

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
MINAS GERAIS - *CAMPUS* BAMBUÍ E SÃO JOÃO EVANGELISTA
PÓS-GRADUAÇÃO EM GESTÃO

Felipe de Freitas e Silva
Wárlley Júnio Andrade

**UM PROCESSO ESTRUTURADO PARA IMPLEMENTAÇÃO DE
TESTES FUNCIONAIS AUTOMATIZADOS EM APLICAÇÕES WEB:
análise de ganhos operacionais**

BambuÍ - MG
2026

FELIPE DE FREITAS E SILVA
WÁRLLEY JÚNIO ANDRADE

**UM PROCESSO ESTRUTURADO PARA IMPLEMENTAÇÃO DE
TESTES FUNCIONAIS AUTOMATIZADOS EM APLICAÇÕES WEB:
análise de ganhos operacionais**

Documento apresentado ao Curso de PÓS-GRADUAÇÃO
EM GESTÃO do Instituto Federal de Educação, Ciência
e Tecnologia de Minas Gerais - *Campus* BAMBUÍ E
SÃO JOÃO EVANGELISTA para a obtenção do título de
Pós-graduando em Gestão com Ênfase em Tecnologia e
Inovação.

Orientador: Prof. Dr^o Fábio Rodrigues Martins

BambuÍ - MG
2026

Catálogo na Fonte Biblioteca IFMG - Campus Bambuí

S586p Silva, Felipe de Freitas e.

Um processo estruturado para implementação de testes funcionais automatizados em aplicações web: análise de ganhos operacionais. / Felipe de Freitas e Silva, Wárley Júnio Andrade. – 2026.

33 f.

Orientador: Dr. Fábio Rodrigues Martins.

Trabalho de Conclusão de Curso (Pós-graduação) - Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – Campus Bambuí, MG, Curso Pós-Graduação Lato Sensu em Educação para as Relações Étnico-Raciais, 2026.

1. Automação. 2. Testes funcionais. 3. Engenharia de software. I. Andrade, Wárley Júnio. II. Martins, Fábio Rodrigues. III. Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – Campus Bambuí, MG. IV. Título.


CDD 670.42755

Felipe de Freitas e Silva
Wárley Júnio Andrade


**UM PROCESSO ESTRUTURADO PARA IMPLEMENTAÇÃO DE
TESTES FUNCIONAIS AUTOMATIZADOS EM APLICAÇÕES
WEB: análise de ganhos operacionais**

Documento apresentado ao Curso de PÓS-GRADUAÇÃO
EM GESTÃO do Instituto Federal de Educação, Ciência
e Tecnologia de Minas Gerais - *Campus BAMBUÍ E
SÃO JOÃO EVANGELISTA* para a defesa do trabalho de
conclusão de curso.


Aprovado em: 06/03/2026 pela banca examinadora:

Documento assinado digitalmente
 **FABIO RODRIGUES MARTINS**
Data: 06/04/2026 12:47:01-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr^o Fábio Rodrigues Martins
Orientador

Documento assinado digitalmente
 **EBER LOPES MENDES**
Data: 06/04/2026 15:12:14-0300
Verifique em <https://validar.iti.gov.br>

Me. Eber Lopes Mendes
Examinador

Documento assinado digitalmente
 **DENIS ROCHA DE CARVALHO**
Data: 06/04/2026 13:19:29-0300
Verifique em <https://validar.iti.gov.br>

Me. Denis Rocha de Carvalho
Examinador

RESUMO

Este trabalho apresenta um relato técnico sobre a implementação gradual de testes funcionais automatizados em uma aplicação web de gestão de serviços de tecnologia da informação. A crescente demanda por entregas ágeis e aplicações confiáveis tem impulsionado a adoção da automação de testes, entretanto sua implantação ainda enfrenta desafios relacionados à estruturação do processo, definição de escopo e esforço inicial requerido. O objetivo deste estudo é propor e avaliar um processo estruturado para implementação de testes automatizados, considerando aspectos de planejamento, testabilidade e sustentabilidade do processo de qualidade. A metodologia caracteriza-se como pesquisa aplicada, com abordagem qualitativa e quantitativa, conduzida por meio de estudo de caso. Foram definidos quatro cenários funcionais executados inicialmente de forma manual e posteriormente automatizados, sendo avaliados em três ciclos controlados.

A análise comparativa considerou o tempo de execução e o esforço humano total empregado em cada ciclo. Os resultados evidenciaram redução significativa no tempo de execução dos testes automatizados, maior previsibilidade entre execuções e melhor adequação para testes de regressão recorrentes. Observou-se que, embora a automação exija maior esforço inicial, o investimento é compensado após poucos ciclos, configurando ponto de equilíbrio favorável. Como contribuição, o relato técnico apresenta um processo estruturado e replicável que integra testes manuais e automatizados de forma planejada, apoiando a tomada de decisão sobre quando e como automatizar. Conclui-se que a automação de testes funcionais deve ser tratada como iniciativa estratégica de processo, complementar aos testes manuais e capaz de elevar a maturidade da qualidade em aplicações web.

Palavras-chave: automação; testes funcionais; regressão; qualidade; engenharia de software.

ABSTRACT

This paper presents a technical report on the gradual implementation of automated functional tests in a web-based information technology service management application. The growing demand for agile deliveries and reliable web applications has driven the adoption of test automation, although its implementation still faces challenges related to process structuring and the initial effort required. The objective of this study is to propose and evaluate a structured process for implementing automated functional tests, considering planning, testability, and sustainability of the quality process. The methodology is characterized as applied research with a mixed qualitative and quantitative approach, conducted through a case study. Four functional scenarios were defined, initially executed manually and later automated, and evaluated in three controlled execution cycles.

The comparative analysis considered execution time and total human effort in each cycle. The results showed a significant reduction in execution time, greater predictability between cycles, and better suitability for recurring regression tests. Although automation requires higher initial effort, the investment is offset after a few cycles, indicating a favorable break-even point. As a contribution, this technical report proposes a structured and replicable process that integrates manual and automated testing in a planned way, supporting decision-making on when and how to automate. It is concluded that functional test automation should be treated as a strategic process initiative, complementary to manual testing, and capable of improving software quality in web applications.

