Estratégia de Mapeamento de Elementos - Squad Codetectives

Esta atividade em grupo teve como objetivo mapear os botões e os elementos da tabela da página The Internet , utilizando boas práticas para garantir precisão, estabilidade e facilidade de manutenção em automações de teste.

Hierarquia adotada para mapeamento

A squad seguiu a seguinte ordem de prioridade para selecionar os melhores localizadores:

1. **ID** – Quando presente e único. No entanto, não foi utilizado nesta página, pois os IDs são inexistentes ou instáveis.
2. **Seletor customizado** – Criado a partir de atributos estáveis como `class`, `data-*`, `onclick`, etc.
3. **Name** – Quando presente e único. Também não aplicável nesta página.
4. **Class name** – Utilizado quando a classe é específica, significativa e consistente.
5. **CSS Selectors** – Utilizados com base na hierarquia do DOM, uso de `nth-child`, ou combinações de classes. São preferidos por serem mais rápidos e legíveis.
6. **XPath sem texto/index** – Usado quando o CSS não é suficiente, priorizando caminhos relativos e atributos em vez de índices ou textos visíveis.
7. **Link text ou partial link text** – Usado apenas quando o texto é exclusivo e estático (ex: "edit", "delete").
8. **XPath com texto/index** – Utilizado como último recurso, quando nenhum dos anteriores é aplicável. Deve ser evitado sempre que possível.

Considerações importantes

- Se um atributo (como `id`) muda a cada recarregamento da página, ele não deve ser utilizado como base para mapeamento, mesmo que pareça ideal.
- A prioridade é usar seletores resistentes a alterações estruturais ou visuais, evitando caminhos absolutos e dependência de texto.

Exemplos práticos

Botões

Botão	*CSS Selector*	*XPath*
Azul (padrão)	`.button:not(.alert):not(.success)`	`//a[@class='button']`
Vermelho	`.button.alert`	`//a[contains(@class, 'alert')]`
Verde	`.button.success`	`//a[contains(@class, 'success')]`

Tabela

Elemento	*CSS Selector*	*XPath*
Toda a tabela	`table`	`//table`
Todas as linhas	`table tbody tr`	`//table/tbody/tr`
Segunda célula da primeira linha	`table tbody tr:first-child td:nth-child(2)`	`//table/tbody/tr[1]/td[2]`
Link "edit" da primeira linha	`table tbody tr:first-child td:last-child a:first-child`	`//table/tbody/tr[1]/td[last()]/a[contains(text(), 'edit')]`
Link "delete" da primeira linha	`table tbody tr:first-child td:last-child a:last-child`	`//table/tbody/tr[1]/td[last()]/a[contains(text(), 'delete')]`

Exemplo em Robot Framework

```
*** Settings ***
Library    SeleniumLibrary

*** Test Cases ***
Clicar Nos Botoes
    Open Browser    https://the-internet.herokuapp.com/challenging_dom     chrome
    Click Element    css:.button
    Click Element    css:.button.alert
    Click Element    css:.button.success
    [Teardown]    Close Browser
```