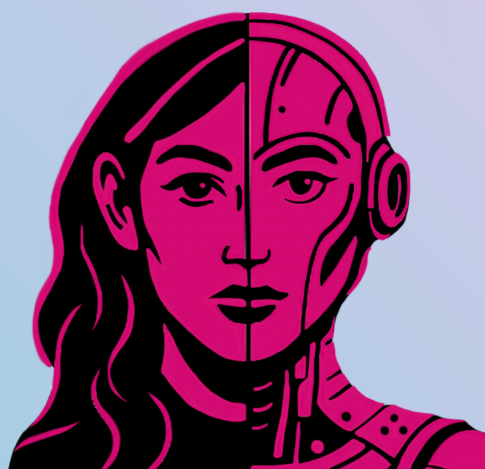


LOCALIZADORES

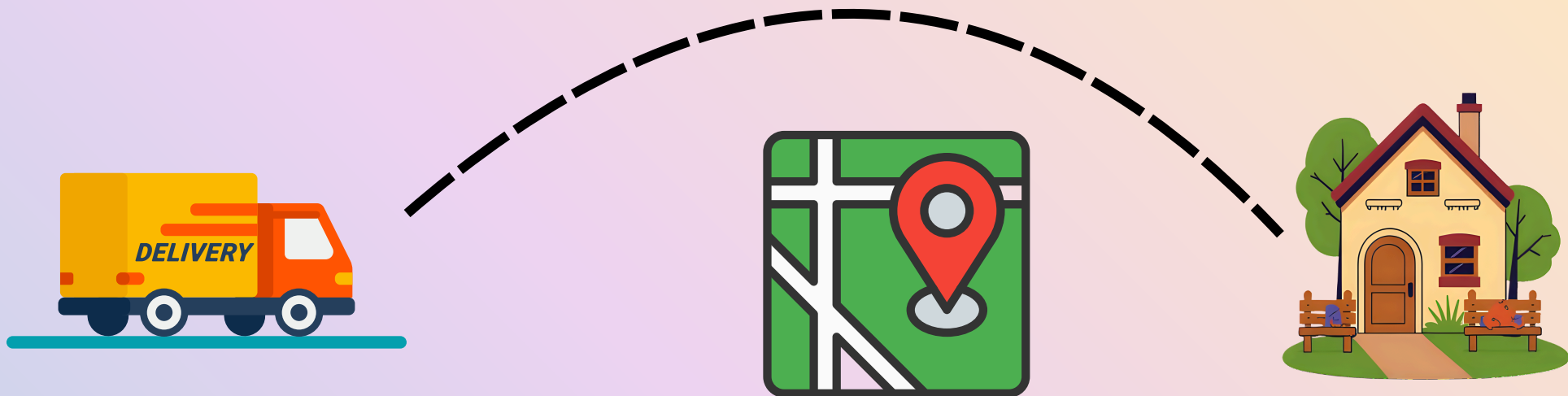
Para automatización de pruebas

Mujeres Testing – Latam



¿Por qué localizadores?

Si haces un **pedido** online, debes indicar tu **dirección** para que llegue a tu **casa**.



Si haces una **prueba manual**, por ejemplo, de log in, sigues una lista de pasos **interactuando** manualmente con los **elementos** dentro de la página web.

Si haces una **prueba** automatizada, debes indicarle a la herramienta cómo **localizar** los **elementos** para **interactuar** con ellos.

Por ejemplo, para probar el botón “log in”, necesitas indicar el identificador (localizador) de este elemento para hacerle clic (interacción).



¿Qué es un localizador?

Es la manera en que una herramienta de automatización identifica un elemento HTML dentro del DOM* para interactuar con él.

Ejemplos simples:

- El botón “Log in”
- El campo “Correo electrónico”
- El menú principal
- El ícono del carrito

Para las personas es fácil reconocer estos elementos visualmente, pero para una herramienta de automatización es necesario usarlos basándose en atributos y estructura del código para identificarlos correctamente.

*DOM (Documento Object Model) es la estructura en forma de árbol que representa todos los elementos de una página web.



Consideraciones al automatizar casos

- Los localizadores son la base de todo script de pruebas.
- Si un localizador es frágil, la prueba será frágil.

Un buen localizador permite:

- Evitar falsos negativos.
- Reducir flakiness (inconsistencia/inestabilidad).
- Evitar mantenimiento innecesario.
- Hacer scripts legibles.
- Soportar cambios en la UI.