Keywords: automation; functional testing; regression; quality; software engineering.

LISTA DE ILUSTRAÇÕES

Figura 1 – Pirâmide de Testes.	12
Figura 2 – Fluxograma do processo proposto para implementação de testes funcionais automatizados em aplicações web.	25

LISTA DE TABELAS

Tabela 1 – <i>Checklist</i> de Testabilidade aplicado aos cenários.	18
Tabela 2 – Aplicação do <i>checklist</i> ao cenário “Abertura de chamado”.	18
Tabela 3 – Tempo total de execução por ciclo (testes manuais e automatizados).	21
Tabela 4 – Esforço total investido por atividade (E_total).	23
Tabela 5 – Síntese comparativa entre testes funcionais manuais e automatizados (com base nos resultados empíricos).	24

SUMÁRIO

1	INTRODUÇÃO	9
1.1	Objetivos	10
<i>1.1.1</i>	<i>Objetivo Geral</i>	<i>10</i>
<i>1.1.2</i>	<i>Objetivos Específicos</i>	<i>10</i>
1.2	Justificativa	10
1.3	Organização do Texto	10
2	FUNDAMENTAÇÃO TEÓRICA	12
2.1	Testes de Software e Garantia da Qualidade	12
2.2	Testes Funcionais Manuais e suas Limitações	13
2.3	Automação de Testes Funcionais em Aplicações Web	13
2.4	Testabilidade e Adoção Gradual da Automação	14
2.5	Ferramentas de Automação e Enfoque em Processo	14
3	METODOLOGIA	16
3.1	Ambiente computacional e ferramentas	16
3.2	Planejamento, seleção e priorização dos cenários de teste	17
3.3	Checklist de testabilidade	17
3.4	Procedimento de execução	19
3.5	Métricas e definições operacionais	19
4	RESULTADOS	21
4.1	Análise dos Resultados	21
4.2	Processo proposto para Implementação de Testes Funcionais Automatizados	25
5	CONCLUSÃO	26
5.1	Conclusões e Contribuições	26
	REFERÊNCIAS	28

1 INTRODUÇÃO

A crescente demanda por soluções digitais entregues com rapidez e confiabilidade tem intensificado os desafios relacionados à garantia da qualidade no desenvolvimento de software. Em ambientes ágeis e iterativos, as organizações precisam equilibrar a velocidade das entregas com a necessidade de validações consistentes, tornando os testes de software uma atividade essencial ao longo do ciclo de desenvolvimento (Pressman; Maxim, 2021; Sommerville, 2011). Embora os testes manuais ofereçam flexibilidade e capacidade de adaptação a mudanças frequentes, sua execução recorrente tende a tornar-se onerosa e suscetível a falhas humanas, especialmente em cenários de regressão (Myers; Sandler; Badgett, 2012). Nesse contexto, a automação de testes vem sendo amplamente discutida na literatura como estratégia para otimizar a eficiência do processo de testes, reduzir o tempo de execução e assegurar maior consistência nas validações, sobretudo em aplicações web que apresentam fluxos críticos e repetitivos (Pressman; Maxim, 2021).

Apesar dos benefícios da automação de testes, diferentes autores apontam desafios em sua adoção. Feldmann (2019) destaca que a ausência de planejamento e de critérios para seleção de cenários frequentemente resulta em testes automatizados instáveis e com alto custo de manutenção. Crispin e Gregory (2010) também ressaltam que a automação implementada sem estratégia tende a gerar baixo retorno, especialmente quando não há definição clara do que deve ser automatizado. De forma complementar, o ISTQB (2023) indica que a avaliação da testabilidade e a priorização baseada em risco são fatores determinantes para o sucesso da automação. Nesse contexto, a ausência de processos estruturados, aliada à dificuldade na seleção de cenários adequados e à avaliação da testabilidade das funcionalidades, pode resultar em implementações pouco eficientes e com elevado custo de manutenção, comprometendo os ganhos esperados.

Nesse cenário, destaca-se o GLPI (Gestionnaire Libre de Parc Informatique), um sistema amplamente utilizado para a gestão de ativos e serviços de tecnologia da informação em organizações públicas e privadas. Falhas em funcionalidades relacionadas à abertura, acompanhamento e resolução de chamados podem impactar diretamente a operação dessas organizações, tornando essencial a validação desses fluxos por meio de testes funcionais confiáveis e recorrentes.

Dessa forma, este relato técnico tem como objetivo relatar e analisar a aplicação de testes funcionais manuais e automatizados em um contexto real, comparando o tempo de execução e o esforço total investido, bem como propor um processo estruturado que apoie a adoção gradual e sustentável da automação de testes funcionais em aplicações web, contribuindo para a redução de riscos e para a maturidade dos processos de qualidade de software.

Embora a automação de testes seja utilizada para aumentar a eficiência do processo de qualidade, sua adoção sem critérios claros pode resultar em soluções frágeis e com elevado custo de manutenção. Feldmann (2019) destaca que a automação implementada sem planejamento e sem seleção adequada de cenários tende a gerar testes instáveis e com alto custo de manutenção. Testes manuais, embora permitam flexibilidade e compreensão do comportamento do sistema,

tornam-se menos escaláveis em cenários de validação recorrente (Valente, 2022). Bach e Bolton (2025) afirmam que a automação deve ser compreendida como apoio ao trabalho intelectual do testador, e não como substituição da atividade de teste, exigindo planejamento, seleção criteriosa de cenários e avaliação da testabilidade.

1.1 Objetivos

1.1.1 Objetivo Geral

Estruturar um processo para implementação de testes funcionais automatizados em aplicações web.

1.1.2 Objetivos Específicos

- Definir cenários funcionais representativos para execução manual e automatizada;
- Medir o tempo de execução dos testes em ciclos controlados;
- Comparar o esforço humano total entre testes manuais e automatizados;
- Identificar o ponto de equilíbrio entre esforço inicial e ganho operacional;
- Propor diretrizes para adoção gradual da automação de testes funcionais em aplicações web.

