

Caso de Teste – Testes de Alertas e Pop-ups

Testador Responsável: Heitor Louzeiro

1. Identificador do Caso de Teste

- ID: CT-Alert-001

2. Nome do Caso de Teste

- Nome: Verificar funcionamento de alertas e pop-ups JavaScript

3. Descrição

- Descrição Geral: Este caso de teste verifica a funcionalidade de alertas e pop-ups JavaScript, testando tanto o cenário de confirmação de alerta quanto o cenário de captura de prompt. O teste automatizado utiliza Selenium WebDriver para simular as interações do usuário com os alertas.

4. Pré-Condições

- Pré-requisitos:
 1. O usuário deve ter acesso à internet
 2. Chrome WebDriver deve estar instalado e configurado
 3. Python e Selenium WebDriver devem estar instalados
 4. O site de teste (the-internet.herokuapp.com) deve estar acessível

5. Entradas (Dados de Teste)

- Entradas Necessárias:
 - Cenário de Confirmação:
 - Nenhuma entrada, apenas interação com o alerta
 - Cenário de Prompt:
 - Texto a ser inserido no prompt: "Teste Automatizado"

6. Procedimentos (Passo a Passo)

1. Abrir o navegador Chrome
2. Acessar a URL https://the-internet.herokuapp.com/javascript_alerts
3. Para teste de confirmação de alerta:
 - Localizar o botão que gera o alerta de confirmação
 - Clicar no botão
 - Aceitar o alerta
 - Verificar mensagem de sucesso
4. Para teste de prompt:
 - Localizar o botão que gera o prompt

- Clicar no botão
- Inserir texto no prompt
- Confirmar o prompt
- Verificar mensagem de resultado

7. Saída Esperada

- Resultado Esperado:
 - Cenário de Confirmação:
 - Mensagem "You clicked: Ok" deve ser exibida
 - Cenário de Prompt:
 - Mensagem "You entered: Teste Automatizado" deve ser exibida

8. Pós-Condições

- Estado do Sistema após o Teste:
 - O navegador deve ser fechado após a conclusão dos testes

9. Observações / Comentários

- Observações Finais:
 - O teste utiliza esperas explícitas para garantir que os elementos estejam presentes
 - Pausas estratégicas (time.sleep) foram adicionadas para melhor visualização
 - Os testes são independentes e podem ser executados separadamente

10. Resultado do Teste

- Data da Execução: 19/02/2024
- Ambiente de Teste: Máquina Local
- Status: Aprovado
- Evidências:
 - Os resultados são verificados através de asserções do unittest
 - Mensagens de sucesso/erro são validadas automaticamente
 - O console exibe o passo a passo da execução com pausas para visualização

```

test_confirm_alert (__main__.TestAlerts.test_confirm_alert)
Teste de confirmação de alerta JavaScript. ...
=====
Iniciando teste de alertas e pop-ups...
=====

DevTools listening on ws://127.0.0.1:50425/devtools/browser/c1bf5c30-9582-41ee-865d-a3df458c887b

Executando teste de confirmação de alerta:
-----
1. Localizando botão de alerta de confirmação...
2. Clicando no botão para gerar alerta...
3. Aceitando o alerta...
4. Verificando resultado...
   ✓ Confirmação do alerta bem-sucedida!

>>> Teste de confirmação de alerta PASSOU!

Finalizando teste e fechando navegador...
=====
ok

```

```

-----
Executando teste de prompt:
-----
1. Localizando botão de prompt...
1. Localizando botão de prompt...
2. Clicando no botão para gerar prompt...
3. Interagindo com o prompt...
   - Inserindo texto: 'Teste Automatizado'
   - Inserindo texto: 'Teste Automatizado'
4. Confirmando o prompt...
4. Confirmando o prompt...
5. Verificando resultado...
   ✓ Texto do prompt capturado corretamente!

>>> Teste de prompt PASSOU!

Finalizando teste e fechando navegador...
=====
ok

```

11. Artefato

- Código implementado: main.py
 - Implementação usando Python + Selenium WebDriver
 - Framework de teste: unittest
 - Localização: ct_alert_001/main.py
 - Recursos:
 - Esperas explícitas (WebDriverWait)
 - Pausas para visualização (time.sleep)
 - Mensagens detalhadas no console