

Análise Comparativa entre Cypress e postman para Automação de Testes de API

Comparative Analysis between Cypress and postman for API Test Automation

Gabriel belloni Benevenuto dos Santos ¹

¹Universidade de Passo Fundo

*172951@upf.br; gabriel-belloni@hotmail.com; ...

Abstract

Software quality assurance represents a significant challenge due to the complexity of modern products. Automated testing emerges as a superior alternative to manual methods, offering benefits such as increased speed, cost reduction, and minimized risks associated with development failures. Comparative analysis of automation tools, such as Cypress and Postman, reveals not only their advantages and disadvantages but also specific functionalities that adapt to different testing scenarios. This study provides a better understanding of the tools, going beyond the information provided by documentation. Additionally, case studies in controlled environments confirm the effectiveness of these tools and identify areas for future improvements. This work primarily focuses on comparing Cypress and Postman for API testing, aiming to understand user preferences and analyze performance, the learning curve, and the suitability of the tools for different needs in API testing.

Keywords: API Testing 1; Automated Testing 2; Comparative Analysis 3; Cypress 4; Learning Curve 5; Postman 5; User Preferences 6;

Resumo

A garantia da qualidade de software representa um desafio significativo devido à complexidade dos produtos modernos. Os testes automatizados surgem como uma alternativa superior aos métodos manuais, oferecendo benefícios como maior velocidade, redução de custos e minimização de riscos associados a falhas no desenvolvimento. A análise comparativa entre ferramentas de automação, como Cypress e Postman, revela não apenas suas vantagens e desvantagens, mas também funcionalidades específicas que se adaptam a diferentes cenários de teste. Esse estudo permite uma melhor compreensão das ferramentas, indo além das informações da documentação. Complementarmente, estudos de caso em ambientes controlados confirmam a eficácia dessas ferramentas e identificam áreas para melhorias futuras. Este trabalho tem como foco principal a comparação entre Cypress e Postman para testes de API, buscando entender a preferência dos usuários e analisar a performance, a curva de aprendizado e a adequação das ferramentas para diferentes necessidades em testes de API.

Palavras-Chave: Testes de API 1; Testes Automatizados 2; Análise Comparativa 3; Cypress 4; Curva de Aprendizado 5; Postman 6; Preferências do Usuário 7;

1 Introdução

A garantia da qualidade de um software é um grande desafio devido à grande complexidade dos produtos atualmente. Idealmente, o processo de teste busca manter a consistência do produto com suas especificações e objetivos originais [Myers and et al. \(2004\)](#). A inclusão do teste automatizado na esteira de produção de software traz grandes vantagens em comparação aos testes manuais, tais como velocidade, redução de gastos e prevenção de danos decorrentes de bugs e problemas no desenvolvimento.

Nos estudos comparativos sobre as ferramentas de automação de teste, podemos realizar uma verificação das vantagens, desvantagens, funcionalidades e recursos únicos que podem ser úteis em diferentes cenários de teste, indo além das informações obtidas apenas da documentação das mesmas, e maximizando o potencial da ferramenta para o desenvolvimento de um produto de excelência [Rollwagen and et al. \(2020\)](#).

Utilizando-se de ferramentas de pesquisa baseadas em perguntas, podemos encontrar na comunidade de usuários de automação, tanto no âmbito empresarial quanto no estudantil, quais são as ferramentas mais aceitas e os motivos para isso.

Com a realização de um estudo de caso com cenários de teste, garantiremos a validação dos resultados obtidos em ambientes controlados, proporcionando evidências tangíveis sobre a eficácia do processo de teste. Com base nos resultados do estudo de caso, é possível identificar áreas de melhoria e desafios decorrentes de sua utilização [da Silva Ferreira and et al. \(n.d.\)](#)

1.1 Objetivo geral

O objetivo geral deste trabalho é realizar uma comparação entre as ferramentas Cypress e Postman, visando sua capacidade na área de teste de API e a preferência dos usuários da comunidade de testes. Este estudo tem como propósito fornecer percepções sobre os pontos fortes e fracos das ferramentas, auxiliando na seleção da melhor opção para diferentes necessidades no contexto de testes de API.

1.2 Objetivos Específicos

- Estudo sobre a documentação das duas ferramentas: Analisar a abrangência, clareza e acessibilidade da documentação fornecida, que são aspectos fundamentais para a adoção e eficácia no uso das ferramentas.
- Estudo sobre a curva de aprendizado das ferramentas: Avaliar o tempo e o esforço necessários para que novos usuários se tornem proficientes no uso de Cypress e Postman, influenciando diretamente a eficiência de seu emprego em projetos de software.
- Estudo do Desempenho das ferramentas: Examinar a performance de cada ferramenta em termos de velocidade, capacidade de lidar com grandes volumes de dados e precisão nos testes de API.
- Realizar uma pesquisa sobre preferência dos usuários em diferentes situações utilizando o Google Forms: Obter feedback direto da comunidade de testadores sobre suas experiências e preferências entre as ferramentas,

em variados contextos de uso.

- Desenvolver Casos de Testes para as duas ferramentas para análise: Criar e aplicar cenários de teste específicos para avaliar de maneira prática as capacidades de Cypress e Postman, identificando pontos de destaque e limitações.

Neste contexto, os testes automatizados, em comparação com os métodos manuais, oferecem vantagens significativas que contribuem para a eficiência, economia e segurança dos produtos de software. O estudo proposto buscará explorar essas dimensões, focando na análise comparativa de Cypress e Postman para testes de API, fundamentais no ciclo de desenvolvimento de software contemporâneo.

2 Trabalhos relacionados

Estudo Comparativo entre Ferramentas de Automação de Testes: Selenium e Cypress.

Este estudo compara as ferramentas Selenium e Cypress, destacando sua importância na automação de testes para garantia da qualidade do software. O artigo visa ajudar desenvolvedores e engenheiros de teste a selecionar a ferramenta mais adequada, considerando diferentes critérios como tipo de teste, funcionalidades e aplicabilidade [Sousa \(2023\)](#).

Um Estudo Comparativo entre Ferramentas de Automação de Teste: Selenium, Robot e Cypress. Neste estudo, as ferramentas Selenium, Robot e Cypress são comparadas com base em critérios específicos, como facilidade de aprendizado, qualidade da documentação e eficiência na execução dos testes. Os resultados destacam a importância da seleção adequada da ferramenta de automação de teste para diferentes cenários de projeto [da Silva Ferreira, de Souza, Tonieto and Branco \(n.d.\)](#).

Um Estudo Comparativo de Características das Ferramentas de Automação de Teste End-to-End: Cypress vs QA Wolf vs TestCafé.

Neste estudo, três ferramentas de automação de teste end-to-end são analisadas: Cypress, QA Wolf e TestCafé. O objetivo é identificar diferenças significativas em suas capacidades e usabilidade. Desafios relacionados à validade interna e externa do estudo são discutidos para melhor compreensão dos resultados [de CT Figueiredo et al. \(2023\)](#).