1.2 Justificativa

A automação de testes permite maior eficiência, qualidade e segurança no desenvolvimento de software, reduzindo erros humanos e aumentando a confiabilidade das entregas. Além disso, promove cultura de qualidade, otimiza recursos e fortalece a competitividade das empresas no mercado digital. A proposta deste trabalho contribui diretamente para facilitar a adoção dessa prática em Software Houses, por meio de um guia prático e replicável.

1.3 Organização do Texto

Este trabalho está estruturado da seguinte forma:

- **Capítulo 1** — Introdução ao tema, justificativa, objetivos e organização do trabalho;
- **Capítulo 2** — Fundamentação teórica sobre qualidade de software, automação de testes e ferramentas;

- **Capítulo 3** — Metodologia adotada no desenvolvimento do estudo e definição do ambiente de testes;
- **Capítulo 4** — Desenvolvimento prático, implementação dos testes e análise dos resultados;
- **Capítulo 5** — Conclusão, limitações e recomendações para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA



Figura 1 – Pirâmide de Testes.

Fonte: Adaptado de SEGUINS (2024).

No contexto desta pesquisa, utiliza-se a Pirâmide de Testes como uma representação visual simplificada dos diferentes níveis de testes de software. Esse modelo conceitual orienta a estruturação das estratégias de teste, indicando que, quanto mais próximo do topo da pirâmide, maior tende a ser o custo e o tempo de execução dos testes.

Na base da pirâmide encontram-se os testes de unidade, responsáveis por validar partes isoladas do sistema, garantindo o correto funcionamento de funções ou métodos específicos de forma rápida e com baixo custo. No nível intermediário estão os testes de integração, que verificam a comunicação entre componentes, como integrações com banco de dados, APIs e serviços externos. No topo da pirâmide situam-se os testes de ponta a ponta (End-to-End – E2E), que simulam o fluxo completo de utilização da aplicação, incluindo testes de interface e regressão, com o objetivo de reproduzir o comportamento do usuário final.

A distribuição proposta pela pirâmide busca equilibrar cobertura e eficiência, priorizando testes mais rápidos e menos custosos, ao mesmo tempo em que mantém a validação dos fluxos críticos do sistema, conforme ilustrado na Figura 1. Nesse contexto, a compreensão dos diferentes níveis de teste torna-se essencial para a definição de estratégias eficazes de garantia da qualidade. Dessa forma, os próximos tópicos abordam os conceitos fundamentais de testes de software e garantia da qualidade, além de discutir a aplicação de testes manuais e automatizados, analisando seus impactos no esforço, custo e confiabilidade das entregas.

2.1 Testes de Software e Garantia da Qualidade

Os testes de software constituem uma atividade para a garantia da qualidade, pois permitem verificar se o sistema atende aos requisitos funcionais e não funcionais, além de reduzir riscos associados a falhas em produção. Em contextos contemporâneos de Engenharia de Software,

marcados por ciclos curtos de desenvolvimento e entregas frequentes, a atividade de teste deixa de ser uma etapa final e passa a integrar continuamente o ciclo de vida do software (Pressman; Maxim, 2021).

A garantia da qualidade, nesse cenário, está diretamente relacionada à capacidade de validar o comportamento do sistema de forma consistente, repetível e economicamente viável. A ausência de práticas estruturadas de teste tende a ampliar o retrabalho, reduzir a previsibilidade das entregas e comprometer a confiabilidade do software, especialmente em aplicações web caracterizadas por elevada complexidade funcional e evolução constante (Sommerville, 2011).

2.2 Testes Funcionais Manuais e suas Limitações

Os testes funcionais têm como objetivo validar o comportamento do sistema a partir da perspectiva do usuário, verificando se as funcionalidades implementadas estão em conformidade com os requisitos especificados. Tradicionalmente, esses testes são executados de forma manual, o que possibilita flexibilidade, interpretação contextual e adaptação a mudanças frequentes nos requisitos e nos fluxos da aplicação (Molinari, 2012).

Apesar dessas vantagens, a execução manual apresenta limitações significativas em cenários caracterizados por alta recorrência, como os testes de regressão. A repetição contínua dos mesmos fluxos eleva o esforço humano, aumenta a probabilidade de falhas operacionais e compromete a escalabilidade do processo de testes à medida que o sistema evolui (Myers; Sandler; Badgett, 2012). Conforme destacado por Valente (2022), testes manuais são inerentemente trabalhosos e precisam ser repetidos a cada modificação no sistema, o que os torna pouco adequados como única estratégia de validação em sistemas em evolução contínua, reforçando a necessidade de abordagens complementares para a sustentação da qualidade ao longo do tempo.

2.3 Automação de Testes Funcionais em Aplicações Web

A automação de testes funcionais surge como alternativa para ampliar a eficiência e a confiabilidade do processo de testes, sobretudo em aplicações web que concentram fluxos críticos e repetitivos. A execução automatizada possibilita a reexecução rápida e padronizada dos testes, reduzindo o tempo por ciclo e aumentando a previsibilidade dos resultados, o que se mostra particularmente relevante em cenários de regressão (Feldmann, 2019).

Entretanto, a literatura contemporânea ressalta que a automação não deve ser compreendida como substituta da atividade de teste. Bach e Bolton (2025) defendem que o teste de software é uma atividade intelectual, baseada em julgamento humano, interpretação e análise contextual, e que a introdução de ferramentas automatizadas transforma o processo de teste, em vez de eliminá-lo. Nessa perspectiva, a automação deve ser entendida como um mecanismo de apoio ao trabalho do testador, ampliando sua capacidade de validação e análise ao longo dos ciclos de desenvolvimento, sem descaracterizar o papel humano no processo de teste.

A adoção indiscriminada da automação, sem critérios claros de seleção de cenários e sem avaliação prévia da testabilidade, pode resultar em testes instáveis e elevado custo de manutenção. Crispin e Gregory (2010) destacam que a automação deve ser integrada de forma incremental ao processo de testes, especialmente em ambientes ágeis, a fim de garantir sustentabilidade e efetividade ao longo do tempo. Dessa forma, a automação deve ser tratada como uma decisão estratégica de processo, orientada por critérios técnicos e organizacionais, e não apenas como a adoção de uma ferramenta específica.

