Comparação de Testes Automatizados entre Cypress e Puppeteer em um E-commerce

Gabriel Gonçalves Montanher¹\*; Adirson Maciel de Freitas Junior2

1UNIP – Bauru. Bacharelado em Ciência da Computação. Bauru, São Paulo, Brasil

2Escola Superior de Agricultura Luiz de Queiroz (ESALQ) – Doutor em Economia Aplicada – Av. Pádua Dias, 11 – Agronomia – 13418-900 – Piracicaba (SP), Brasil

\*autor correspondente: gabrielmontanherr@gmail.com

Comparação de Testes Automatizados entre Cypress e Puppeteer em um E-commerce

**Resumo**

A automação de testes desempenha um papel fundamental na garantia da qualidade de aplicações web, especialmente no setor de e-commerce, onde a experiência do usuário e a confiabilidade do sistema são essenciais. O objetivo comparar duas ferramentas amplamente utilizadas na automação de testes: Cypress e Puppeteer. A análise será realizada a partir da implementação de testes automatizados para os processos de login e registro de usuários em uma plataforma de e-commerce.

Os resultados esperados incluem a identificação das principais vantagens e limitações de cada ferramenta, fornecendo *insights* relevantes para profissionais e empresas que buscam otimizar seus processos de automação de testes.

**Palavras-chave:** Comparação; Testes; Automação; Puppeteer; Cypress;

**Introdução**

O crescimento do comércio eletrônico tem impulsionado a necessidade de soluções cada vez mais robustas para garantir a qualidade e a usabilidade das aplicações web. Com o aumento da complexidade dos sistemas e a demanda por atualizações frequentes, os testes automatizados se tornaram essenciais para assegurar o funcionamento adequado das funcionalidades, como login e cadastro de usuários. Nesse contexto, diversas ferramentas surgiram para facilitar a automação de testes, destacando-se os frameworks Cypress e Puppeteer.

O Cypress é amplamente reconhecido por sua abordagem inovadora e sua sintaxe intuitiva, tornando-se uma escolha popular para a automação de testes de aplicações web modernas. Segundo a Equipe Totvs (2021), o Cypress permite simular interações de usuários de maneira simplificada, como preenchimento de formulários e validação de elementos da interface. Além disso, Coutinho (2023) destaca que sua sintaxe acessível e a proximidade com a linguagem JavaScript tornam a ferramenta altamente produtiva para desenvolvedores e profissionais de garantia da qualidade.

Por outro lado, o Puppeteer é uma biblioteca do Node.js que oferece uma API de alto nível para controlar navegadores por meio do protocolo DevTools. De acordo com Oliveira (2024), essa ferramenta permite não apenas a automação de testes, mas também a extração de dados e a geração de capturas de tela, sendo amplamente utilizada para testes end-to-end e tarefas de scraping.

Diante desse cenário, este trabalho busca realizar uma análise comparativa entre Cypress e Puppeteer no contexto da automação de testes de login e cadastro em aplicações de e-commerce. O objetivo é avaliar qual dessas ferramentas apresenta melhor desempenho, considerando fatores como facilidade de implementação, tempo de execução e eficácia na detecção de falhas.

**Metodologia ou Material e Métodos**

Nesta pesquisa, realizou-se uma análise comparativa entre Cypress e Puppeteer, duas das ferramentas mais utilizadas na automação de testes para aplicações web. Para a avaliação, desenvolveu-se uma tela de login em HTML para um e-commerce, servindo como base para testar a eficácia e a eficiência de ambas as ferramentas.

A tela de login implementada incluiu elementos essenciais, como campos para inserção de nome de usuário e senha, um botão de login e uma opção de recuperação de senha. Esses componentes foram selecionados por representarem funcionalidades críticas em plataformas de e-commerce.

Os testes foram conduzidos utilizando dados de login e cadastro de usuários, armazenados em arquivos JSON, sem a utilização de um banco de dados relacional.

A comparação entre Cypress e Puppeteer baseou-se em critérios como facilidade de uso, velocidade de execução dos testes e cobertura de funcionalidades. Considerando a crescente complexidade das aplicações web, a escolha da ferramenta de automação de testes adequada é essencial para garantir a qualidade do software. Este estudo busca fornecer insights sobre a ferramenta mais adequada para diferentes cenários de automação de testes em aplicações de front-end.