Implementação de uma Ferramenta de Análise de Testes Automatizados para Aumento de Produtividade em um Ambiente de Desenvolvimento Este estudo foca na integração de testes automatizados com visualização de dados para análise da qualidade do software. Utiliza-se o framework Cypress para automação de testes end-to-end, integrado com o Metabase para visualização detalhada dos resultados [Campos and da Silva \(2023\)](#).

3 Metodologias

Para garantir uma análise abrangente e precisa entre as ferramentas Cypress e Postman no contexto da automação de testes de API, o trabalho seguirá uma abordagem estruturada que envolve diversas fases, baseada na metodologia de [Gil \(2002\)](#).

3.1 Pesquisa sobre as ferramentas

O primeiro passo da metodologia envolve a análise da documentação de ambas as ferramentas, Cypress e Postman, e a avaliação de sua curva de aprendizado. Esse processo é essencial para determinar a facilidade com que desenvolvedores de diferentes níveis de experiência conseguem adotar essas ferramentas.

- **Documentação:** A documentação oficial de cada ferramenta será examinada com base em clareza, organização e profundidade das informações, incluindo exemplos práticos e a cobertura de funcionalidades avançadas.
- **Curva de Aprendizado:** Será avaliado o tempo necessário para que um novo usuário se torne produtivo, considerando o suporte disponível, como tutoriais e fóruns, e a dificuldade de tarefas básicas e avançadas.

Os critérios de avaliação incluem:

- Facilidade de acesso às informações.
- Clareza e qualidade dos exemplos fornecidos.
- Tempo necessário para aprender e executar funções básicas e avançadas.
- Disponibilidade de recursos de suporte (guias, fóruns, vídeos).

3.2 Estudo Aplicado

Nesta fase, serão desenvolvidos casos de teste específicos para ambas as ferramentas, simulando cenários reais de uso de APIs. Em seguida, será feita uma análise de desempenho com base na execução desses testes.

- **Criação de Casos de Teste:** Serão criados cenários que abrangem desde testes simples até testes mais complexos envolvendo grandes volumes de dados e múltiplos endpoints.
- **Execução e Análise de Desempenho:** A execução dos testes será monitorada para comparar a velocidade de execução, consumo de recursos, estabilidade e precisão na detecção de erros.

Os critérios de avaliação incluem:

- Velocidade de execução dos testes.
- Consumo de recursos (CPU e memória).
- Precisão na detecção de erros e clareza dos logs gerados.
- Estabilidade e capacidade de lidar com grandes volumes de dados.

3.3 Pesquisa com Usuários

Além das análises técnicas, será realizada uma pesquisa com usuários para entender suas percepções e preferências ao usar Cypress e Postman. Esse subcapítulo visa coletar informações qualitativas e quantitativas sobre a experiência de profissionais e estudantes na área de automação de testes.

- **Desenvolvimento de Questionário:** Um questionário será criado para captar informações sobre a experiência

dos usuários com as duas ferramentas. As perguntas cobrirão tópicos como facilidade de uso, curva de aprendizado, qualidade da documentação, e satisfação geral.

- **Coleta de Respostas:** O questionário será distribuído a profissionais da área de teste de software e automação, bem como a estudantes de cursos relacionados. As respostas serão coletadas de forma anônima para garantir a imparcialidade dos dados.
- **Análise das Respostas:** As respostas serão analisadas quantitativa e qualitativamente, buscando identificar padrões de uso e preferência, assim como os principais pontos positivos e negativos apontados por cada grupo de usuários.

Os critérios de avaliação incluem:

- Satisfação dos usuários com a documentação e os recursos de suporte disponíveis.
- Percepção sobre a facilidade de uso e eficiência das ferramentas.
- Preferência entre Cypress e Postman para diferentes tipos de testes.
- Identificação dos principais desafios enfrentados pelos usuários ao utilizar cada ferramenta.

3.4 Análise Comparativa Final

Após a coleta dos dados e observações, será realizada uma análise comparativa final, consolidando os resultados obtidos nas fases anteriores. Esta análise buscará identificar as principais vantagens e desvantagens de cada ferramenta, com base nos seguintes critérios:

- **Documentação e Curva de Aprendizado:** Qual ferramenta apresenta documentação mais acessível e uma curva de aprendizado mais suave.
- **Desempenho:** Comparação da performance de ambas as ferramentas em cenários variados.
- **Preferências dos Usuários:** Através da pesquisa com usuários, será avaliada a popularidade de cada ferramenta e suas principais qualidades apontadas pelos usuários.
- **Adequação ao Contexto:** Em quais cenários uma ferramenta se destaca em relação à outra, considerando tipos de teste, complexidade do projeto e necessidades específicas.

4 Análise da documentação

A documentação de uma ferramenta de automação de testes desempenha um papel crucial na adoção e uso eficiente por parte dos desenvolvedores. Ela influencia diretamente a curva de aprendizado, o suporte na resolução de problemas e a implementação de boas práticas. Nesta seção, será analisada a documentação das ferramentas Cypress e Postman com base em critérios de clareza, organização, abrangência e exemplos práticos.

4.1 Cypress

A documentação do Cypress é essencial para garantir que desenvolvedores consigam utilizar a ferramenta de forma

eficaz. Nesta subseção, discutimos a documentação com base em vários critérios, como clareza, organização, abrangência e exemplos práticos.

4.1.1 Clareza e Organização

A documentação oficial do Cypress é dividida em seções claras, facilitando a navegação entre os tópicos. Cada funcionalidade ou conceito é descrito de maneira direta, o que proporciona uma boa experiência, especialmente para aqueles que estão iniciando no uso da ferramenta. A estrutura bem definida permite ao usuário localizar rapidamente as informações necessárias.

4.1.2 Abrangência

A documentação do Cypress cobre uma ampla gama de funcionalidades, desde o processo de instalação até o uso de recursos avançados, como testes de API e manipulação de dados assíncronos. O conteúdo vai além de uma simples referência técnica, oferecendo explicações detalhadas sobre o uso correto das funcionalidades. Isto inclui cenários comuns e boas práticas que aumentam a eficiência na automação de testes.

4.1.3 Exemplos Práticos

Um dos principais pontos fortes da documentação do Cypress é a presença de exemplos práticos. A cada funcionalidade apresentada, são incluídos exemplos que mostram como aplicá-las em situações reais. Isso facilita o aprendizado, permitindo que desenvolvedores repliquem cenários de teste com base nos exemplos disponibilizados e os adaptem às suas necessidades.

4.1.4 Recursos Adicionais

Além da documentação oficial, o Cypress conta com uma ampla gama de recursos complementares, como tutoriais em vídeo, fóruns de discussão e uma comunidade ativa. Esses recursos contribuem para a troca de experiências entre os usuários e ajudam na resolução de problemas que não estão diretamente cobertos pela documentação oficial.