2.4 Testabilidade e Adoção Gradual da Automação

A testabilidade refere-se ao grau em que um sistema ou funcionalidade permite que os testes sejam definidos, executados e mantidos de forma eficaz ao longo do tempo. Aspectos como previsibilidade dos fluxos, estabilidade da interface, controle e preparação de dados de teste, além do isolamento de dependências externas, influenciam diretamente a viabilidade técnica e a sustentabilidade da automação de testes (Pressman; Maxim, 2021).

Em aplicações web em constante evolução, a ausência de critérios explícitos de testabilidade tende a resultar em testes automatizados frágeis e com elevado custo de manutenção. Conforme destacado por Valente (2022), testes automatizados funcionam como uma rede de proteção contra regressões, mas sua efetividade depende diretamente da qualidade do projeto do sistema e da adequação dos cenários selecionados para automação. Dessa forma, a avaliação prévia da testabilidade constitui etapa essencial para decisões mais assertivas sobre o que automatizar.

A literatura e guias consolidados da área indicam que a adoção gradual da automação, iniciando por cenários estáveis e de alta recorrência, contribui para a redução de riscos técnicos e para a consolidação progressiva do processo de qualidade (ISTQB, 2023). Bach e Bolton (2025) reforçam que a automação deve ampliar a capacidade analítica do testador, e não substituir o julgamento humano, o que demanda uma integração equilibrada entre testes manuais e automatizados.

Nesse contexto, a adoção incremental da automação, apoiada por critérios objetivos de testabilidade e seleção de cenários, favorece a sustentabilidade do processo de testes ao longo do tempo, reduzindo retrabalho, custos de manutenção e riscos associados à automação não planejada. Essa abordagem sustenta a proposta deste estudo ao integrar testes manuais e automatizados de forma estruturada e alinhada às necessidades práticas das organizações.

2.5 Ferramentas de Automação e Enfoque em Processo

Diversas ferramentas têm sido empregadas na automação de testes funcionais em aplicações web, entre as quais se destacam Selenium WebDriver, Playwright e Cypress. O Cypress tem ganhado destaque por sua arquitetura moderna, execução integrada ao navegador e facilidade de

configuração, características que o tornam adequado a contextos organizacionais que buscam estruturar ou amadurecer a automação de testes de forma incremental (Feldmann, 2019).

Todavia, a literatura converge ao apontar que a ferramenta, isoladamente, não garante o sucesso da automação. O fator determinante reside na existência de um processo estruturado, capaz de orientar a seleção de cenários, a avaliação da testabilidade e a integração equilibrada entre testes manuais e automatizados. Esse entendimento fundamenta o presente estudo, que propõe um processo aplicado e replicável para apoiar a adoção sustentável da automação de testes funcionais em aplicações web, alinhando eficiência operacional, julgamento humano e evolução contínua da qualidade de software.

3 METODOLOGIA

Este estudo caracteriza-se como uma pesquisa aplicada, com abordagem mista (qualitativa e quantitativa), operacionalizada por meio de um estudo de caso em uma aplicação web de gestão de serviços de tecnologia da informação. A adoção da abordagem mista em estudos de caso é recomendada quando se busca integrar a análise processual de um fenômeno com a mensuração objetiva de desempenho em contextos reais, sendo amplamente utilizada em pesquisas empíricas aplicadas à Engenharia de Software (Yin, 2015; Runeson; Höst, 2009).

No presente estudo, a dimensão qualitativa foi empregada para descrever e justificar o processo estruturado de automação de testes funcionais proposto, enquanto a dimensão quantitativa foi utilizada para mensurar e comparar o tempo de execução e o esforço humano entre testes manuais e automatizados. Essa combinação metodológica permite evitar comparações simplistas, ao distinguir o custo inicial de implantação do desempenho obtido em ciclos recorrentes de regressão, abordagem recomendada na literatura de Engenharia de Software para a avaliação de processos e ferramentas de teste (Pressman; Maxim, 2021).

O objeto de estudo foi o sistema GLPI, versão 10.0.17, amplamente utilizado para a gestão de serviços de tecnologia da informação em organizações públicas e privadas. O sistema foi executado em um ambiente containerizado, utilizando tecnologia de contêineres por meio do Docker Desktop, versão 4.44.3, o que permitiu a padronização e a reprodutibilidade do ambiente de execução, bem como o controle de variáveis relacionadas à instalação e à configuração do sistema, favorecendo a replicabilidade experimental, conforme recomendado em estudos empíricos aplicados à Engenharia de Software (Pressman; Maxim, 2021).

3.1 Ambiente computacional e ferramentas

O estudo foi conduzido no seguinte ambiente computacional: notebook Dell Inspiron 15 3520, processador Intel Core i5-1235U (12ª geração), 8 GB de memória RAM e SSD de 512 GB. O sistema operacional utilizado foi o Windows 11 Pro, com o navegador Google Chrome, versão 144.0.7559.110 (64 bits).

Para a gestão e execução dos testes funcionais manuais, foram utilizadas as ferramentas Jira Software e Xray Test Management, responsáveis pela documentação dos casos de teste, rastreabilidade, registro de evidências e mensuração do tempo de execução manual. A automação dos testes funcionais foi realizada com a ferramenta Cypress, versão 15.3.0, executada com Node.js v22.20.0, em modo headless, por meio de execução via terminal. A utilização de ferramentas de gerenciamento, execução e coleta de evidências, bem como a padronização do ambiente de testes, está alinhada às diretrizes do ISTQB, que destacam esses elementos como suporte à eficiência, ao monitoramento e ao controle do processo de testes (ISTQB, 2023).

3.2 Planejamento, seleção e priorização dos cenários de teste

O planejamento dos testes iniciou-se com a identificação das principais funcionalidades e fluxos críticos do sistema, priorizando ações recorrentes e com maior impacto potencial ao usuário final. Foram definidos quatro cenários funcionais, inicialmente executados de forma manual e, posteriormente, automatizados, para fins de comparação entre as abordagens.