Recuerda:

Asegúrate de que el localizador apunte a un solo elemento, de lo contrario la prueba fallará.



Buenas prácticas para crear tus localizadores

Encapsula

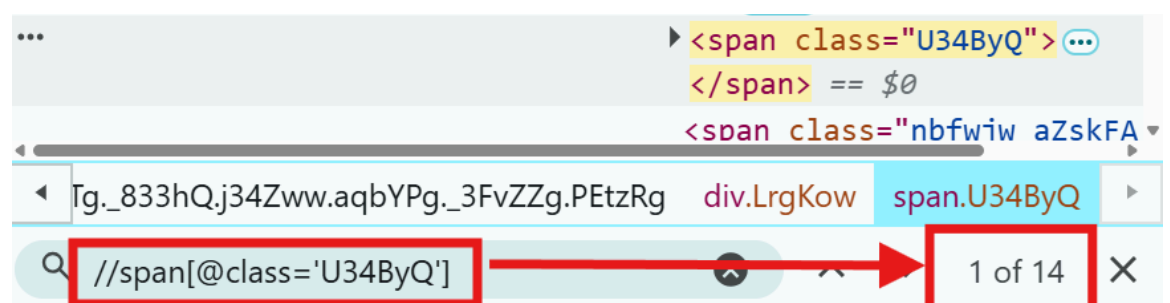
No repitas los localizadores en cada test, la centralización es clave para facilitar mantenimiento y evitar duplicación.

Prioriza localizadores basados en accesibilidad (atributos **ARIA***): Aporta estabilidad, mejor legibilidad y refuerza la accesibilidad para los usuarios, reflejando interacciones reales.

*ARIA: Accessible Rich Internet Applications. Conjunto de atributos que se añaden a las páginas web para mejorar la accesibilidad.

Unicidad: Asegúrate de que tus localizadores seleccionen sólo un elemento, de lo contrario la prueba fallará.

En este caso el localizador XPath es muy genérico, apuntando a 14 posibles elementos.



Buenas prácticas para crear tus localizadores

Evitar localizadores **inestables** como textos variables por marketing, IDs **autogenerados** o clases **dinámicas** disminuye la probabilidad de pruebas frágiles.

Solicitar **atributos** específicos como **data-testid** proporciona una forma estable, independiente de estilo para ubicar elementos, especialmente útil en proyectos con flujos complejos o sin enfoque fuerte en accesibilidad.

*ARIA: Accessible Rich Internet Applications. Conjunto de atributos que se añaden a las páginas web para mejorar la accesibilidad.



Algunas señales de que tu localizador NO es bueno

Es muy largo: Demasiado sensibles a cambios estructurales mínimos. Especialmente XPath absolutos o muy anidados.

Mala practica

```
/html/body/div[1]/main/div/section[2]/div/form/input[1]
```

Ejemplo de recomendación

```
page.getByRole('button', {  
  name: 'Comprar' })
```

Depende de una clase CSS de estilo: Depende de apariencia visual y no propósito funcional.

Mala practica

```
button.btn-success
```

Ejemplo de recomendación

```
page.getByRole('button', { name:  
  'Comprar' })
```

Usa índice (nth-child(n) o [n]): Depende de la posición del elemento.

Mala practica

```
ul > li:nth-child(3) > a
```

Ejemplo de recomendación

```
a[data-nav-item="contacto"]
```



Algunas señales de que tu localizador NO es bueno

Usa ids generadas automáticamente: Volátiles y cambian con cada compilación.

Mala practica

```
button.c-012b-abc-12-a
```

Ejemplo de recomendación

```
[data-testid="boton-enviar-datos"]
```

Nadie entiende qué selecciona: La falta de legibilidad complica la mantención y la depuración del código de pruebas.

```
div > input:focus + span
```

```
page.getByLabel('Contraseña')
```



No utilices índice para localizadores

En lugar de usar la posición (índice) del elemento dentro del DOM (Document Object Model), se debería identificar el elemento por su contenido o por un atributo estable.

En vez de posición...

```
//button[4]
```

Usa el rol y el texto accesible.

```
page.getByRole('button',{ name: 'Comprar' })
```

En vez de posición...

```
ul > li:nth-child(5)
```

Usa el texto visible del elemento

```
page.getByText('Detalle del  
Producto')
```

En vez de posición...

```
//*[@class='item'])[4]
```

Usa un atributo único

```
page.getByTestId('art-promocion')
```

No recomendamos utilizar índice, ya que si el dev añade un nuevo ítem, todos los elementos siguientes se desplazarán, haciendo que la prueba falle.