4.1.5 Curva de Aprendizado

Embora a documentação seja bastante clara, novos usuários podem encontrar dificuldades iniciais devido ao uso intensivo de JavaScript. A familiaridade com este ecossistema pode influenciar a rapidez com que a ferramenta é adotada de maneira eficaz. No entanto, a documentação fornece suporte adequado para desenvolvedores de diferentes níveis de experiência, minimizando essa curva de aprendizado.

4.1.6 Pontos Negativos da Documentação do Cypress

A documentação do Cypress é amplamente elogiada por sua clareza e abrangência em testes de front-end, porém apresenta algumas limitações. Entre os pontos negativos, destacam-se o foco limitado em cenários avançados e a cobertura insuficiente de testes de API. Além disso, a documentação pressupõe conhecimento prévio em JavaScript e Node.js, o que pode aumentar a curva de aprendizado para iniciantes. Exemplos práticos para integrações com ferramentas de CI/CD também são superficiais. Outro ponto relevante é que, devido às frequentes atualizações do Cy-

press, a documentação nem sempre acompanha essas mudanças de forma imediata, o que pode causar dificuldades na adoção de novas funcionalidades.

4.2 Postman

A documentação do Postman é uma das suas principais características que contribui para a sua popularidade entre os desenvolvedores. Abaixo estão os principais aspectos positivos e negativos da documentação do Postman.

4.2.1 Clareza e Organização

A documentação do Postman é intuitiva, com seções bem definidas que permitem fácil navegação. A clareza na apresentação dos tópicos favorece a compreensão, especialmente para novos usuários.

4.2.2 Abrangência

Cobrindo desde requisições simples até automação complexa, a documentação oferece explicações detalhadas e boas práticas, facilitando a utilização eficaz da ferramenta em diversos cenários.

4.2.3 Exemplos Práticos

Os exemplos práticos são um dos destaques, permitindo que os usuários vejam a aplicação das funcionalidades em situações reais. Isso simplifica o aprendizado e a adaptação das soluções às necessidades específicas de cada projeto.

4.2.4 Recursos Adicionais

Além da documentação, o Postman oferece tutoriais em vídeo e uma comunidade ativa que compartilha dicas e soluções. Esses recursos complementares enriquecem a experiência do usuário e ajudam na resolução de problemas.

4.2.5 Curva de Aprendizado

Embora a documentação seja acessível, iniciantes podem encontrar dificuldades com conceitos de APIs. No entanto, os recursos disponibilizados ajudam a reduzir essa curva de aprendizado, tornando a ferramenta mais amigável para desenvolvedores de diferentes níveis.

4.2.6 Pontos Negativos

Apesar de suas qualidades, a documentação do Postman também apresenta algumas limitações:

- **Superficialidade em Funcionalidades Avançadas:** O suporte para funcionalidades mais complexas, como a automação de testes com scripts, pode ser considerado superficial, exigindo que os usuários busquem informações adicionais em fontes externas.
- **Inconsistência em Atualizações:** Algumas partes da documentação não são atualizadas com a mesma frequência que as novas versões do software, o que pode causar confusão.
- **Falta de Exemplos Reais:** Em algumas seções, a falta de exemplos de uso em cenários reais pode dificultar a aplicação prática do que é ensinado.

Ao analisar a documentação do Cypress e do Postman,

observa-se que ambas as ferramentas apresentam características notáveis em termos de clareza, organização e suporte aos desenvolvedores. No entanto, cada uma tem suas particularidades que as tornam mais adequadas a diferentes contextos de uso.

A documentação do Cypress possui uma estrutura clara e bem definida, com seções organizadas que facilitam a navegação, especialmente para aqueles familiarizados com JavaScript. Essa abordagem detalhada abrange desde conceitos básicos até funcionalidades avançadas, fornecendo exemplos práticos que ilustram o uso da ferramenta em cenários reais. No entanto, limitações são perceptíveis em áreas específicas, como a cobertura de testes de API e integrações com ferramentas de CI/CD. Além disso, a dependência do ecossistema JavaScript pode ser uma barreira para iniciantes, embora a documentação ofereça suporte suficiente para minimizar essa dificuldade.

Por outro lado, a documentação do Postman destaca-se pela acessibilidade e por ser intuitiva para um público mais amplo, indo além de desenvolvedores experientes. A cobertura de funcionalidades básicas e intermediárias é excelente, com explicações detalhadas e boas práticas que auxiliam na utilização da ferramenta. Apesar disso, a documentação apresenta lacunas no suporte a funcionalidades avançadas, como a automação de testes complexos e a aplicação de scripts. Exemplos práticos são abundantes, mas em algumas áreas, a falta de casos reais detalhados pode dificultar a aplicação direta em projetos específicos.

5 Estudo Aplicado

Este estudo aplicado utilizará o **ServeRest**, uma API REST gratuita que simula o ambiente de uma loja virtual. O **ServeRest** foi desenvolvido especificamente como material didático, com o objetivo de auxiliar no aprendizado e na prática de testes de API. A API simula funcionalidades típicas de uma aplicação de e-commerce, como o cadastro e login de usuários, a adição e manipulação de produtos, a criação e busca de carrinhos de compra, entre outras operações essenciais para o funcionamento de uma loja virtual. Dessa forma, permite a reprodução de cenários realistas de teste, que são essenciais para avaliar a eficácia e o desempenho das ferramentas de automação de testes de API.

Para fins comparativos, este estudo executará uma automação de testes de um conjunto de vinte e oito cenários de testes idênticos tanto com o Cypress quanto com o Postman. Esses cenários de teste foram desenhados para cobrir diversas funcionalidades oferecidas pela API **ServeRest**, incluindo operações de CRUD (Create, Read, Update, Delete) em usuários e produtos, autenticação de usuários, operações em carrinhos de compra e fluxos completos de compra e cancelamento de pedidos. Cada cenário foi criado para explorar aspectos específicos da API, desde a validação de entradas válidas e inválidas até a verificação de respostas para ações autorizadas e não autorizadas.

A execução desses vinte e oito cenários em ambas as ferramentas permitirá uma análise detalhada de como o Cypress e o Postman se comportam em termos de desempenho (tempo de execução) e eficácia (habilidade em detectar erros ou comportamentos inesperados). Com isso,

será possível identificar pontos fortes e limitações de cada ferramenta em diferentes situações, além de avaliar qual delas oferece uma abordagem mais adequada para a automação de testes de API. Assim, este estudo contribuirá para a escolha mais informada de uma ferramenta de automação de testes, especialmente em ambientes que demandem alta confiabilidade e eficiência no desenvolvimento e validação de APIs.

5.1 Cenários Criados

Esses vinte e oito cenários de teste englobam diversas operações essenciais para a validação completa da API. Entre eles estão:

- **Cadastro de Usuários:**

1. Cadastro de usuário com dados válidos: Valida se é possível registrar um usuário com dados corretos.
2. Cadastro de usuário com dados aleatórios: Verifica a criação de usuários com dados gerados automaticamente.
3. Cadastro de usuário sem senha: Testa se o sistema impede o cadastro de usuário sem senha.
4. Cadastro de usuário com campo de senha vazio: Avalia se o sistema bloqueia o cadastro com o campo de senha em branco.