**Resultados e Discussão**

Neste tópico, é apresentada uma análise comparativa entre o Cypress e o Puppeteer no contexto de testes de login. O objetivo é avaliar o desempenho de ambas as ferramentas em diferentes cenários, como tentativas de login com credenciais inválidas, ausência de preenchimento de campos, logout, e também a interação com elementos da página, como navegação e clique no botão de produto para compra. Essa comparação busca identificar qual ferramenta oferece melhor desempenho em termos de tempo de resposta para cada cenário, considerando tanto a eficácia na validação de campos quanto a agilidade nas interações de navegação e compra, que são comuns em aplicações de *e-commerce.*

No processo de autenticação do sistema, a tela de login apresenta dois campos principais: um para o e-mail e outro para a senha. Esses campos são essenciais para garantir que o usuário consiga acessar a plataforma de maneira segura e eficiente. A tela é simples e direta, com destaque para os campos de inserção de dados, permitindo uma navegação fácil e rápida para o usuário. A seguir, a Tabela 1 apresenta a tela de login, destacando os testes de interface mencionados.

**Tabela 1. Primeira comparação de testes entre Cypress e Puppeteer (Tela de Login).**

| **Cypress** | **Resultado** | **Puppeteer** | **Resultado** |
| --- | --- | --- | --- |
| Deve logar com e-mail ou senha inválido. | 690ms | Deve logar com e-mail ou senha inválido. | 299ms |
| Deve logar sem preencher os campos e-mail e senha. | 62ms | Deve logar sem preencher os campos e-mail e senha. | 458ms |
| Deve logar sem preencher o campo e-mail. | 143ms | Deve logar sem preencher o campo e-mail. | 295ms |
| Deve logar sem preencher o campo senha. | 125ms | Deve logar sem preencher o campo senha. | 655ms |
| Deve fazer o login com sucesso. | 4.1s | Deve fazer o login com sucesso. | 604ms |
| Deve deslogar e retornar a página de login. | 1s | Deve deslogar e retornar a página de login. | 655ms |

Fonte: o autor, 2024.

Os testes realizados, tanto com Cypress quanto com Puppeteer, visam validar o comportamento esperado do sistema de autenticação, cobrindo diferentes cenários que podem ocorrer durante o processo de login e logout de um usuário. A tabela a seguir apresenta os resultados desses testes, que incluem verificações de login com credenciais inválidas, campos obrigatórios não preenchidos, login bem-sucedido e deslogamento.

Os testes realizados no Cypress (690ms) e no Puppeteer (299ms) verificaram a capacidade do sistema de lidar com tentativas de login utilizando credenciais incorretas. O tempo de execução foi relativamente curto, o que indica que o sistema responde de forma eficiente a tentativas falhas. No entanto, a diferença no tempo de execução entre as duas ferramentas sugere que o Cypress, por ser um framework end-to-end voltado para testes de interação do usuário com a interface, realiza um processo mais complexo de interação e validação. Embora o desempenho esteja dentro do esperado, ele pode ser otimizado caso o número de tentativas inválidas aumente consideravelmente.

O teste que validou o comportamento do sistema quando nenhum campo é preenchido (62ms no Cypress e 458ms no Puppeteer) revelou uma variação considerável nos tempos de execução entre as ferramentas. Esse resultado mostra que o sistema valida adequadamente a ausência de dados de entrada, mas também evidencia diferenças de desempenho nas ferramentas utilizadas. Enquanto o Cypress apresenta um tempo de execução mais rápido, o Puppeteer mostra uma leve demora. Essa variação se deve à arquitetura das ferramentas, sendo o Cypress projetado para testar interações diretas com a página no navegador, enquanto o Puppeteer envolve etapas adicionais, como manipulação do DOM e captura de eventos assíncronos.

Nos testes em que apenas um dos campos (e-mail ou senha) não foi preenchido, ambos os frameworks apresentaram tempos de execução variáveis (Cypress: 143ms e 295ms, Puppeteer: 125ms e 655ms). Essa diferença reflete a forma como cada ferramenta gerencia a validação de entrada e o fluxo de erro. A discrepância nos tempos pode ser explicada pela arquitetura dos testes, pelo nível de complexidade da interação com o *DOM* e pelo tempo necessário para validar as mensagens de erro exibidas ao usuário. O desempenho do Cypress, com sua abordagem mais robusta para testes de interação, resulta em um *overhead* maior quando valida erros, como a falta de preenchimento de campos.