Considerando o porte e a complexidade da aplicação analisada, que contempla um amplo conjunto de funcionalidades e fluxos operacionais, adotou-se uma seleção deliberada e orientada por critérios objetivos de um subconjunto de cenários funcionais, com o objetivo de viabilizar a realização da prova de conceito. Essa estratégia é adequada em estudos de caso aplicados cujo foco reside na avaliação de processos, técnicas ou ferramentas, e não na cobertura exaustiva de todas as funcionalidades do sistema (Yin, 2015; Runeson; Höst, 2009). A escolha de cenários representativos possibilita a análise empírica do processo proposto em condições realistas de uso, preservando o controle metodológico, a viabilidade do estudo e a validade interna dos resultados.

A definição e a priorização dos cenários consideraram critérios objetivos, a saber: recorrência de execução, criticidade funcional, relevância operacional e potencial de automação, avaliado a partir da estabilidade do fluxo e da interface. A priorização de testes com base em risco, impacto e frequência de uso é amplamente recomendada na literatura de testes de software como forma de direcionar o esforço para as funcionalidades mais críticas do sistema (Myers; Sandler; Badgett, 2012; Pressman; Maxim, 2021). Como regra de priorização, foram considerados candidatos à automação os cenários que apresentaram maior criticidade e recorrência. Esses critérios foram definidos previamente à execução dos testes e aplicados de forma uniforme a todos os cenários, em consonância com as diretrizes do ISTQB (2023).

3.3 Checklist de testabilidade

Com o objetivo de reduzir a subjetividade e registrar de forma auditável a decisão de automação, foi aplicado um *checklist* de testabilidade após a execução manual de cada cenário, caracterizando um instrumento de avaliação qualitativa estruturada. O *checklist* considerou aspectos como determinismo do fluxo, dependência de dados, dependências externas, estabilidade da interface, observabilidade, controlabilidade e presença de barreiras à automação.

Cada item foi avaliado em escala ordinal de três pontos (0 = inadequado; 1 = parcialmente adequado; 2 = adequado). Um cenário foi considerado elegível para automação quando atingiu pontuação mínima de 10, em um total máximo de 14 pontos, e não apresentou avaliação inadequada em itens considerados críticos. A utilização de instrumentos estruturados para avaliação da testabilidade está alinhada às diretrizes do ISTQB (2023), além de contribuir para a sustentabilidade da automação ao reduzir custos de manutenção (Pressman; Maxim, 2021).

A tabela 1 apresenta o *checklist* de testabilidade utilizado no estudo, estruturado a partir

dos critérios definidos para avaliação da viabilidade de automação dos cenários analisados.

Tabela 1 – *Checklist* de Testabilidade aplicado aos cenários.

Critério	Descrição	Pontuação (0–2)
Determinismo do fluxo	O cenário apresenta comportamento previsível e repetível?	
Dependência de dados	O cenário depende de dados externos ou variáveis dinâmicas?	
Dependências externas	Há dependência de APIs, serviços ou integrações externas?	
Estabilidade da interface	A interface apresenta estabilidade ao longo das execuções?	
Observabilidade	É possível validar claramente os resultados esperados?	
Controlabilidade	É possível controlar os estados e dados necessários para o teste?	
Barreiras à automação	Existem fatores que dificultam ou impedem a automação?	

Fonte: Elaborado pelos autores (2026).

O *checklist* foi aplicado a todos os cenários definidos no estudo; entretanto, para fins de apresentação, é demonstrado a seguir um exemplo representativo de sua aplicação.

Tabela 2 – Aplicação do *checklist* ao cenário “Abertura de chamado”.

Critério	Descrição	Pontuação (0–2)
Determinismo do fluxo	O cenário apresenta comportamento previsível e repetível?	2
Dependência de dados	O cenário depende de dados externos ou variáveis dinâmicas?	1
Dependências externas	Há dependência de APIs, serviços ou integrações externas?	2
Estabilidade da interface	A interface apresenta estabilidade ao longo das execuções?	2
Observabilidade	É possível validar claramente os resultados esperados?	2
Controlabilidade	É possível controlar os estados e dados necessários para o teste?	2
Barreiras à automação	Existem fatores que dificultam ou impedem a automação?	1
Pontuação total		12 / 14

Fonte: Elaborado pelos autores (2026).

Como exemplo de aplicação do *checklist* de testabilidade, o cenário “Abertura de chamado” foi avaliado após sua execução manual, obtendo pontuação total de 12 pontos. O cenário apresentou alto grau de determinismo, estabilidade de interface e boa capacidade de observação e controle dos resultados, sendo considerado adequado para automação. As limitações observadas referem-se à dependência parcial de dados e a pequenas barreiras relacionadas à variação de elementos na interface. Ainda assim, o cenário atendeu ao critério mínimo estabelecido, sendo classificado como elegível para automação.

3.4 Procedimento de execução

O procedimento metodológico foi conduzido de forma sequencial e controlada, em consonância com as atividades de teste descritas nas diretrizes do ISTQB, abrangendo planejamento, análise e modelagem, implementação e execução (ISTQB, 2023). As etapas envolveram a modelagem dos fluxos funcionais, o mapeamento e cadastro dos casos de teste na ferramenta Xray, a execução manual dos cenários para validação funcional e registro de tempos e evidências, a aplicação do checklist de testabilidade, a implementação incremental dos testes automatizados e a execução automatizada em modo headless.

Todas as etapas foram realizadas mantendo-se constantes as versões das ferramentas e o ambiente computacional ao longo do estudo, de modo a preservar a validade interna e a replicabilidade dos resultados, conforme recomendado em estudos de caso em Engenharia de Software (Runeson; Höst, 2009).

3.5 Métricas e definições operacionais

As métricas de execução e esforço foram utilizadas para monitorar e reportar o processo de testes, em consonância com as diretrizes do ISTQB quanto ao acompanhamento e relato do processo de testes por meio de medidas e evidências (ISTQB, 2023). A literatura de Engenharia de Software recomenda o uso de métricas de tempo e esforço como suporte à avaliação da eficiência e do custo de atividades de teste (Pressman; Maxim, 2021).