- **Login de Usuários:**

5. Login com dados válidos: Valida o processo de autenticação com dados corretos.
6. Login com senha contendo caracteres especiais: Verifica a aceitação de senhas com caracteres especiais.
7. Login sem e-mail: Testa se o sistema impede login sem o campo de e-mail preenchido.
8. Login sem senha: Valida se o sistema bloqueia o login sem senha.
9. Login com parâmetros vazios: Checa se o sistema lida corretamente com todos os campos vazios.

- **Buscar, Modificar e Deletar Usuário por ID:**

10. Busca de usuário por ID: Valida a busca de um usuário específico na base de dados pelo identificador único.
11. Modificação de usuário por ID: Verifica a atualização de informações de um usuário específico usando o ID.
12. Deletar usuário por ID: Testa se o sistema permite remover um usuário específico pelo seu identificador.

- **Listagem e Cadastro de Produtos:**

13. Listagem de produtos: Valida se o sistema exibe corretamente a lista de produtos cadastrados.
14. Cadastro de produto com dados válidos: Verifica o registro de um produto com informações completas e válidas.
15. Cadastro de produto com parâmetros vazios: Testa se o sistema impede o registro de um produto com campos vazios.
16. Cadastro de produto sem nome, mas com aspas duplas: Avalia o tratamento de campos de nome em branco com aspas.
17. Cadastro de produto sem preço, mas com aspas

duplas: Verifica o tratamento do campo de preço em branco com aspas.

18. Cadastro de produto sem descrição, mas com aspas duplas: Checa o tratamento do campo de descrição vazio com aspas.
19. Cadastro de produto sem quantidade, mas com aspas duplas: Testa o sistema com o campo de quantidade vazio e aspas.

• Busca, Modificação e Deletar Produto por ID:

20. Busca de produto por ID: Valida a busca de produtos específicos pelo identificador único.
21. Modificação de produto por ID: Verifica a atualização de detalhes de um produto específico pelo ID.
22. Deletar produto por ID: Testa a remoção de um produto específico usando seu identificador único.

• Carrinho de Compras:

23. Listagem de carrinhos: Valida se o sistema lista corretamente os carrinhos de compras ativos.
24. Cadastro de carrinho: Verifica o processo de criação de um novo carrinho de compras.
25. Busca de carrinho por ID: Testa a busca de um carrinho específico pelo seu identificador único.
26. Conclusão de compra: Avalia o fluxo de finalização de uma compra associada a um carrinho.
27. Cancelamento de compra: Valida a funcionalidade de cancelamento de um pedido associado a um carrinho.

• Listagem de Usuários:

28. Listagem de todos os usuários: Valida se o sistema exibe corretamente todos os usuários registrados.

Esses cenários permitem uma análise abrangente das funcionalidades críticas, assegurando que tanto o Cypress quanto o Postman sejam testados em condições equivalentes.

5.2 Tempo de Execução

A seguir, a Tabela 1 apresenta o tempo de dez execuções dos vinte e oito cenários de teste automatizados, que abrange todos os cenários listados e discutidos anteriormente. Esses cenários foram cuidadosamente selecionados para abranger as funcionalidades críticas e avaliar a performance das ferramentas Cypress e Postman em condições comparáveis.

Tabela 1: Comparação dos Tempos de Execução dos Testes entre Cypress e Postman

Execução	Tempo (s) - Cypress	Tempo (s) - Postman
1	12,5	10,2
2	13,1	11,0
3	12,8	10,7
4	14,3	11,5
5	13,7	10,9
6	15,0	11,8
7	12,2	10,3
8	14,8	12,0
9	13,0	10,5
10	14,5	11,2

Ao observar os valores de tempo registrados, é possível perceber que ambas as ferramentas apresentam tempos de execução semelhantes para um conjunto de vinte e oito cenários de teste. Esse número relativamente limitado de cenários facilita uma análise direta, pois permite que os tempos sejam comparados sem uma carga excessiva de dados. A proximidade nos tempos reforça a capacidade de ambas as ferramentas em lidar com um volume moderado de cenários com eficiência.

Contudo, ao extrapolarmos essa análise para ambientes de teste mais robustos, onde o número de cenários de teste pode ser significativamente maior, os impactos de performance tornam-se mais perceptíveis. Nesse contexto, a execução do Postman mostrou-se aproximadamente 18,98% mais rápida do que a do Cypress. Essa diferença percentual indica uma vantagem considerável em termos de eficiência, especialmente para cenários de grande escala onde tempos de execução menores podem significar uma economia substancial de recursos e tempo no ciclo de testes.

5.3 Consumo de recursos

A configuração da computador utilizada para esse teste está descrita na tabela 2. É importante notar que a utilização de recursos pode variar consideravelmente dependendo da configuração da máquina utilizada. Diversos fatores influenciam o desempenho das ferramentas de automação de testes, como a capacidade do processador, a quantidade de memória RAM e o tipo de placa de vídeo instalada.

Tabela 2: Especificações do Computador

Componentes	Detalhes
Sistema Operacional	Windows 10
Processador	Intel Core i5 (7ª Geração)
Placa de Vídeo	NVIDIA GeForce GTX 1050

A análise do consumo de CPU e memória RAM entre Cypress e Postman revela um comportamento diferente em termos de demanda de recursos computacionais durante as execuções dos vinte e oito cenários de testes automatizados de API. Conforme os dados apresentados nas tabelas 3 e 4, foram realizadas dez execuções para cada ferramenta, com o monitoramento do uso de CPU em % e memória RAM em MB.

Tabela 3: Uso de CPU e Memória RAM do Cypress em Dez Execuções

Execução	Uso máximo de CPU	Uso máximo de RAM
1	34.1%	725.3 MB
2	27.2%	712.3 MB
3	36.5%	735.2 MB
4	29.7%	715.4 MB
5	39.7%	754.8 MB
6	35.0%	730.9 MB
7	38.0%	742.8 MB
8	28.5%	720.1 MB
9	31.0%	718.8 MB
10	32.8%	722.2 MB

Tabela 4: Uso de CPU e Memória RAM do Postman em Dez Execuções

Execução	Uso máximo de CPU	Uso máximo de RAM
1	50.8%	755.0 MB
2	42.0%	738.4 MB
3	36.0%	723.7 MB
4	49.3%	752.1 MB
5	52.4%	762.7 MB
6	40.3%	735.9 MB
7	52.9%	768.1 MB
8	44.7%	742.6 MB
9	46.5%	747.3 MB
10	38.2%	730.5 MB

O consumo de CPU do Cypress varia de um mínimo de 27,2% a um máximo de 39,7%. Em comparação, o Postman apresenta um uso de CPU que varia de 36% a um máximo de 52,9% ao longo das dez execuções. Esse padrão indica que o Postman requer mais processamento para executar os mesmos cenários de teste, o que pode impactar o desempenho em máquinas com capacidade de CPU limitada, especialmente em contextos de testes extensivos ou de alta carga.