O teste realizado para validar o login com credenciais válidas apresentou uma diferença significativa nos tempos de execução: 4.1s no Cypress e 604ms no Puppeteer. Essa discrepância é devida à maior complexidade dos testes realizados pelo Cypress, que frequentemente envolve interações mais detalhadas com a interface, como preenchimento de campos e validação de navegação entre páginas. Esse tempo mais longo no Cypress reflete um processo de teste que simula com maior precisão o comportamento real do usuário, considerando todas as interações e verificações necessárias. Por outro lado, o Puppeteer, focado na automação de navegador, consegue ser mais rápido, pois não realiza as interações detalhadas feitas pelo Cypress.

O teste final, que validou o comportamento do sistema durante o deslogamento do usuário, demonstrou tempos de execução de 1s no Cypress e 655ms no Puppeteer. Ambos os tempos são rápidos, indicando que o processo de deslogamento e redirecionamento para a tela de login está bem otimizado. A diferença de tempo entre as ferramentas é pequena, sendo atribuída principalmente à execução da navegação entre as páginas e à validação do redirecionamento.

Os testes realizados na tela de cadastro focaram na validação da funcionalidade dos principais campos presentes no formulário de registro do sistema, garantindo que o processo de criação de conta seja eficiente e seguro. Os campos testados incluem: **e-mail**, **nome completo**, **senha**, **confirmação de senha** e **sexo (gênero)**. Cada um desses campos possui requisitos específicos que precisam ser validados, como a formatação correta do e-mail, a verificação de correspondência entre a senha e sua confirmação, além da validação de um campo de seleção para o sexo do usuário.

O objetivo dos testes foi garantir que o sistema responda adequadamente em diferentes cenários, como preenchimento incompleto dos campos, dados inválidos ou a ausência de informações essenciais para a criação da conta. A seguir, na Tabela 2, serão apresentados os resultados desses testes, discutindo a eficácia da tela de cadastro e a experiência do usuário em cada situação simulada.

**Tabela 2. Segunda comparação de testes entre Cypress e Puppeteer (Tela de Cadastro de Usuários).**

| **Cypress** | **Resultado** | **Puppeteer** | **Resultado** |
| --- | --- | --- | --- |
| Deve mostrar erro ao enviar o formulário vazio. | 1s | Deve mostrar erro ao enviar o formulário vazio. | 1.0024s |
| Deve preencher o formulário corretamente. | 3.5s | Deve preencher o formulário corretamente. | 2.024s |
| Deve mostrar erro se as senhas não coincidirem. | 3.5s | Deve mostrar erro se as senhas não coincidirem. | 8.288s |
| Deve exibir o formulário. | 1.1s | Deve exibir o formulário. | 8.598s |

Fonte: o autor, 2024.

Os testes realizados na tela de cadastro têm como objetivo garantir a validação adequada dos campos essenciais para a criação de uma conta, como e-mail, nome completo, senha, confirmação de senha e sexo (gênero). Esses testes visam verificar o comportamento do sistema diante de diferentes entradas e cenários, incluindo o preenchimento incorreto ou a ausência de dados, bem como a validação da correspondência entre as senhas. A seguir, são discutidos os resultados obtidos a partir dos testes realizados com Cypress e Puppeteer.

O primeiro cenário analisado foi o envio do formulário sem preenchimento de nenhum campo. Tanto no Cypress (1s) quanto no Puppeteer (1.0024s), os tempos de execução foram praticamente idênticos, refletindo que o sistema responde de maneira eficiente ao tentar submeter um formulário vazio. A validação de erro nesse caso é crucial, pois garante que o usuário não consiga prosseguir com dados incompletos. O desempenho semelhante entre as duas ferramentas sugere que, nesse caso específico, ambos os frameworks são igualmente eficientes em realizar a verificação de campos obrigatórios e na apresentação das mensagens de erro adequadas.

Quando o formulário é preenchido corretamente, o tempo de execução no Cypress foi de 3.5s, enquanto no Puppeteer foi ligeiramente mais rápido, com 2.024s. A diferença nos tempos de execução pode ser explicada pela natureza dos testes: o Cypress, por ser um *framework* mais voltado para testes de interação com a interface do usuário, tende a realizar verificações adicionais no processo, como a validação de interações e mudanças no estado da interface. Já o Puppeteer, focado na automação de navegação em navegador, pode ser mais ágil, pois realiza as tarefas de forma mais direta e sem tanta interação com o usuário.