Foram adotadas métricas de tempo de execução por ciclo e de esforço humano total. O tempo de execução manual (T-man) foi mensurado em minutos por meio do cronômetro da ferramenta Xray, iniciando-se no momento da abertura da execução do caso de teste e encerrando-se com a conclusão do teste, incluindo o registro do status e das evidências.

O tempo de execução automatizada (T-aut) foi obtido a partir do relatório gerado pelo Cypress, em execução automatizada em modo headless (sem interface gráfica), considerando o intervalo entre o disparo da suíte de testes no terminal e o término da execução automatizada, com o resultado consolidado.

O esforço humano total investido (E-total) foi mensurado em horas-homem, a partir

da soma do tempo despendido nas atividades de configuração do ambiente, planejamento e modelagem, mapeamento dos casos de teste, execução manual, implementação dos scripts automatizados e execução automatizada. A utilização dessas métricas permite avaliar, de forma comparativa, a eficiência das abordagens manual e automatizada ao longo de ciclos de teste, especialmente em cenários de regressão, nos quais a automação tende a reduzir significativamente o esforço humano por execução (Myers; Sandler; Badgett, 2012).

4 RESULTADOS

Neste capítulo são apresentados os resultados alcançados durante todo o trabalho, bem como uma discussão e comparação com os resultados encontrados na literatura, destacando a importância desta pesquisa no contexto acadêmico.

4.1 Análise dos Resultados

A análise dos resultados concentrou-se em dois eixos principais: o tempo de execução dos testes por ciclo e o esforço humano total investido (E-total) em cada abordagem. Essa análise foi conduzida a partir da execução dos quatro cenários funcionais em três ciclos independentes, tanto na abordagem manual quanto na automatizada, mantendo-se as mesmas versões, ferramentas e condições de execução.

A Tabela 3 apresenta os tempos médios de execução por ciclo observados nos três ciclos de teste manual e automatizado, evidenciando as diferenças de desempenho entre as abordagens e a variabilidade observada em cada caso.

Tabela 3 – Tempo total de execução por ciclo (testes manuais e automatizados).

Abordagem	Ciclo 1	Ciclo 2	Ciclo 3	Média	Desvio padrão	Variância
Manual	132,87	130,37	129,95	131,06	1,58	2,49
Automatizada	4,55	4,58	4,55	4,56	0,02	0,0003

Fonte: Elaborado pelos autores (2026).

A partir da análise empírica dos resultados obtidos, torna-se possível estabelecer um diálogo direto com a literatura especializada. Os resultados evidenciam que a automação de testes introduz um esforço inicial adicional, relacionado ao planejamento, à implementação e à manutenção dos scripts. Conforme discutido por Bach e Bolton (2025), a introdução de ferramentas automatizadas transforma o processo de teste, incorporando novas atividades e responsabilidades ao trabalho humano. No entanto, quando aplicada de forma seletiva e incremental, a automação passa a atuar como suporte ao trabalho do testador, proporcionando ganhos expressivos de eficiência em ciclos recorrentes de validação. Esses achados também corroboram Valente (2022), ao indicar que testes automatizados funcionam como mecanismo de proteção contra regressões, reduzindo o esforço humano necessário para revalidação do sistema após modificações, especialmente em aplicações web com evolução contínua.

Os resultados indicam que, mantidas as mesmas versões, ferramentas e condições de execução, o tempo dos testes manuais tende a permanecer relativamente estável entre os ciclos, com variações associadas principalmente a fatores humanos. Entretanto, à medida que o sistema evolui e demanda ajustes nos fluxos funcionais, atualização do mapeamento dos testes ou ampliação do escopo de regressão, a abordagem manual passa a exigir maior esforço de reexecução e

revalidação, comprometendo sua escalabilidade. Em contraste, os testes automatizados apresentaram tempo médio significativamente inferior por ciclo, com baixa variação entre execuções, evidenciando maior previsibilidade e estabilidade do processo. Esses achados corroboram a literatura de Engenharia de Software, que aponta a automação como particularmente vantajosa em contextos de regressão e execução recorrente (Pressman & Maxim, 2021; Myers, Sandler, & Badgett, 2012).

Essa diferença torna-se especialmente relevante em cenários de testes de regressão end-to-end, nos quais os mesmos fluxos precisam ser validados repetidamente após alterações no sistema. Enquanto a execução manual permanece fortemente dependente da disponibilidade humana, a automação permite reexecuções frequentes com baixo custo marginal, além de fornecer feedback rápido sobre falhas e desvios de comportamento do sistema, contribuindo para a correção antecipada de defeitos e para a redução de retrabalho. O ISTQB (2023) destaca que esse feedback rápido é um dos principais benefícios da automação quando integrada a um processo de testes bem estruturado.

No que se refere ao esforço humano total investido (E-total), os resultados evidenciam que a automação requer um investimento inicial mais elevado, concentrado principalmente nas atividades de planejamento, mapeamento dos testes e implementação dos scripts automatizados. O detalhamento do esforço humano total investido por atividade encontra-se apresentado na Tabela 4, permitindo compreender a distribuição do esforço entre planejamento, execução manual e automação.

Tabela 4 – Esforço total investido por atividade (E_total).

Atividade	Descrição resumida	Tempo investido
Instalação do Docker Desktop e execução da imagem do GLPI	Preparação do ambiente containerizado	15 min
Configurações iniciais do GLPI	Ajustes básicos do sistema e criação de usuários	1 h
Exploração inicial do GLPI	Levantamento de funcionalidades e fluxos principais	30 min
Planejamento dos cenários de teste	Definição dos cenários e elaboração de mapas mentais	2 h
Configuração do Jira Software e Xray Test Management	Setup das ferramentas de gestão de testes	2 h
Mapeamento dos casos de teste no Xray	Cadastro dos testes, rastreabilidade e evidências	10 h
Execução dos testes manuais (3 ciclos)	Execução dos quatro cenários e registro de tempos e evidências	6 h 33 min
Implementação dos scripts automatizados (Cypress)	Criação e organização dos testes automatizados	12 h
Execução dos testes automatizados (3 ciclos)	Execução em modo headless e coleta dos tempos	15 min
Esforço humano total (E_total)		34 h 18 min

Fonte: Elaborado pelos autores (2026).