Em relação ao uso de memória RAM, o Cypress consome entre 712,3 MB e 754,8 MB durante as execuções dos testes. Esse consumo relativamente contido sugere que o Cypress é eficiente na gestão de recursos, permitindo uma execução mais leve em termos de memória. Por sua vez, o Postman apresenta um consumo que varia de 723,7 MB a 768,1 MB ao final das dez execuções.

Embora a diferença no uso de RAM entre as duas ferramentas seja relativamente pequena, com o Postman utilizando, em média, mais memória, essa variação pode ser significativa em ambientes de teste com recursos limitados. O Postman demonstra um consumo maior e mais consistente ao longo das execuções, o que pode levar a um aumento da carga no sistema e potencialmente afetar o desempenho de outras aplicações que estejam sendo executadas simultaneamente.

Os resultados indicam que o Cypress pode ser mais eficiente em termos de uso de recursos, especialmente de CPU, durante a execução de testes. Para equipes com infraestrutura limitada, essa diferença pode representar uma vantagem significativa em termos de economia de recursos e estabilidade do sistema durante a automação de testes de API. Em contrapartida, o Postman, ao consumir mais CPU e RAM, pode exigir uma maior demanda por recursos em ambientes de execução contínua ou com grandes volumes de dados.

5.4 Precisão na Detecção de Erros e Clareza dos Logs

A precisão na detecção de erros e a clareza dos logs são fatores essenciais em qualquer ferramenta de automação de testes. Esses elementos garantem que os desenvolvedores identifiquem rapidamente falhas nos cenários testados e compreendam as causas subjacentes dos problemas. Logs bem estruturados e informativos agilizam a correção de erros, proporcionando maior eficiência no ciclo de desenvolvimento.

Além disso, uma detecção precisa de falhas evita falsos positivos e negativos, aumentando a confiança nos resultados dos testes. A seguir será discutido como Cypress e Postman se comportam em termos de precisão e clareza na geração de logs, evidenciando as diferenças e similaridades no suporte à análise e depuração dos testes realizados.

A partir do estudo aplicado, é possível perceber as diferenças entre as ferramentas na precisão e clareza dos logs gerados durante a execução dos testes. As Figuras 1, 2, 3 e 4 ilustram essas diferenças.

Figura 1: Log de erro gerado pelo Cypress

Figura 2: Segunda parte do Log de erro gerado pelo Cypress

A partir das figuras 1, 2, é possível observar que o log gerado pelo Cypress é significativamente mais descritivo em

e os elementos que foram verificados, indicando visualmente que cada etapa foi concluída conforme esperado.

Esse tipo de log é vantajoso porque ajuda os desenvolvedores a validar rapidamente se os testes passaram sem problemas, sem sobrecarregar o log com informações excessivas. No caso de testes de sucesso, o Cypress adota uma abordagem minimalista, mostrando apenas o essencial para confirmar que o teste foi bem-sucedido. Essa simplicidade é útil para o monitoramento de rotinas de teste, pois permite que o usuário identifique rapidamente o status geral dos testes sem precisar examinar linhas de dados desnecessárias.

Além disso, o Cypress inclui um indicador visual em verde, para os testes bem-sucedidos, o que torna ainda mais fácil a identificação de resultados positivos na interface gráfica. Esse feedback visual contribui para uma experiência mais intuitiva, permitindo que a equipe de QA e desenvolvedores compreendam rapidamente o estado dos testes automatizados e se concentrem nas possíveis falhas que requerem atenção.

Por outro lado a partir das figuras 6 e 7, o Postman, sendo uma ferramenta especializada em testes de API, oferece um log semelhante ao do Cypress, mas com características adicionais que o tornam mais intuitivo e fácil de interpretar.

Entre essas características, destacam-se as abas de *Body*, *Header* e *Test Result*, que segmentam claramente as informações relevantes de cada requisição. Essa estrutura modular facilita a navegação e compreensão dos resultados pelos desenvolvedores, tornando o Postman uma escolha eficiente para a análise de testes de APIs. Assim, enquanto o Cypress é eficaz para validar fluxos gerais de teste, o Postman oferece maior praticidade na visualização de logs de sucesso específicos para APIs. além de possuir o console 7 que traz ainda mais informações.

5.5 Pesquisa com Usuário

A pesquisa com usuários tem como objetivo obter insights sobre a experiência prática de profissionais que utilizam as ferramentas *Cypress* e *Postman* para automação de testes de API. Esta etapa busca validar os aspectos teóricos apresentados na comparação técnica, avaliando fatores como facilidade de uso, curva de aprendizado, eficiência na execução de testes e satisfação dos usuários.

5.5.1 Descrição do Questionário

O questionário foi dividido em seções que abordam diferentes aspectos da experiência com as ferramentas. As perguntas são apresentadas a seguir:

5.5.2 Experiência com Postman

- Qual a sua experiência com Postman?
- Para quais finalidades você utiliza Postman?
- O quão satisfeito você está com a documentação do Postman?
- O quão satisfeito você está com o suporte para automação de testes do Postman?
- O quão fácil é realizar testes de API com Postman?
- O quão satisfeito você está com o feedback do Postman sobre os testes?

- Quais são os pontos fortes do Postman, na sua opinião?

5.5.3 Experiência com Cypress

- Qual a sua experiência com Cypress?
- Para quais finalidades você utiliza Cypress?
- O quão satisfeito você está com a documentação do Cypress?
- O quão satisfeito você está com o suporte para automação de testes do Cypress?
- O quão fácil é realizar testes de API com Cypress?
- O quão satisfeito você está com o feedback do Cypress sobre os testes?
- Quais são os pontos fortes do Cypress, na sua opinião?

5.5.4 Comparação entre Postman e Cypress

- Qual ferramenta você considera mais fácil de usar para testes de API?
- Qual ferramenta tem melhor suporte para automação de testes?
- Qual ferramenta você acha que oferece uma melhor documentação e comunidade de suporte?
- Na sua experiência, qual ferramenta oferece melhores relatórios e feedback sobre os testes?

5.5.5 Respostas

A partir de dezoito respostas ao questionário, tanto de usuários do meio acadêmico quanto do corporativo, chegamos às seguintes informações:

Dez pessoas possuem conhecimento em ambas as ferramentas, sete apenas no Postman e uma pessoa apenas no Cypress, como exemplificado na imagem abaixo 8.