Distintos tipos de localizadores

Ordenados por prioridad

Localizadores Semánticos o Accesibles (ARIA):

Descripción

Es la forma en que los usuarios interactúan con la interfaz.

Ventaja

Son los más robustos y más valiosos porque validan que el elemento sea accesible y funcionalmente correcto.

Desventaja

Solo funcionan si el elemento tiene un rol semántico correcto (ej. un botón, un enlace, un encabezado).

```
page.getByRole('button', { name: 'Iniciar Sesión' })
```

Basados en Contenido Visible:

Descripción

Basado en el texto o atributo visible para el usuario (etiqueta, placeholder, texto, alt, title).

Ventaja

Son altamente legibles y menos propensos a romperse por cambios de estilo.

Desventaja

El texto visible puede cambiar por una refactorización de copy o traducción, lo que requiere actualizar el localizador.

Ejemplos

- `page.getByText('Iniciar sesión')`
- `page.getByLabel('Contraseña')`
- `page.getByPlaceholder('Escribe tu búsqueda aquí')`



Distintos tipos de localizadores

Atributos Específicos para Testing (getByTestId):

Descripción

Un atributo HTML personalizado (ej. data-testid) que se añade al elemento únicamente para propósitos de automatización. Es un contrato de estabilidad entre QA y Desarrollo.

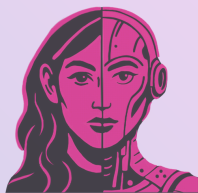
Ventaja

Son extremadamente estables, rápidos de buscar y son legibles. Desacoplan la prueba de las clases de estilo.

Desventaja

Requiere coordinación con el equipo de desarrollo para que los añadan y se comprometan a no cambiarlos sin previo aviso.

```
page.getByTestId('direcciones')
```



Distintos tipos de localizadores

CSS Selectors

Descripción

Se basan en atributos nativos como id, class u otros selectores CSS. Son la forma más común de localizar elementos si los métodos semánticos fallan.

Ventaja

Son rápidos, concisos y están bien soportados por todos los navegadores y frameworks.

Desventaja

Son propensos a la fragilidad si dependen de clases CSS de estilo, clases autogeneradas o combinaciones de anidación largas.
**Playwright no los recomienda.*

```
page.locator('#login-button')  
page.locator('.producto-card')
```

XPath (XML Path Language)

Descripción

Un lenguaje de consulta diseñado para navegar por elementos y atributos de un documento XML o HTML. Permite búsquedas complejas (ej. buscar el padre de un elemento o texto parcial).

Ventaja

Es el más flexible y potente; puede alcanzar elementos inalcanzables con CSS (como buscar texto contenido).

Desventaja

Es el menos legible, el más lento de evaluar para el navegador y tiene el mayor riesgo de fragilidad posicional y mantenimiento elevado. Se considera el "último recurso".

```
page.locator('//button[@id="login"]')
```



Beneficios de GetByRole

Alineado con Accesibilidad y experiencia de usuario.
Nos permite localizar elementos por como los usuarios los ven.

Imita la interacción del usuario

Un usuario no ve un div con `data-testid="login-button"`, ve un botón que dice "Iniciar Sesión".

```
page.getByRole('button', {name: 'Iniciar Sesión'})
```

Robustez Inherente

A demás de evaluar que el elemento hace lo que debe hacer, también confirmas que la accesibilidad se mantenga.

