

WEB AUTOMATION

Cómo reconocer los elementos para automatizar con Selenium

Cuando automatizas pruebas con Selenium, el primer paso es identificar los elementos de la página web con los que deseas interactuar.

Las páginas web están estructuradas con **HTML**, donde cada elemento (botones, formularios, campos de entrada, etc.) tiene atributos que permiten identificarlos. Algunos de los más importantes son:

Atributo	Descripción
id	Identificador único del elemento.
class	Nombre(s) de la clase CSS asociada.
name	Nombre del campo (útil en formularios).
value	Valor del campo (como el texto de un botón).

Ejemplo de un botón en HTML:

```
<input type="button" id="searchButton" name="search" class="btn">
```

Para interactuar con este botón en Selenium, podemos usar diferentes estrategias de localización.

Localizadores en Selenium

Los **localizadores** permiten encontrar elementos dentro del código HTML. Algunos de los más comunes son:

1. **Por ID** (`By.id("searchButton")`)
2. **Por Nombre** (`By.name("search")`)
3. **Por Clase CSS** (`By.className("btn")`)
4. **Por Selectores CSS avanzados** (`By.cssSelector(...)`)

¿Qué son los localizadores personalizables?

Los **localizadores personalizables** en Selenium permiten seleccionar elementos de manera más precisa cuando los atributos básicos como **id** o **class** no son suficientes o no están disponibles.

Estos localizadores aprovechan combinaciones avanzadas de selectores **CSS** para identificar elementos específicos con mayor flexibilidad.

Ejemplos de personalización:

- Selección por **atributos específicos**.
- Combinación de **etiquetas, clases y atributos**.
- Uso de **pseudoclases** para seleccionar elementos dinámicos.

Selectores CSS en Selenium

Los **selectores CSS** permiten ubicar elementos de forma más flexible y precisa. Se pueden utilizar en Selenium con `findElement(By.cssSelector())`.

Principales selectores CSS

Selector	Descripción
Etiqueta (tag)	Selecciona directamente con el nombre de la etiqueta.
ID	Se indica con <code>#</code> seguido del id del elemento.
Clase	Se indica con <code>.</code> seguido del nombre de la clase.
Atributos	Se usan <code>[]</code> para especificar un atributo y su valor.

Ejemplos de uso en Selenium:

```
// Selecciona un elemento por etiqueta
WebElement tag = driver.findElement(By.cssSelector("input"));

// Selecciona un elemento por ID
WebElement tagId = driver.findElement(By.cssSelector("input#searchButton"));

// Selecciona un elemento por clase
WebElement tagClass = driver.findElement(By.cssSelector("input.btn"));

// Selecciona un elemento por atributo específico
WebElement tagAttr = driver.findElement(By.cssSelector("a[attribute='value']"));

// Combinación de clase y atributo
WebElement tagClassAttr =
driver.findElement(By.cssSelector("a.btn-success[data-testid='loginButton']"));
```

Acciones sobre los elementos en Selenium

Una vez localizado un elemento, se pueden realizar acciones como:

- **Hacer clic en un botón:** `tagId.click();`
- **Escribir en un campo de entrada:** `tag.sendKeys("usuarioEjemplo");`

Estos métodos permiten simular la interacción de un usuario real con la aplicación web.

El uso de localizadores en Selenium es clave para interactuar de manera efectiva con los elementos de una página web. Si bien los **ID** y **clases** suelen ser suficientes, los **selectores CSS avanzados** permiten personalizar la localización cuando los atributos estándar no son suficientes.

Al comprender cómo funcionan los selectores CSS y cómo personalizarlos, puedes crear pruebas más robustas y flexibles para diferentes estructuras web.