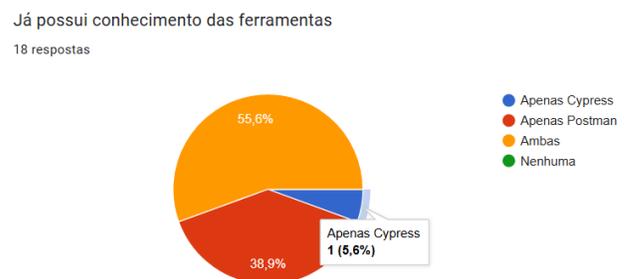


Figura 8: Gráfico de conhecimento prévio das ferramentas

Das pessoas que possuem experiência em ambas as ferramentas, cinco têm conhecimento avançado em Postman, duas possuem conhecimento intermediário e três têm conhecimento iniciante. Já no Cypress, quatro possuem conhecimento avançado, quatro têm conhecimento intermediário e duas possuem conhecimento iniciante exemplificado no gráfico abaixo 9.



Figura 9: Gráfico de experiência com ambas as ferramentas

Das pessoas que possuem apenas experiência em Postman, duas têm conhecimento avançado, quatro têm conhecimento intermediário e uma tem conhecimento iniciante exemplificado no gráfico abaixo 10.

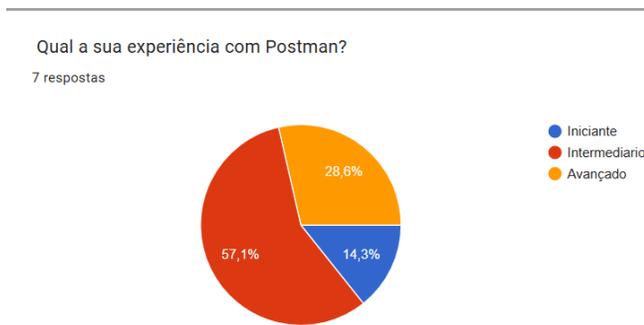


Figura 10: Gráfico de experiência com Postman

Apenas uma pessoa respondeu que possuía conhecimento avançado exclusivamente em Cypress.

5.5.6 Usuários com experiências em ambas as ferramentas

As dez pessoas que responderam possuir conhecimento em ambas as ferramentas apresentaram o uso do Cypress e do Postman para as seguintes finalidades 11:

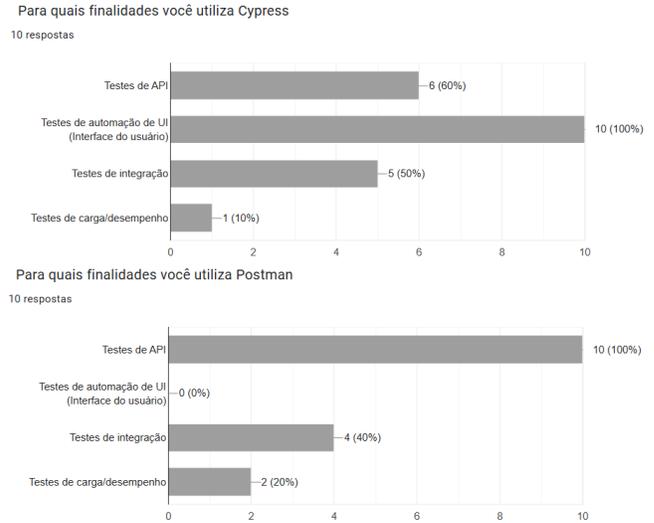


Figura 11: Distribuição do uso do Cypress e Postman.

Em uma escala de 1 a 5, onde 1 representa "pouco satisfeito" e 5 representa "muito satisfeito", a satisfação dos usuários com a documentação do Postman e do Cypress apresentou os seguintes resultados 12:

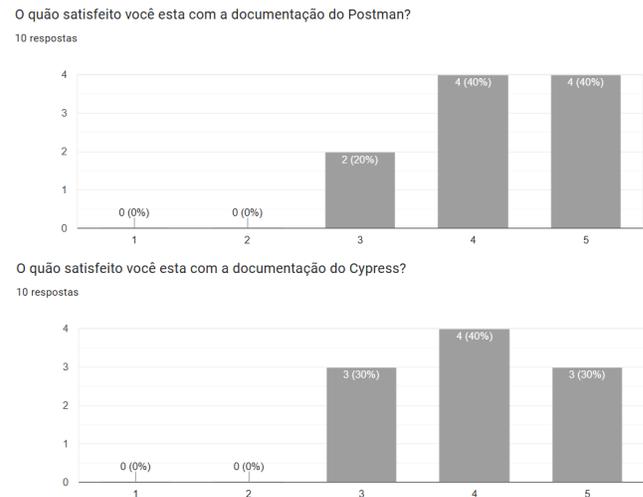
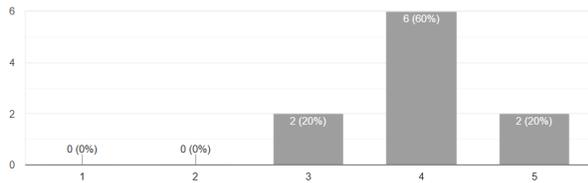


Figura 12: Satisfação com a documentação do Postman e Cypress.

A satisfação dos usuários com o suporte para automação de testes do Postman e do Cypress, em uma escala de 1 a 5, com os seguintes resultados 13:

O quão satisfeito você está com o suporte para automação de testes do Postman?

10 respostas



O quão satisfeito você está com o suporte para automação de testes do Cypress?

10 respostas

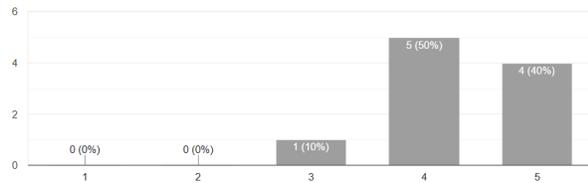
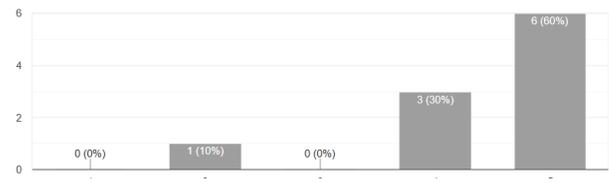


Figura 13: Satisfação com o suporte para automação de testes do Postman e Cypress.

O quão fácil é realizar testes de API com Postman?

10 respostas



O quão fácil é realizar testes de API com o Cypress?

10 respostas

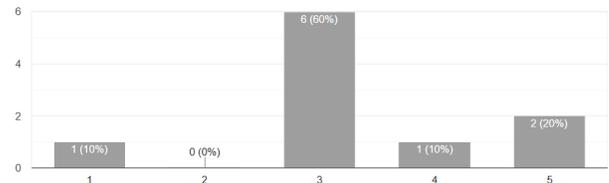


Figura 15: Facilidade de realizar testes de API do Postman e Cypress.