Embora o esforço inicial associado à automação seja superior ao da execução manual isolada, a análise conjunta do E-total e do ganho médio de tempo por ciclo evidencia que esse investimento tende a ser compensado após um número reduzido de ciclos de execução, caracterizando um ponto de equilíbrio (break-even) favorável à automação. Esse comportamento é consistente com o que é descrito na literatura, segundo a qual o custo inicial da automação tende a ser diluído ao longo das execuções recorrentes, especialmente em cenários de validação contínua e testes de regressão (Pressman; Maxim, 2021).

Outro aspecto relevante identificado nos resultados foi o papel do *checklist* de testabilidade na seleção dos cenários a serem automatizados. A aplicação desse instrumento contribuiu para evitar a automação de cenários instáveis ou com alto custo potencial de manutenção, direcionando o esforço para fluxos mais determinísticos e recorrentes. Essa prática mostrou-se fundamental para a sustentabilidade do processo de automação, reduzindo retrabalho e aumentando a confiabilidade dos testes automatizados ao longo dos ciclos, em alinhamento com as diretrizes do ISTQB (2023).

A Tabela 5 apresenta uma síntese comparativa entre as abordagens de testes funcionais manuais e automatizados, consolidando os principais achados empíricos discutidos nesta seção. Essa síntese evidencia como as diferenças observadas em termos de tempo, esforço e previsibilidade fundamentam a adoção de uma abordagem híbrida e planejada para a automação de

testes.

Tabela 5 – Síntese comparativa entre testes funcionais manuais e automatizados (com base nos resultados empíricos).

Aspecto analisado	Testes Manuais	Testes Automatizados
Esforço inicial	Menor esforço inicial, concentrado na execução e validação manual dos cenários	Maior esforço inicial, associado ao planejamento, preparação do ambiente e implementação dos scripts
Tempo de execução por ciclo	Tempo elevado por ciclo, mantendo-se relativamente estável enquanto o escopo não é alterado	Tempo significativamente reduzido por ciclo, com baixa variação entre execuções
Execução recorrente	Limitada, dependente da disponibilidade humana e suscetível a variações operacionais	Elevada, permitindo re-execuções frequentes com baixo custo marginal
Adequação a testes de regressão	Baixa, especialmente à medida que o escopo de regressão cresce	Alta, favorecendo execuções repetidas e feedback rápido
Sustentabilidade do processo	Reduzida em cenários de alta recorrência ou evolução frequente do sistema	Maior sustentabilidade em médio e longo prazo, com ganho progressivo ao longo dos ciclos
Previsibilidade dos resultados	Maior variabilidade entre execuções manuais	Alta previsibilidade, evidenciada pela estabilidade dos tempos automatizados

Fonte: Elaborado pelos autores (2026).

Os resultados obtidos neste estudo corroboram a literatura ao indicar que a automação de testes funcionais apresenta maior efetividade quando aplicada de forma planejada, seletiva e incremental, especialmente em contextos de testes de regressão. A análise empírica evidenciou que a integração entre testes manuais e automatizados permite equilibrar o esforço inicial de implantação, o esforço humano total e a eficiência operacional ao longo dos ciclos de teste, evitando tanto a automação prematura quanto a dependência excessiva de execuções manuais recorrentes.

Nesse contexto, os achados empíricos fundamentam o processo estruturado proposto, que consolida as práticas observadas neste estudo em um fluxo sequencial e decisório, orientado por critérios objetivos de seleção e avaliação da testabilidade. O processo atua como um guia prático baseado em evidências, apoiando equipes de desenvolvimento e qualidade de software na tomada de decisão sobre quando, o que e como automatizar testes funcionais em aplicações web.

4.2 Processo proposto para Implementação de Testes Funcionais Automatizados

Com base nos resultados obtidos, este estudo propõe um processo estruturado para a implementação de testes funcionais automatizados em aplicações web. O processo é composto por fases sequenciais e pontos de decisão, iniciando pela compreensão da aplicação e pelo planejamento dos testes, passando pela execução manual e pela avaliação da testabilidade, e culminando na automação e na manutenção contínua dos testes.

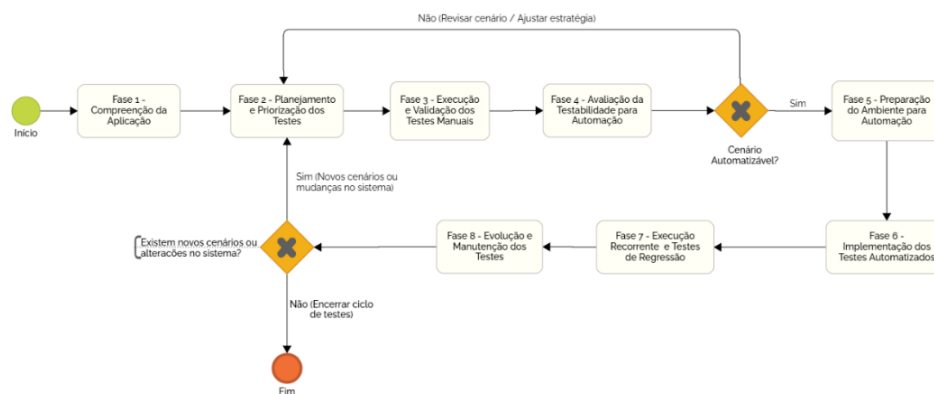


Figura 2 – Fluxograma do processo proposto para implementação de testes funcionais automatizados em aplicações web.

Fonte: Elaborado pelos autores (2025), com apoio da ferramenta HEFLO (versão gratuita).

O processo proposto não tem como objetivo substituir metodologias consolidadas de desenvolvimento ou de teste, mas atuar como um instrumento de apoio à decisão, especialmente em contextos organizacionais que buscam estruturar ou amadurecer a automação de testes de forma gradual e sustentável. Ao integrar testes manuais e automatizados de maneira equilibrada, o processo busca reduzir riscos, retrabalho e custos associados à automação não planejada ou tecnicamente inviável.

