

Atividade Individual – Mapeamento de Elementos

Site utilizado: [QA Practice](#) | [Learn with RV](#)

Apresentei abaixo três mapeamentos distintos, cada um utilizando uma estratégia diferente, de acordo com a hierarquia e estrutura dos elementos HTML e os princípios aprendidos na trilha.

Estratégia 1: Mapeamento por Texto Visível (XPath com texto fixo) [🔗](#)

Elemento mapeado: Botão “Create Product”

Localizador utilizado:

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

Justificativa:

Utilizei o texto visível como critério de seleção, conforme recomendado quando o conteúdo é fixo e previsível. Essa estratégia simplifica o seletor e facilita a manutenção, evitando caminhos longos e frágeis. Segui a recomendação de evitar XPath absoluto.

Estratégia 2: Mapeamento por Seletor CSS Hierárquico [🔗](#)

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

Localizador utilizado:

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

Justificativa:

Apliquei um seletor CSS baseado na hierarquia dos elementos, utilizando combinadores `>` para indicar relação direta entre os elementos HTML. Essa abordagem é alinhada à recomendação de priorizar CSS claros e curtos sempre que possível.

Estratégia 3: Mapeamento por Estrutura Posicional (XPath com índices) [🔗](#)

Elemento mapeado: Nome do segundo produto da lista (linha 2 da tabela)

Localizador utilizado:

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

Justificativa:

Usei XPath com índice para navegar pela tabela de forma precisa. Essa estratégia é útil quando se sabe exatamente qual linha/coluna deve ser acessada. Embora seja menos flexível a mudanças no layout, pode ser adequada em cenários estáveis e controlados como este.