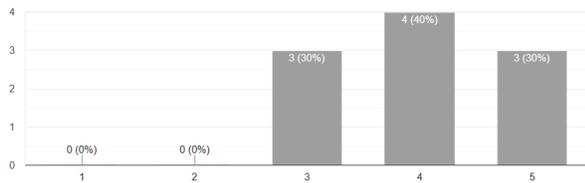
5.5.7 Usuários com experiências apenas em Postman

Para as sete pessoas que possuem conhecimento apenas em Postman apresentaram seu uso para as seguintes finalidades 16:

A satisfação dos usuários com o feedback sobre os testes do Postman e do Cypress, em uma escala de 1 a 5, com os seguintes resultados 14:

O quão satisfeito você está com o feedback do Postman sobre os testes?

10 respostas



O quão satisfeito você está com o feedback do Cypress sobre os testes?

10 respostas

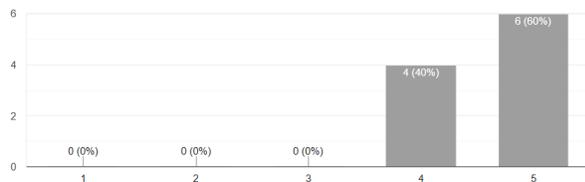


Figura 14: Satisfação com o feedback sobre os testes do Postman e Cypress.

Para quais finalidades você usa Postman?

7 respostas

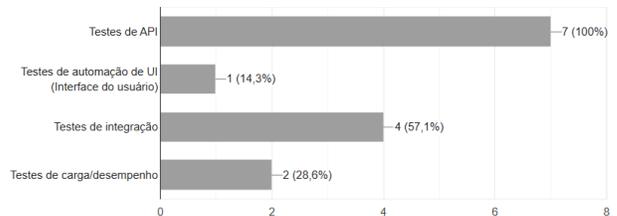


Figura 16: Distribuição do uso do Postman.

A satisfação dos usuários com a documentação do Postman com os seguintes resultados 17:

O quão satisfeito você está com a documentação do Postman?

7 respostas

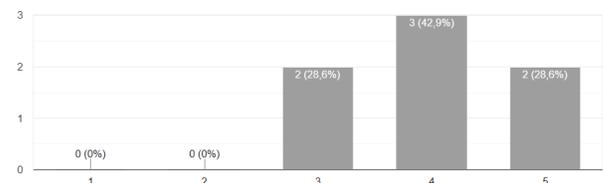


Figura 17: Satisfação com a documentação do Postman.

Os usuários avaliaram a facilidade de realizar testes de API com o Cypress e o Postman, em uma escala de 1 a 5, com os seguintes resultados 15:

Em uma escala de 1 a 5, onde 1 representa "pouco satisfeito" e 5 representa "muito satisfeito", a satisfação dos usuários com o suporte para automação de testes apresen-

tou os seguintes resultados 18:

O quão satisfeito você esta com o suporte para automação de testes do Postman?

7 respostas

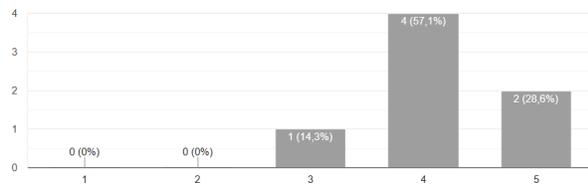


Figura 18: Satisfação com o suporte para automação de testes do Postman.

A satisfação dos usuários com o feedback sobre os testes, em uma escala de 1 a 5, com os seguintes resultados 19:

O quão satisfeito você esta com o feedback do Postman sobre os testes?

7 respostas

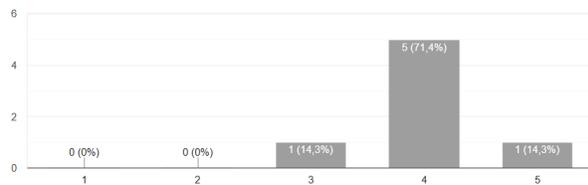


Figura 19: Satisfação com o feedback sobre os testes do Postman.

Os usuários avaliaram a facilidade de realizar testes de API em uma escala de 1 a 5, com os seguintes resultados 20:

O quão fácil é realizar testes de API com Postman?

7 respostas

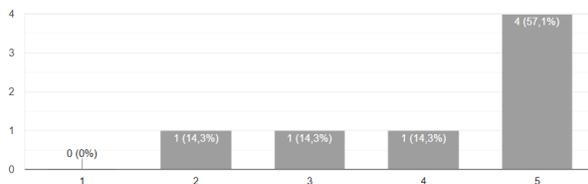


Figura 20: Facilidade de realizar testes de API no Postman.

5.5.8 Usuários com experiências apenas em Cypress

Para a única pessoa que possui conhecimento apenas no Cypress, apresentou seu uso para as seguintes finalidades 21:

Para quais finalidade você usa o Cypress?

1 resposta

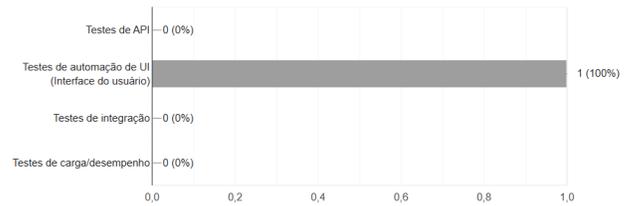


Figura 21: Distribuição do uso do Cypress.

Em uma escala de 1 a 5, onde 1 representa "pouco satisfeito" e 5 representa "muito satisfeito", a satisfação dos usuários com a documentação do Cypress apresentou os seguintes resultados 22:

O quão satisfeito você esta com a documentação do Cypress?

1 resposta

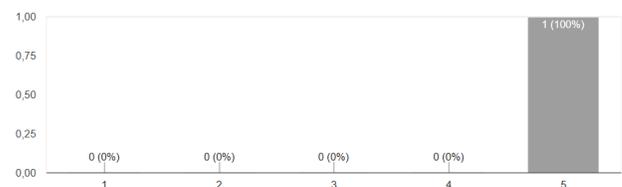


Figura 22: Satisfação com a documentação do Cypress.

A satisfação do usuário com o suporte para automação de testes, em uma escala de 1 a 5, com os seguintes resultados 23:

O quão satisfeito você esta com o suporte para automação de testes do Cypress?

1 resposta

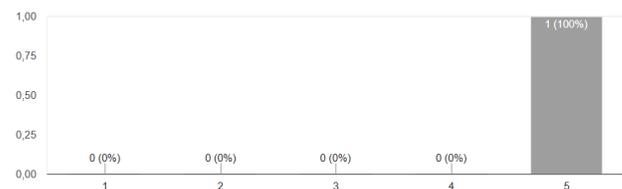


Figura 23: Satisfação com o suporte para automação de testes do Cypress.

A satisfação do usuário com o feedback sobre os testes, em uma escala de 1 a 5, com os seguintes resultados 24:

O quão satisfeito você está com o feedback do Cypress sobre os testes?

