

PUPPETEER





¿Qué es Puppeteer?

Puppeteer es una librería de Node.js desarrollada por el equipo de Google que permite controlar el navegador Chrome o Chromium mediante código.

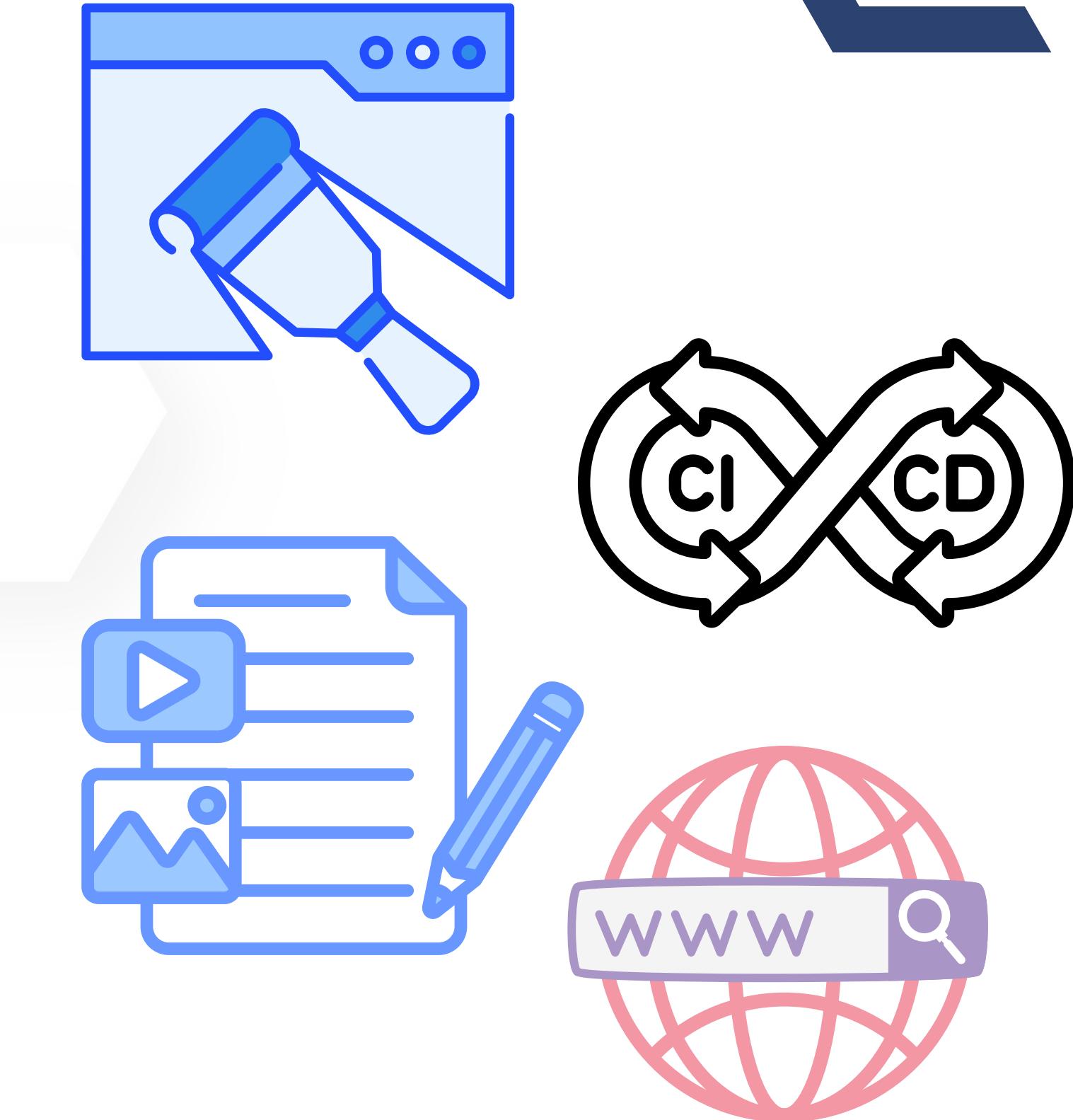
Usos:

Web Scraping y Extracción de datos

Testeo Automatizado

Generación de Contenido

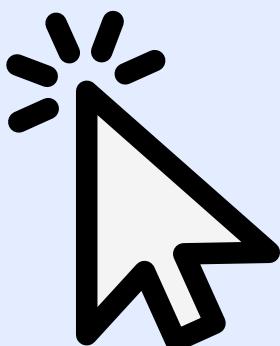
Monitorización de rendimiento y SEO*



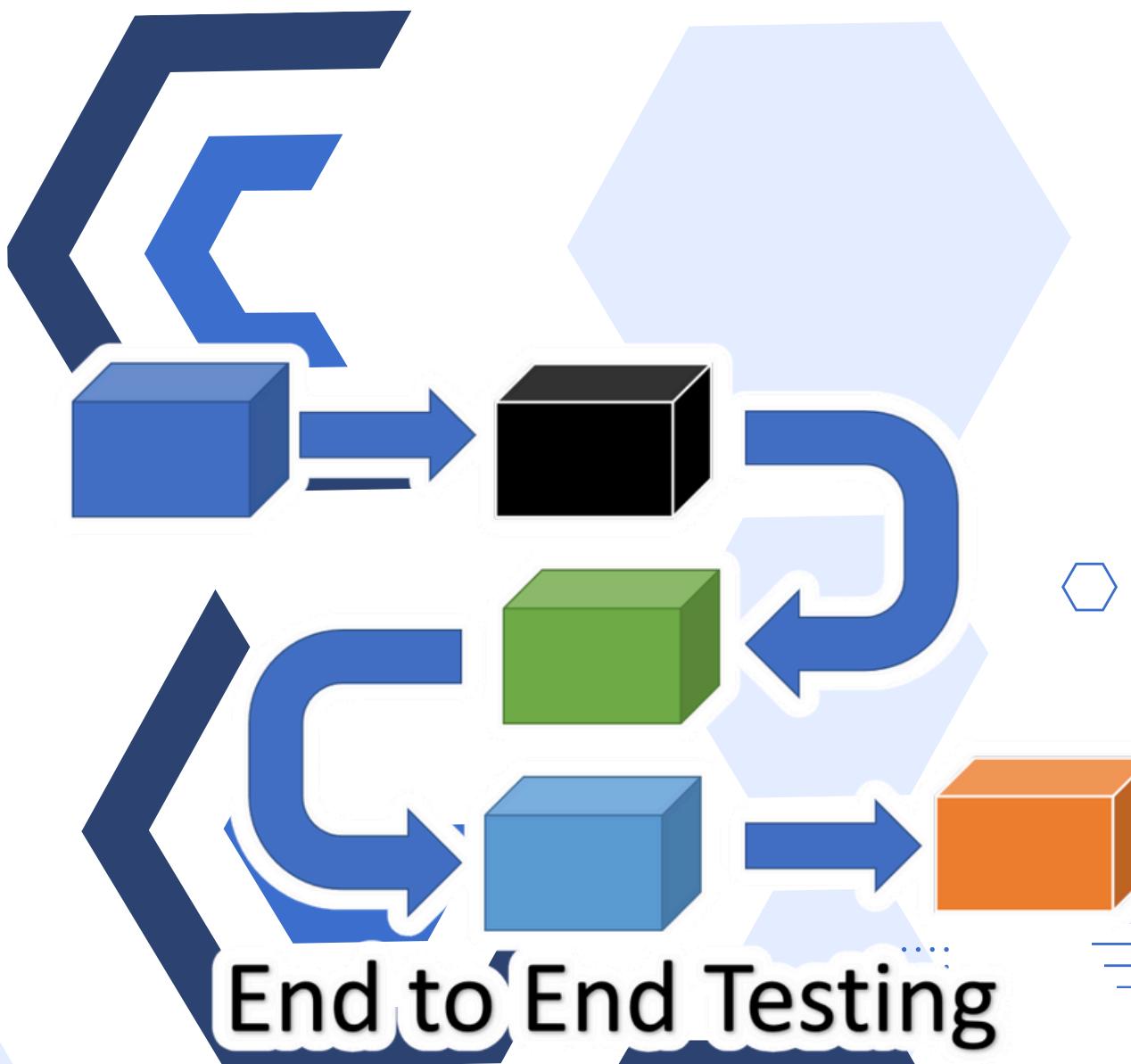
*SEO: *Search Engine Optimization - Optimización para motores de búsqueda*

¿Qué son las pruebas E2E?

Son una metodología para probar una aplicación completa, simulando la experiencia del usuario real para verificar que todos los componentes funcionan correctamente juntos, desde el principio hasta el final del flujo



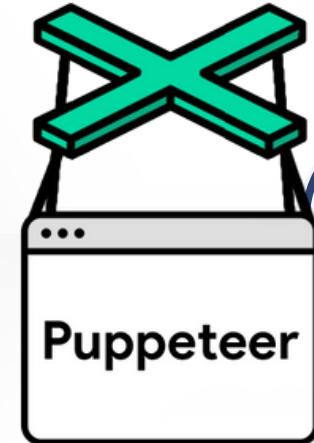
Con puppeteer se hace la simulación de interacciones, tales como clicks, inputs de teclado y/o navegación, para asegurar funcionalidad.



Iniciando a usar puppeteer

1. Requisito principal

Descargar Node.js (<https://nodejs.org/en/download>)



2. Instalar puppeteer:

`npm install puppeteer`

Iniciando a usar puppeteer

3. Probar el funcionamiento

```
const puppeteer = require('puppeteer');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://example.com');
  console.log(await page.title());
  await browser.close();
})();
```

Conceptos clave

Funcionalidades básicas:

- **launch()** – inicia el navegador
- **newPage()** – abre pestañas
- **goto(url)** – navegar
- **click(selector)** – clicks
- **type(selector, text)** – escritura
- **waitForSelector()** – sincronización
- **evaluate()** – ejecutar JS en la página

Conceptos clave

Selección de elementos

La selección correcta de elementos es la base de toda automatización E2E.