O nível de detalhamento adotado na modelagem do processo proposto visa torná-lo compreensível, replicável e adaptável a diferentes contextos organizacionais, mantendo seu caráter aplicado e tecnológico, sem a pretensão de se constituir como um manual operacional. Dessa forma, o processo consolida os achados empíricos deste estudo em um fluxo decisório estruturado, apoiando a adoção gradual e sustentável da automação de testes funcionais em aplicações web.

5 CONCLUSÃO

Nesta seção, são apresentados, de forma sucinta, os resultados obtidos e um fechamento de todo o trabalho desenvolvido.

5.1 Conclusões e Contribuições

Este artigo apresentou um relato técnico sobre a aplicação e a comparação de testes funcionais manuais e automatizados em uma aplicação web de gestão de serviços de tecnologia da informação. Por meio de um estudo de caso aplicado ao sistema GLPI, foi possível analisar empiricamente o tempo de execução por ciclo e o esforço humano total investido (E-total) em cada abordagem, considerando a execução dos cenários em três ciclos independentes, sob condições controladas.

Os resultados indicam que, embora a automação de testes exija maior esforço inicial especialmente nas atividades de planejamento, mapeamento e implementação dos scripts, ela proporciona ganhos expressivos de eficiência em execuções recorrentes, com redução significativa do tempo por ciclo e maior previsibilidade dos resultados. Esse comportamento mostra-se particularmente relevante em cenários de testes de regressão, nos quais a execução manual apresenta limitações de escalabilidade à medida que o sistema evolui. A análise conjunta do esforço inicial e do ganho médio por ciclo evidencia um ponto de equilíbrio favorável à automação, reforçando sua viabilidade em contextos de uso contínuo.

Como principal contribuição tecnológica, este estudo propôs um processo estruturado para a implementação gradual e sustentável da automação de testes funcionais, integrando testes manuais e automatizados de forma planejada e orientada por critérios objetivos. O processo enfatiza etapas como a execução manual inicial, a avaliação sistemática da testabilidade e a automação seletiva de cenários, reduzindo riscos associados à automação não planejada e favorecendo a sustentabilidade do processo ao longo do tempo. Dessa forma, o trabalho ultrapassa a simples comparação entre abordagens, oferecendo um instrumento prático e replicável de apoio à decisão para equipes de desenvolvimento e qualidade de software.

Os objetivos estabelecidos foram alcançados, uma vez que o estudo permitiu analisar empiricamente as abordagens de testes funcionais manuais e automatizados, comparar tempo de execução e esforço humano total e, a partir dos resultados obtidos, estruturar um processo aplicável à adoção gradual da automação de testes funcionais em aplicações web.

Como limitações do estudo, destaca-se a condução da pesquisa em um único contexto organizacional e a análise restrita a um conjunto específico de cenários funcionais, selecionados de forma deliberada para a realização da prova de conceito. Essas características limitam a generalização direta dos resultados para outros domínios ou tipos de sistemas. Além disso, o estudo concentrou-se em testes funcionais end-to-end, não abrangendo outros tipos de testes automatizados.

Como trabalhos futuros, sugere-se a aplicação do processo proposto em outros sistemas e contextos organizacionais, bem como a ampliação da análise para diferentes tipos de testes automatizados, ferramentas e níveis de complexidade. Adicionalmente, o processo apresentado estabelece uma base técnica favorável para evoluções futuras, como a integração da automação de testes a pipelines de CI/CD e a incorporação gradual de técnicas baseadas em inteligência artificial para apoio à manutenção dos testes, à identificação de falhas e ao aumento da eficiência do processo de qualidade de software.

Referências

ANICHE, Maurício. *Test-driven development: teste e design no mundo real*. São Paulo: Casa do Código, 2014.

BACH, James; BOLTON, Michael. *Taking testing seriously: the rapid software testing approach*. Hoboken: Wiley, 2025.

BRITO, Rafael; CARVALHO, Eduardo. *Automação de testes em aplicações web: práticas e ferramentas modernas*. São Paulo: Casa do Código, 2023.

CAPGEMINI. *World quality report 2023–2024*. 2023. Disponível em: <https://worldqualityreport.com>. Acesso em: 25 mar. 2026.

CRISPIN, Lisa; GREGORY, Janet. *Agile testing: a practical guide for testers and agile teams*. Boston: Addison-Wesley, 2010.

FELDMANN, Ralf. *Test automation: best practices and pitfalls*. Cham: Springer, 2019.

INTERNATIONAL SOFTWARE TESTING QUALIFICATIONS BOARD (ISTQB). *Certified tester foundation level (CTFL) v4.0*. 2023. Disponível em: <https://istqb.org/certifications/certified-tester-foundation-level-ctfl-v4-0/>. Acesso em: 25 mar. 2026.

MOLINARI, Leonardo. *Testes de software: produzindo sistemas melhores e mais confiáveis*. 2. ed. São Paulo: Érica, 2012.

MYERS, Glenford J.; SANDLER, Corey; BADGETT, Tom. *The art of software testing*. 3. ed. Hoboken: John Wiley & Sons, 2012.

PRESSMAN, Roger S.; MAXIM, Bruce R. *Software engineering: a practitioner's approach*. 9. ed. New York: McGraw-Hill Education, 2021.

RUNESON, Per; HÖST, Martin. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, New York, v. 14, n. 2, p. 131–164, 2009. Disponível em: <https://doi.org/10.1007/s10664-008-9102-8>. Acesso em: 25 mar. 2026.

SEGUINS, Neilton. Testes automatizados: como melhorar qualidade e confiabilidade do seu software. Alura, 2024. Disponível em: <https://www.alura.com.br/artigos/testes-automatizados>. Acesso em: 26 mar. 2026.

SOMMERVILLE, Ian. *Software engineering*. 9. ed. Boston: Pearson Education, 2011.

TRICENTIS. *State of test automation report*. 2023. Disponível em: <https://www.tricentis.com/resources/state-of-test-automation-report-2023>. Acesso em: 25 mar. 2026.

VALENTE, Marco Tulio. *Engenharia de software moderna*. Belo Horizonte: Autor, 2022.

YIN, Robert K. *Case study research and applications: design and methods*. 6. ed. Thousand Oaks: Sage, 2015.