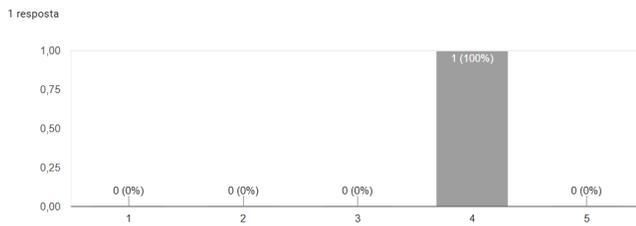


Figura 24: Satisfação com o feedback sobre os testes do Cypress.

O usuário avaliou a facilidade de realizar testes de API em uma escala de 1 a 5, com os seguintes resultados 25:

O quão fácil é realizar testes de API com o Cypress?

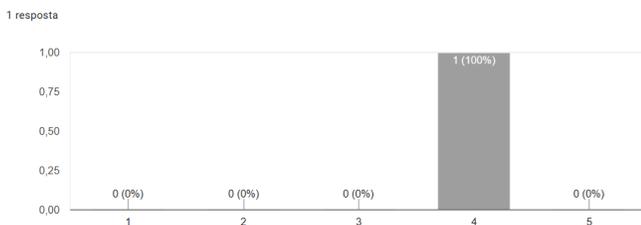


Figura 25: Facilidade de realizar testes de API no Cypress.

A partir das seguintes perguntas Quais são os pontos fortes do Cypress na sua opinião? e Quais são os pontos fortes do Postman na sua opinião? encontramos os seguintes pontos principais:

5.5.9 Postman

Os principais pontos fortes citados do Postman são sua facilidade de uso, gratuidade e facilidade de organização. Cada um desses aspectos foi constantemente destacado nas respostas do público do questionário, refletindo a experiência positiva dos usuários com a ferramenta.

5.5.10 Cypress

Os principais pontos fortes do Cypress citados pelos usuários são sua velocidade em debugar, a versatilidade das ferramentas, que inclui a possibilidade de adicionar suporte a mocks, relatórios detalhados e a integração com CI/CD. Esses pontos foram destacados de forma consistente nas respostas do público.

6 Considerações Finais

- **Documentação e Curva de Aprendizado:** As duas ferramentas possuem uma documentação bem elaborada; no entanto, o Cypress se destaca por oferecer mais exemplos práticos, enquanto o Postman, apesar de fornecer menos exemplos de cenários práticos, ganha em clareza e simplicidade, proporcionando uma grande acessibilidade para os usuários. Em relação à curva de aprendizado, o Postman se destaca por sua facilidade de

aprendizado e utilização, ao contrário do Cypress, que apresenta uma curva de aprendizado mais acentuada devido à necessidade de conhecimento em JavaScript, essencial para sua utilização.

- **Desempenho:** Na questão de desempenho, o Postman se destaca pela sua velocidade de execução de testes, sendo até 18,98% mais rápido que o Cypress. No entanto, o Cypress se sobressai no uso de recursos, consumindo menos CPU e memória RAM do que o Postman para a mesma quantidade de testes. Isso pode representar uma grande vantagem quando se trata de uma maior massa de testes.
- **Preferências dos Usuários:** A partir da pesquisa de usuários realizada, foi constatada uma leve preferência pelo Postman para testes de API, devido à sua facilidade de utilização, gratuidade e rápida iniciação.
- **Adequação ao Contexto:** Em análise dos estudos realizados, constatou-se que o Postman se adequa mais a contextos simples, que não necessitam de grande flexibilidade. Já o Cypress, por sua capacidade de se moldar a diversos cenários, desde os mais simples até os mais complexos, torna-se uma ferramenta adequada para projetos complexos e com grande volume de validações.

7 Conclusão

Este estudo apresentou uma análise comparativa entre as ferramentas com base na análise comparativa entre Cypress e Postman para testes de API alguns pontos importantes podem ser destacados, considerando a documentação, desempenho, uso de recursos, precisão na detecção de erros, clareza dos logs e a preferência dos usuários coletada na pesquisa.

A documentação de ambas as ferramentas são bem elaboradas e com abordagens distintas. O Postman se destaca por sua documentação com foco na clareza, acessibilidade e facilidade de utilização, embora a exploração de funcionalidades avançadas seja limitada devido à presença de poucos exemplos. Por outro lado, o Cypress oferece uma documentação muito abrangente, com recursos avançados bem detalhados. No entanto, a necessidade de conhecimento em JavaScript traz uma maior dificuldade de aprendizado para novos usuários.

Em questão de desempenho e uso de recursos nos testes aplicados, o Postman demonstrou ser 18,98% mais rápido na execução de cenários, demonstrou ser ideal para ambientes que demandam velocidade de execução. Por outro lado, o Cypress, com menor consumo de CPU e memória RAM, é mais adequado para ambientes com infraestrutura limitada ou testes complexos com grande volume de dados.

Em questão dos logs gerados pelas ferramentas o Cypress possui logs detalhados e descritivos, é recomendado para ambientes mais complexos e com grande volume de dados. Já o Postman, com logs simplificados e um console robusto para mais informações, é mais adequado para análises simples e diretas de APIs.

Para projetos mais simples e que demandam agilidade na configuração e execução de testes, o Postman se apresenta como uma escolha mais adequada, graças à sua acessibilidade, clareza e facilidade de uso. Já o Cypress se des-

taca em projetos mais complexos, que exigem maior controle sobre os testes, eficiência no uso de recursos e maior detalhamento na depuração de erros. A escolha entre as ferramentas dependerá, portanto, do contexto do projeto e das necessidades específicas para cada projeto e da experiência da equipe de desenvolvimento e testes.

Referências

- Campos, M. M. and da Silva, E. d. O. (2023). Implementação de uma ferramenta de análise de testes automatizados para aumento de produtividade em um ambiente de desenvolvimento, *Caderno de Estudos em Engenharia de Software* 5(1).
- da Silva Ferreira, A. C., de Souza, D. A., Tonieto, M. T. and Branco, R. V. C. (n.d.). Selenium, robot e cypress: Um estudo comparativo entre ferramentas de automação de teste.
- da Silva Ferreira, A. C. and et al. (n.d.). Selenium, robot e cypress: Um estudo comparativo entre ferramentas de automação de teste.
- de CT Figueiredo, R. M., Fernandes, R. E. S., de CT Figueiredo, R. M., Araujo, C. M. and de Menezes, J. S. (2023). Um estudo comparativo de características das ferramentas de automação de teste end-to-end: Cypress vs qa wolf vs testcafé, *Anais do X Encontro Nacional de Computação dos Institutos Federais*, SBC, pp. 53–60.
- Gil, A. C. (2002). *Como elaborar projetos de pesquisa*, Editora Atlas SA.
- Myers, G. J. and et al. (2004). *The Art of Software Testing*, John Wiley & Sons, Chichester.
- Rollwagen, A. F. and et al. (2020). Comparativo entre ferramentas de automação de testes de software para sistemas web, *Salão do Conhecimento* 6(6).
- Sousa, R. E. d. (2023). Um estudo comparativo entre ferramentas de automação de testes: selenium e cypress.