Permite:

- Interactuar (click, type, hover)
- Obtener valores
- Esperar estados de la UI

Tipos de Selectores:

- Selector CSS
- Selector por texto (evaluate o XPath)
- Selector personalizado

Conceptos clave

SELECTOR CSS

- Son los más usados. Simples, rápidos y compatibles con todo.

SELECTOR POR TEXTO

- Cuando no hay IDs ni clases útiles.



```
await page.click('#btnLogin');           // ID  
await page.type('.input-email', 'test'); // Clase  
await page.click('button.primary');     // Tag + clase  
await page.click('input[name="email"]'); // Atributo
```



```
await page.$x("//button[contains(text(), 'Enviar')]");
```

Conceptos clave

SELECTOR

PERSONALIZADO

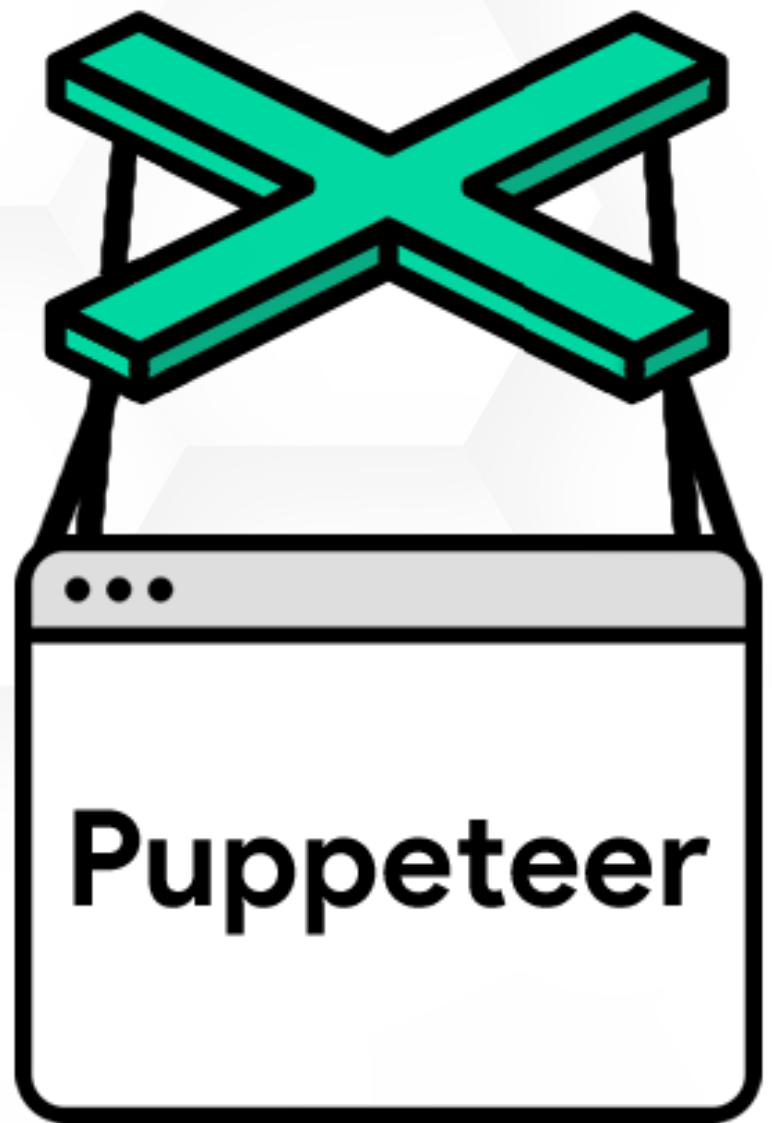
- En estos, se utilizan atributos más estables, definidos por uno mismo con el fin de ser usados por la herramienta.



```
await page.click(' [data-testid="login-btn"] ' );
```



```
<button data-testid="login-btn">Login</button>
```



Introduction | Puppeteer

Puppeteer is a JavaScript library which provides a high-level API to

➤ pptr.dev

DEMO

Conclusiones

El uso de Puppeteer para pruebas automatizadas E2E demuestra ser una herramienta eficaz para validar flujos completos dentro del navegador. Su capacidad para controlar navegadores de forma precisa permite simular interacciones reales del usuario, lo que contribuye a detectar errores que otras capas de pruebas no logran visualizar.

Hay que tener en cuenta que Puppeteer no es un framework de testing completo; requiere complementarse con herramientas adicionales para lograr aserciones, reportería y una estructura sólida de pruebas. Además, debido a que las pruebas E2E suelen ser más lentas y sensibles a cambios en la interfaz, es fundamental implementar buenas prácticas como el uso de selectores estables, sincronización adecuada y un diseño modular de los casos de prueba.

GRACIAS