



Atividade Individual – Identificação de Elementos

| QA Practice |

A seguir, apresento três formas diferentes de mapeamento, cada uma baseada em uma estratégia distinta, escolhidas conforme a estrutura e a organização dos elementos HTML, alinhadas aos conceitos abordados na trilha de aprendizado.

Estratégia 1: Seleção pelo Texto Exibido (XPath com texto estático)

Elemento identificado: Botão “Create Product”

Localizador adotado:

```
1 Click Element    xpath=//a[text()='Create Product']
```

Motivação da escolha:

Optei por utilizar o conteúdo textual visível como critério de localização, uma prática recomendada quando o texto é constante e previsível. Essa técnica torna o seletor mais direto e de fácil manutenção, evitando caminhos complexos e suscetíveis a alterações. Também respeita a boa prática de evitar o uso de XPath absoluto.

Estratégia 2: Seleção com Seletor CSS por Hierarquia

Elemento identificado: Cabeçalho da tabela de produtos (“Product Name”)

Localizador adotado:

```
1 ${title}=    Get Text    css=thead > tr > th:nth-child(1)
```

Motivação da escolha:

Utilizei um seletor CSS baseado na estrutura hierárquica dos elementos HTML, com o operador `>` para indicar relações diretas entre os nós. Essa abordagem segue as boas práticas de priorizar seletores CSS curtos, objetivos e fáceis de interpretar.

Estratégia 3: Seleção por Posição na Estrutura (XPath com índice)

Elemento identificado: Nome do segundo produto listado (segunda linha da tabela)

Localizador adotado:

```
1 ${product}=    Get Text    xpath=//tbody/tr[2]/td[1]
```

Motivação da escolha:

Empreguei um XPath com índice para acessar diretamente a célula desejada da tabela. Esse método é eficaz quando se conhece a exata posição dos dados dentro da estrutura. Embora seja menos adaptável a alterações na interface, mostra-se adequado para cenários controlados e com layout estável, como neste caso.