

---

## Funciones Principales que Realiza Playwright

### 1. Interacción y Control de la Página (Core)

Función/Capacidad	Descripción
Control Multi-Navegador	Automatiza y ejecuta pruebas en <b>Chromium, Firefox y WebKit</b> (Safari), usando una sola API.
Control Multi-Lenguaje	Permite escribir automatizaciones en <b>JavaScript/TypeScript, Python, Java y C#/.NET</b> .
Emulación	Permite emular <b>dispositivos móviles</b> (iPhone, Android), <b>zonas horarias</b> , <b>geolocalización</b> y <b>esquemas de color</b> (prefers-color-scheme).
Pruebas de Aislamiento	Utiliza <b>Contextos de Navegador Aislados</b> para garantizar que cada prueba sea

	independiente y no comparta estado.
<b>Capturas de Pantalla y Video</b>	Permite tomar <b>capturas de pantalla</b> de la página completa o de elementos específicos y <b>grabar video</b> de las interacciones.

## 2. Fiabilidad y Estabilidad

Función/Capacidad	Descripción
<b>Auto-Espera (Auto-Waiting)</b>	<b>Espera automáticamente</b> a que los elementos sean visibles, estén habilitados y se puedan interactuar antes de realizar una acción (clic, escribir, etc.), lo que hace las pruebas más fiables.
<b>Selectores Potentes</b>	Ofrece selectores basados en <b>texto, rol de accesibilidad, contenido (:has), y encadenamiento (&gt;&gt;)</b> para apuntar a elementos de manera más robusta.
<b>Assertions (Expect)</b>	Proporciona aserciones específicas de Playwright que también <b>auto-esperan</b>

	para validar estados de elementos (ej: <code>expect(locator).toBeVisible()</code> ).
--	--

### 3. Rendimiento y Paralelismo

Función/Capacidad	Descripción
Ejecución Paralela	Ejecuta las pruebas en <b>paralelo</b> de forma nativa a nivel de archivo o de test, acelerando significativamente el tiempo de ejecución.
Contextos Persistentes	Permite <b>reutilizar sesiones de inicio de sesión</b> o estados complejos a través de múltiples pruebas sin la sobrecarga de la configuración.

### 4. Desarrollo y Debugging

Función/Capacidad	Descripción
-------------------	-------------

<b>Code Generation (codegen)</b>	<b>Graba las interacciones del usuario</b> en el navegador y genera automáticamente el código Playwright correspondiente.
<b>Playwright Inspector</b>	Una <b>herramienta gráfica</b> que permite <b>debuggear</b> las pruebas, <b>explorar selectores</b> y <b>ejecutar comandos</b> paso a paso.
<b>Trace Viewer</b>	Una herramienta visual para <b>analizar los detalles de la ejecución</b> de una prueba (pasos, DOM, red, capturas de pantalla) después de que ha fallado o terminado.

## 5. Red, API y Otros

<b>Función/Capacidad</b>	<b>Descripción</b>
<b>Interceptación de Red</b>	Permite <b>modificar, simular o bloquear</b> solicitudes y respuestas de red (HTTP/HTTPS) para controlar el comportamiento del frontend.
<b>Pruebas de API</b>	Incluye una librería específica

	(@playwright/test / request) para realizar <b>pruebas de API REST directamente</b> , complementando las pruebas E2E.
<b>Manejo de Archivos</b>	Capacidad para <b>subir archivos</b> (setInputFiles) y <b>descargar archivos</b> (waitForEvent('download')).
<b>Manejo de IFrames y Diálogos</b>	Interacción con marcos incrustados ( <b>iframes</b> ) y manejo de diálogos de navegador ( <b>alert, confirm, prompt</b> ).

---

Playwright es una solución **completa y moderna** que busca reemplazar y superar a las herramientas de automatización de primera generación, enfocándose en la **fiabilidad y la velocidad**.