização de ideias. Nenhuma ferramenta de IA foi utilizada para substituição da autoria intelectual, análise científica, interpretação dos resultados ou tomada de decisões acadêmicas. A responsabilidade integral pelo conteúdo, originalidade, veracidade das informações e conformidade com os princípios de integridade científica permanece integralmente atribuída aos autores.

REFERÊNCIAS

FARIAS, G. C. d. **Análise comparativa de ferramentas de testes automatizados de ponta a ponta em ambientes de aplicações web**. Tese (bachelorThesis) — Brasil, mar. 2024. Disponível em: <http://repository.ufrpe.br/handle/123456789/5850>. Acessado em: 2025-05-02.

FERREIRA, M. et al. **Acceptance Test Generation with Large Language Models: An Industrial Case Study**. 2025. Disponível em: <https://arxiv.org/abs/2504.07244>. Acessado em: 2025-05-05.

GANDINI, A.; SALVATOIRES, M. **The Physics of Subcritical Multiplying Systems**. [S.l.]: Taylor Francis, v. 39. 673-686 p. 2002.

LIMA, R. **Testes funcionais automatizados com Selenium em sistemas web: Um estudo de caso**. 2017. Disponível em: <http://repositorio.ufc.br/handle/riufc/25003>. Acessado em: 2025-04-29.

MATOS, L. **Automação de testes em ambientes com integração contínua: Estudo de caso no aplicativo e-SUS AB Atividade Coletiva**. 2020. Disponível em: <https://repositorio.ufsc.br/handle/123456789/202446>. Acessado em: 2025-05-04.

MYERS, G. J. **The Art of Software Testing**. 3. ed. [S.l.]: Wiley. ISBN 9781118031964. 2012.

PRESSMAN, R. S. **Engenharia De Software**. [S.l.]: AMGH; 7ª edição. 780 p. 2011.

SILVA, M. L. M. d. **Uma visão geral sobre automação de testes**. Revista Científica Multidisciplinar Núcleo do Conhecimento, v. 01, n. 12, p. 117–130. 2019, Disponível em: <https://www.nucleodoconhecimento.com.br/tecnologia/automacao-de-testes>. Acessado em: 2025-03-18.

SOARES, A. F. **Comparativo entre ferramentas de testes de softwares E2E em sistemas web**. 2022, Disponível em: <https://repositorio.ifpe.edu.br/xmlui/handle/123456789/954>. Acessado em: 2025-05-05.

SOMMERVILLE, I. **Engenharia de software**. [S.l.]: Editora Pearson. 548 p. 2011.

SOUZA, G. G. d. et al. **AUTOMAÇÃO DE TESTES: uma abordagem comparativa entre ferramentas**. Revista Interface Tecnológica, v. 20, n. 2, p. 100–111. ISSN 2447-0864. 2023, Disponível em: <https://revista.fatectq.edu.br/interfacetecnologica/article/view/1771>>. Acessado em: 2025-04-30.

TEIXEIRA, E. **Análise de ferramentas de automação de testes de software**. 2022, Disponível em: <http://repositorio.ufsm.br/handle/1/29791>>. Acessado em: 2025-05-05.