

WEB AUTOMATION

¿Cómo reconocer los elementos para automatizar con Selenium?

Para automatizar tareas web con Selenium, es fundamental comprender cómo interactúa con los elementos HTML. Cada componente de una página web, como botones, enlaces y campos de texto, está definido en el código HTML. Selenium utiliza distintos métodos para identificarlos en el DOM (*Document Object Model*) y manipularlos eficientemente.

Estructura de los elementos HTML

Los elementos HTML se organizan mediante etiquetas, cada una con atributos que proporcionan información adicional, como `id`, `class` y `name`. Por ejemplo:

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

En este caso, los atributos `id` y `name` permiten a Selenium identificar y manipular el botón de manera precisa.

¿Qué son los localizadores?

Los **localizadores** son herramientas que permiten encontrar elementos dentro de una página web. Son esenciales para la automatización de pruebas, ya que permiten interactuar con los elementos, como ingresar texto, hacer clic en botones y verificar contenido.

Selenium ofrece varios tipos de localizadores, cada uno con diferentes niveles de especificidad y utilidad.

Clasificación de los localizadores

A continuación, se presentan los principales localizadores en Selenium:

1. ID (más específico y confiable) :El atributo `id` es único en la página, lo que lo convierte en el método más eficiente para localizar elementos.

```
<input type="text" id="searchButton">
WebElement searchField = driver.findElement(By.id("searchField"));
```

2. Name (útil cuando no hay ID disponible): El atributo `name` suele usarse en formularios, aunque puede haber duplicados en la página.

```
<input type="text" name="search">
WebElement searchBox = driver.findElement(By.name("search"));
```

3. Link Text (para enlaces con texto exacto): Permite identificar enlaces por su texto visible.

```
<a href="/login.html">Iniciar sesión!</a>
WebElement loginLink = driver.findElement(By.linkText("Iniciar sesión"));
```

4. Partial Link Text (para enlaces con texto parcial): Permite localizar enlaces con solo una parte del texto visible.

```
WebElement helpLink = driver.findElement(By.partialLinkText("Ayuda"));
```

5. Class Name (útil para elementos con clases específicas): Se basa en el atributo `class`, aunque debe usarse con precaución si hay múltiples elementos con la misma clase.

```
<input type="text" class="search">
WebElement searchField = driver.findElement(By.className("search"));
```

6. Tag Name (para obtener listas de elementos): Identifica elementos por su etiqueta HTML. Se usa generalmente para trabajar con múltiples elementos a la vez.

```
<input type="text">
List<WebElement> inputs = driver.findElements(By.tagName("input"));
```

¿Cómo elegir el localizador más adecuado?

Para garantizar pruebas estables y resistentes a cambios en la página, es fundamental elegir el localizador con la mayor **especificidad** posible. Cuanto más específico sea el localizador, menor será el riesgo de que la prueba falle debido a modificaciones en la estructura del HTML.

El orden recomendado de preferencia es el siguiente:

- ❶ **ID** – La opción más precisa y rápida.
- ❷ **Name** – Alternativa cuando no hay **id** disponible.
- ❸ **Link Text** – Solo para enlaces con texto exacto.
- ❹ **Class Name** – Puede haber múltiples elementos con la misma clase.
- ❺ **Tag Name** – Útil para obtener listas de elementos similares.

Siempre es preferible utilizar el localizador más específico y único posible para evitar fallos en las pruebas.

¿Cómo se localiza?

El método **findElement** permite buscar y seleccionar un elemento en la página. Devuelve un objeto **WebElement** que puede ser manipulado con diferentes acciones.

Ejemplos con distintos localizadores:

♦ Por ID (más específico)

```
WebElement submitButton = driver.findElement(By.id("submit"));
```

♦ Por Name

```
WebElement searchField = driver.findElement(By.name("search"));
```

♦ Por Link Text y Partial Link Text

```
WebElement contactLink = driver.findElement(By.linkText("Contacto"));
WebElement helpLink = driver.findElement(By.partialLinkText("Ayuda"));
```

♦ Por Class Name

```
WebElement description = driver.findElement(By.className("description"));
```

- ♦ **Por Tag Name** (para múltiples elementos)

```
List<WebElement> inputs = driver.findElements(By.tagName("input"));
```

¿Cómo probar los localizadores?

Una vez identificados los elementos, es fundamental probar su interacción con Selenium. Algunas de las acciones más comunes incluyen:

- ✓ **Hacer clic en un botón**

```
driver.findElement(By.id("login")).click();
```

- ✓ **Ingresa texto en un campo**

```
driver.findElement(By.name("username")).sendKeys("tuUsuario");
```

- ✓ **Obtener y validar un texto**

```
String mensajeBienvenida = driver.findElement(By.id("welcome")).getText();  
assertEquals("Bienvenido, Usuario!", mensajeBienvenida);
```

Implementar correctamente los localizadores garantizará pruebas automatizadas más eficientes y robustas.

📌 **Recomendación:** Para más detalles, consulta la [documentación oficial](#).