No teste em que as senhas não coincidiam, o Cypress apresentou um tempo de 3.5s, semelhante ao tempo do teste anterior com formulário preenchido corretamente. No Puppeteer, no entanto, o tempo foi consideravelmente maior, 8.288s. A discrepância entre os tempos de execução pode ser atribuída ao modo como cada ferramenta lida com os processos de validação e as interações de erro. O Cypress possivelmente realiza verificações mais detalhadas e complexas ao lidar com interações na interface, enquanto o Puppeteer pode estar enfrentando dificuldades para realizar as verificações de erro de forma eficiente, o que leva a um tempo de execução mais longo.

Ambos os *frameworks* apresentaram tempos de execução bastante rápidos para o teste de exibição do formulário: 1.1s no Cypress e 8.598s no Puppeteer. A diferença substancial nos tempos de execução sugere que o Puppeteer pode estar enfrentando um atraso no carregamento ou na renderização do conteúdo da página. Esse comportamento pode ser causado pela arquitetura do Puppeteer, que, ao contrário do Cypress, depende de uma comunicação direta com o navegador para realizar as ações e validações. O Cypress, por ser mais integrado ao ambiente de testes e trabalhar diretamente com o DOM da página, consegue exibir e interagir com o formulário de maneira mais rápida e eficiente.

**Conclusão(ões) ou Considerações Finais**

O título da seção Conclusão(ões) ou Considerações Finais deve ser alinhado à esquerda e grafado em negrito. Fica a critério do aluno e do orientador a escolha de qual termo melhor se adequa ao trabalho. Esta seção deve conter frases curtas, apresentando as conclusões e inferências elaboradas a partir da discussão dos resultados. É importante que estas frases não sejam construídas como meras reproduções dos resultados, respondendo aos objetivos propostos no trabalho. Os autores não devem, em hipótese alguma, mencionar, citar ou reproduzir resultados de outros estudos na(s) Conclusão(ões) ou Considerações Finais do TCC. Por fim, salienta-se que essa seção não deve conter tabelas ou figuras, sendo redigida de forma sucinta.

**Atenção:** antes de enviar o arquivo para o Sistema de TCCs, remova todas as instruções originais que estão abaixo do conteúdo dos tópicos.

**Agradecimento** (opcional, 1 parágrafo, bem sucinto)

O título da seção Agradecimentos deve ser alinhado à esquerda e grafado em negrito, com a primeira letra da palavra grafada em letra maiúscula. Trata-se de uma seção opcional, de no máximo três linhas, na qual o autor agradece àqueles que contribuíram de maneira relevante para o desenvolvimento do trabalho e elaboração do TCC, mas que não tiveram o envolvimento intelectual necessário à atribuição de coautoria.O autor deve se absterda menção ou citação dos nomes das empresas, instituições ou pessoas que permitiram ou contribuíram com o desenvolvimento do trabalho, a menos que esteja documentalmente autorizado a fazê-lo.

**Atenção:** antes de enviar o arquivo para o Sistema de TCCs, remova todas as instruções originais que estão abaixo do conteúdo dos tópicos.

**Referências**

Neste tópico deverão ser listadas todas as referências dos trabalhos citados no texto, formatadas seguindo rigorosamente as normas do MBA USP/Esalq. Para mais informações, consulte o “[Manual de Instruções e Normas para Trabalhos de Conclusão de Curso](https://www.dropbox.com/scl/fi/m1t43csoioys8x54pz60z/Manual-de-Instru-es-e-Normas-TCC_PT.pdf?rlkey=7isvkxss2lcl9f6gj9tkx9ipw&st=81xbs4gh&dl=0)” disponível no Sistema de TCCs.

**Atenção:** antes de enviar o arquivo para o Sistema de TCCs, remova todas as instruções originais que estão abaixo do conteúdo dos tópicos.

**Apêndice ou Anexo** (opcional)

Os apêndices são textos e/ou documentos que foram elaborados pelos autores e que são importantes para complementar a argumentação do trabalho. Anexos são textos ou documentos que ilustram o trabalho, mas que não foram elaborados pelos autores. Apêndices deverão seguir as mesmas normas de formatação do restante do texto, inclusive para as figuras e tabelas. O TCC deverá conter no máximo 30 páginas, incluindo o(s) Apêndice(s) e/ou Anexo(s).

**Atenção:** antes de enviar o arquivo para oSistema de TCCs, remova todas as instruções originais que estão abaixo do conteúdo dos tópicos.