

Selenium

Integrantes:

- Matheus Dias Mendes
- Lucas José Lopes Ferreira
- Olavo Vales Gomes de Castro Silva
- Marco Tullio Oliveira

Introdução

Este trabalho tem como objetivo demonstrar a utilização do framework Selenium para automação de testes em aplicações web.

Foi desenvolvida uma aplicação simples de calculadora para validar o funcionamento do framework e aplicar uma estratégia de derivação de casos de teste.

Descrição do Framework

O Selenium é um dos frameworks mais utilizados para automação de testes em aplicações web. Ele permite simular interações reais do usuário com o navegador, como cliques, preenchimento de formulários, navegação entre páginas e validação de elementos.

Possui suporte a diversos navegadores e a múltiplas linguagens de programação, incluindo Java, Python, C#, JavaScript, entre outras. É muito usado em cenários de testes funcionais e testes de regressão, garantindo que a aplicação web continue funcionando corretamente após alterações.

Categorização do Framework

i) Técnicas de Teste:

- **Caixa-preta:** O Selenium testa a aplicação sem precisar conhecer o código-fonte.
- **Baseado em especificação:** Foca no comportamento esperado da interface.

ii) Níveis de Teste:

- **Testes de Sistema:** valida o sistema como um todo através da interface web.
- **Testes de Aceitação:** pode ser usado para validar requisitos de negócio junto ao cliente.

iii) Tipos de Teste:

- **Teste Funcional:** garante que as funcionalidades da aplicação web funcionam como esperado.

- **Teste de Regressão:** executado após modificações para verificar se nada foi quebrado.
- **Testes de Integração:** quando a aplicação depende de APIs ou backends, o Selenium pode validar se a integração funciona via UI.

Instalação/Integração (Java)

Para utilizar o Selenium com Java, siga os passos:

1. Instalar o JDK (Java Development Kit) e configurar o ambiente.
2. Instalar o Maven (ou Gradle) para gerenciar dependências.
3. Criar um projeto Java com maven
4. Adicionar dependências do Selenium no pom.xml
5. Baixar o WebDriver do navegador e colocá-lo no PATH do sistema.
6. Criar classes de teste que abram o navegador e interajam com a aplicação.
7. Executar o teste
 - mvn compile
 - mvn exec:java

Estratégia de Derivação de Casos de Teste

Neste trabalho, será usada a estratégia de Particionamento de Equivalência.

Aplicando à Calculadora Web:

- **Casos válidos:**
 - $2 + 2 = 4$
 - $6 - 3 = 3$
 - $5 * 2 = 10$
 - $8 / 2 = 4$
- **Casos inválidos:**
 - $7 / 0 = \text{erro}$ (divisão por zero)
 - Entrada de caracteres não numéricos ($a + b = \text{erro}$)
 - Campo vazio (sem valores inseridos)

Conclusão

Através do uso do Selenium, foi possível automatizar testes funcionais na calculadora web, garantindo que as operações fossem validadas de forma rápida e confiável. O framework demonstrou grande eficiência na simulação de interações do usuário e na verificação dos resultados esperados.