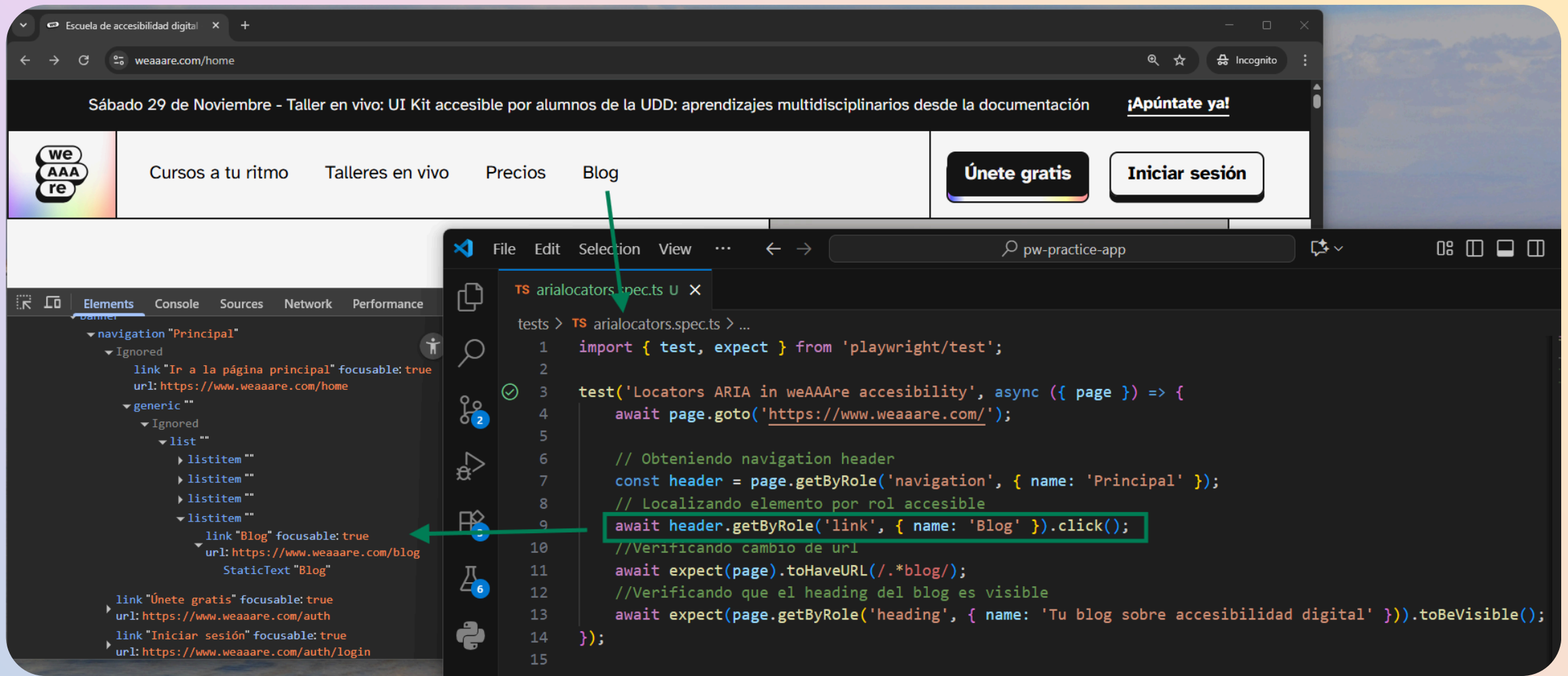
Usa estándares ARIA

Identifica elementos según su función real:
button, textbox, checkbox, link, heading, list, table.

Lo ideal sería que todos los proyectos tuviesen foco en la accesibilidad, pero considerando que esto no es así en todos los casos, puedes utilizar los otros tipos de localizadores.



Caso de ejemplo

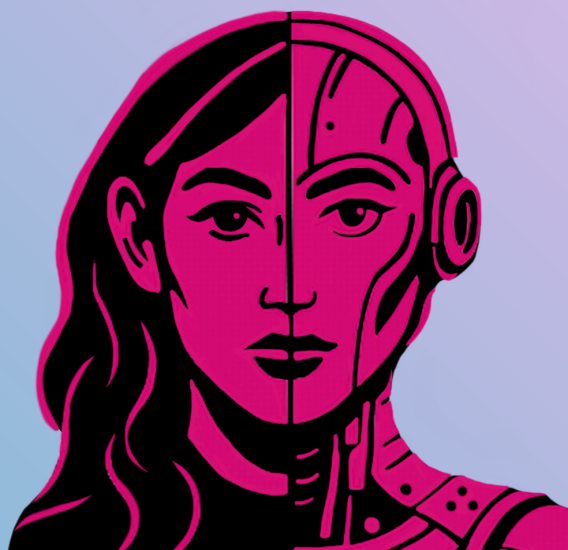


Sitio web utilizado para el ejemplo: <https://www.weaaare.com/app>



Conectando y apoyando a
Mujeres en LATAM para crecer
juntas en el camino del Testing

Mujeres Testing – Latam



Noviembre, 2025